Essays on Predictive Analytics in E-Commerce

Dissertation

zur Erlangung des wirtschaftswissenschaftlichen Doktorgrades der
Wirtschaftswissenschaftlichen Fakultät der Georg-August-Universität Göttingen

vorgelegt von

Patrick Urbanke

aus Erlangen

Göttingen, 2016

Contents

List of Abbreviations

AdaBoost = Adaptive boosting

BI = Business intelligence

CART = Classification and regression trees

CHAID = Chi-squared automatic interaction detection

ChiSelect = Feature selection based on chi-squared statistic

ERT = Extremely randomized trees

GASSIST = Genetic algorithms based classifier system

GB = Gradient boosting

HIDER = Hierarchical decision rules

LDA = Linear discriminant analysis

LogDimRed = Dimensionality reduction based on logistic functions

LR = Logistic regression

IS = Information systems

MLP = Multilayer perceptron

NMF = Non-negative matrix factorization

NNDimRed = Dimensionality reduction based on neural network

NoNom = No nominal indicators

NOW G-net = Network of workstations genetic networks

OCEC = Organizational coevolutionary algorithm for classification

OrigData = Original data

PCA = Principal component analysis

RanProj = Random projection

RanTSVD = Randomized truncated singular value decomposition

RectLin = Rectified linear transformation

RF = Random forest

RMSE = Root mean squared error

RNN = Randomized neural network

ROC = Receiver operating characteristic

SIA = Supervised inductive algorithm

SVM = Support vector machine

SWIG = Simplified Wrapper and Interface Generator

UCS = Supervised classifier system

XCS = Extended classifier system

List of Figures

List of Tables

1 Introduction

1.1 Motivation

The “data explosion” enabled by the fact that the costs of storing and processing large amounts of data have decreased significantly (Bhimani and Willcocks, 2014) and the new technologies resulting from this trend constitute the biggest disruption in business practise and business research since the rise of the internet (Agarwal and Dhar, 2014). In particular, Business Intelligence (BI) has been identified as an important research topic for both practitioners and academics in the field of Information Systems (IS) (Chen et al., 2012). Machine learning algorithms have been successfully applied to a large variety of BI problems, including sales forecasting (Choi et al., 2014; Sun et al., 2008), forecasting wind power output (Wan et al., 2014), analysis of patient outcome (Liu et al., 2015), fraud detection (Abbasi et al., 2012) or recommender systems (Sahoo et al., 2012). However, very little research is concerned with machine learning issues that are unique to BI: Even though existing machine learning algorithms are occasionally modified for a specific BI problem (Abbasi et al., 2010; Sahoo et al., 2012), IS research in BI as well as BI practice is generally limited to applying existing machine learning approaches and statistical concepts that were originally developed for other domains to specific BI problems (Wu et al., 2008; Chen et al., 2012).

One of the two motivations for this dissertation is to close this gap. The dissertation will focus on four problems that are unique to BI:

First, when evaluating a new predictive method, the machine learning literature traditionally focuses on predictive accuracy almost exclusively. However, in a BI context, the effectiveness of a decision support system is also determined by other factors, such as how well it provides new, additional information.

Second, BI problems often involve a combination of numerical and nominal variables. Modeling the complex interactions between these variables will in many cases lead to datasets consisting of a very high number of sparse indicators in combination with a low number of dense indicators. There is little research on how these complex interactions can be modeled effectively.

Third, business intelligence problems are more heterogeneous than artificial intelligence problems such as image classification or speech recognition. Whereas an algorithm that works well on one image recognition problem could reasonably expected to do well on another image recognition problem, an algorithm that does well on one business problem does not necessarily do very well on another. This necessitates an approach for customizing machine algorithms to specific business problems.

Fourth, most machine learning algorithms are "black box" approaches that can not be used to gain an understanding of the underlying relationships between the indicators. However, business practitioners problems often require interpretability. This dissertation also introduces machine learning approaches that are interpretable and enable an intuitive visualisation of non-linear relationships contained in BI datasets.

This dissertation will focus on the important BI problems of product returns in online retail for an illustration and a practical application of the proposed concepts. Many online retailers fail to be profitable (Rigby, 2014) and product returns have been recognized as a major cause for this problem (Grewal et al., 2004). In addition to being a cost factor for online retailers, product returns are problematic from an environmental point of view: In the

logistics literature, it is widely recognized that the ”last mile” of the delivery chain, when the product is delivered from the store to the customer's doorstep, is most CO_2-intensive (Browne et al., 2008; Halldórsson et al., 2010; Song et al., 2009). Product returns repeat this energy-intensive step, thus decreasing the environmental friendliness of online retail as a business model relative to more traditional forms of retail. However, online retailers cannot simply prohibit product returns, because they are an essential part of their business model: It has been demonstrated that enabling customers to return unwanted products has a positive impact on customer satisfaction (Cassill, 1998), purchase rates (Wood, 2001), future buying behaviour (Petersen and Kumar, 2009) and customers' emotional responses (Suwelack et al., 2011). A promising approach is to focus on impulsive or even compulsive shopping behaviour (LaRose, 2001) and fraudulent returns (Speights and Hilinski, 2005; Wachter et al., 2012). To date, there are no such strategies in the academic literature on the topic. In fact, most strategies are one-size-fits-all approaches which do not differentiate between wanted and unwanted returns (Walsh et al., 2014).

Another motivation for this dissertation is to present the basis for a strategy for handling product returns that addresses the identified shortcoming in the extant literature, namely a strategy of prediction and prevention which identifies consumption patterns associated with a high probability of a product returns and intervenes before the transaction even takes place. This dissertation develops several prediction models that form the basis for such a strategy and show that it is feasible, given effective interventions. The dissertation also studies the interactions of different product return drivers at greater depth, enabling practitioners to develop other approaches for avoiding product returns.

In summary, the motivation for this dissertation is a dual one: On the hand it is methodological, as it introduces new statistical and machine learning approaches, on the other hand it is practical, as these approaches are applied to provide solutions for and study a real-world business problem, namely that of product returns in online retail. This duality, which is also reflected in the research questions, is appropriate given the strongly interdisciplinary nature of IS.

1.2 Research Questions

In the following section, the research questions underlying this dissertation will be introduced at greater detail. The section also describes how this dissertation can be positioned in IS research as well as the philosophy of science.

1.2.1 Machine Learning for Business Intelligence

Conducting a review of the recent literature regarding predictive analytics in IS (see chapter 2) reveals that the literature almost exclusively focuses on the concept of *outperformance*. A newly introduced algorithm is considered to be a relevant contribution to the literature if and only if it can be shown to provide better predictive accuracy than state-of-the-art approaches.

Whereas this dissertation does not question the usefulness of the concept of outperformance, there are alternative concepts for evaluating predictive methods that are also useful in a BI context, for instance *overall predictive accuracy* or *forecast encompassing* (Harvey et al., 1998).

Overall predictive accuracy measures whether a predictive method generates statistically

significant results. This is important, as assessing predictability is an important part of predictive analytics (Shmueli and Koppius, 2011).

The concept of forecast encompassing measures whether a predictive method constitutes a statistically significant contribution to an ensemble of existing machine learning algorithms and thus compensates for shortcomings of extant methods. This is particularly relevant to business practitioners when they already rely on a selection of predictive methods for important business prediction and would like to ascertain whether adding a new predictive method generates substantially new information or whether the information generated by the new predictive method is already contained in the predictive models already used.

However, to date there is no statistical approach that integrates the concepts of outperformance, superior predictive accuracy and forecast encompassing into a single, coherent statistical framework. In this dissertation, we would like to close that gap, which leads us to research question 1.1:

> *RQ 1.1: How can predictive methods in business intelligence be statistically evaluated?*

BI datasets are often stored in relational or more modern database systems (e.g. Apache Hadoop or Apache Spark) and consist of a combination of numeric variables, nominal variables and sometimes unstructured data. These variables interact with each other to produce a certain outcome that researchers or practitioners want to predict.

Taking the example of product returns in online retail, which is the most important use case in this dissertation, such a dataset might look as follows: The likelihood of a product return might depend on the percentage of products that the customer has previously returned (numeric variable) and the price of the product (numeric variable), but also on factors such as the size of the product (nominal variable) and the brand of the product (nominal variable). Moreover, the likelihood of a product return will also be influenced by interactions between the sizes and brands of products within a particular basket or in comparison to previous purchases made by the customer. For instance, if most of the items of clothing in the virtual shopping basket are of size XL, but there is one product that is of size M, then one might expect that the likelihood of the smaller product being returned is higher. Likewise, if it is known that a customer has shown a very high propensity to return products of a certain brand in the past, he or she could be expected to continue doing so in the future. In addition, if the brand is similar to brands that the customer has returned often in the past, similar interactions might be observed. In addition, there is no industry-wide standard for product sizes in fashion, meaning that size XL for one brand might mean something different than for another. This implies that researchers who hope to model these interactions would have to create a possibly five-digit number of sparse variables.

Scenarios similar to the one illustrated above for the problem of product returns in online retail can be found in many other BI problems. The challenge for BI researchers is to model the complex interactions between these variables effectively and then reduce the resulting, possibly very high-dimensional, dataset for a prediction model with minimal information loss. This results in research question 1.2:

> *RQ 1.2: How can the complex interaction between nominal and numeric variables be modeled and the resulting a high-dimensional datasets reduced with minimal information loss?*

Artificial intelligence researchers aim to develop approaches that generalize well and can be

adapted to many different datasets. In theory, the multilayer perceptron is such an universal approximator (Hornik et al., 1989). However, the artificial intelligence literature has developed specialized neural networks structures such as the long short memory neural network for speech recognition (Hochreiter and Schmidhuber, 1997) and other sequential datasets or convolutional neural networks for image segmentation (Ciresan et al., 2011, 2012). This suggests that the inclusion of prior knowledge on the structure of a dataset can improve predictive accuracy. However, datasets related to business intelligence problems are typically more heterogeneous that image or sound data, often comprising of datasets from different sources and combining structured with unstructured data. This calls for the development of an approach to include prior knowledge on a business problem in the architecture of a machine learning algorithm.

> *RQ 1.3: How can machine learning algorithms be customized to specific business intelligence problems?*

Most machine learning algorithms are "black box" approaches: They are useful for generating predictions, but not useful for explaining relationships. Standard statistical approaches are useful for measuring simple, linear relationships, but are less useful for complex, non-linear relationships and interactions between different variables. In the framework proposed by Gregor (2006), standard machine learning approaches are useful for P-theories (theories that predict, but don't explain) whereas classical statistical approaches are useful for E-theories (theories that explain, but don't predict).

However, in most BI settings both explanation and prediction are required. To date, few approaches exist that are designed for both capturing complex non-linear interactions thus generating accurate predictions as well as being interpretable thus providing researchers and practitioners with a good understanding of the underlying relationships within the datasets. Recent developments in data storage and processing, known as the "data explosion" (Bhimani and Willcocks, 2014), and the resulting large-scale datasets provide the opportunity for applying and constitute the necessity for developing such approaches.

Research question 1.4 summarizes these considerations:

> *RQ 1.4: How can machine learning algorithms be designed to be interpretable and provide researchers and practitioners with an understanding of complex, non-linear relationships in large-scale datasets?*

1.2.2 Product Returns in E-Commerce

In order to fill the identified gap of dynamic, customer-oriented strategies (Walsh et al., 2014), this dissertation proposes a strategy of prediction and prevention. The main idea of this strategy is to develop a system that predicts the likelihood of a product being returned as the customer puts together the virtual shopping basket. If necessary, the system can intervene, before the customer has even hit the order button.

This dissertation uses self-developed algorithms to develop the most important prerequisite for such a strategy, namely a prediction model for product returns in e-commerce. It also demonstrates how the prediction system can be embedded into the business context of product returns in e-commerce and evaluate whether the system is sufficiently accurate for the business case, thus investigating whether the proposed strategy of prediction and prevention is feasible. This research goal is summarized in research question 2.1:

RQ 2.1: How can product returns in e-commerce be predicted such that the resulting predictive accuracy is sufficient for a strategy of prediction and prevention to be feasible?

In addition to developing the prerequisite for a specific prediction strategy, it is desirable to widen the horizon and gain a more general understanding of customer behavior with regard to product returns as a whole. In particular, the goal is to understand what the main drivers for product returns are and how they interact with each other.

For instance, it can be shown from simple, descriptive statistics that product returns are positively correlated to the price of the product. However, this relationship might be moderated by additional variables such as other products in the basket. For instance, if there is a high number of similar products in the basket, it might be more likely that the customer will not keep them all, especially if they are all very expensive. Studying these interactions can give us a deeper understanding of customer behavior and can form the basis of developing new strategies for return management.

This leads us to research question 2.2:

RQ 2.2: What are the main drivers of product returns and how are they related to each other?

1.3 Structure of the Dissertation

This section provides an overview of the publications included in this dissertation. It also explains how each of the chapters is related to the research questions underlying this dissertation.

1.3.1 Publications included in the dissertation

An overview of the publications included in this dissertation is provided in Table 1. The table includes the arrangement of the chapters, the title of the original publication, the outlet in which it was published or is submitted to, the rating according to VHB Jourqual 3[1] of the outlet, the current status of the publication and the author's contribution.

All of the publications have been modified before inclusion in this dissertation. Each of the chapters has been enriched with additional, previously unpublished material. An overview of these changes is presented in Table 2.

1.3.2 Relationship between chapters and research questions

An overview of how the publications included in this dissertation relate to each of the research questions is provided in Figure 1.

As explained in detail in section 1.2, this dissertation addresses two blocks of research questions, which are related to the methodology of machine learning in advanced business intelligence and the use case of product returns in online retail respectively. Each of the chapters addresses these blocks of research questions from a different perspective.

Chapter 2 develops a framework for evaluating predictive methods. This framework is di-

[1] VHB-Jourqual 3, http://vhbonline.org/service/jourqual/vhb-jourqual-3/teilrating-wi/, retrieved 2015-12-27

Chapter	Title	Outlet	Rating (VHB)	Current status	Own Contribution
2	A Unified Framework for Evaluating Predictive Methods	Proceedings of the International Conference on Information Systems 2014, Auckland, New Zealand	A	Published	90%
3	Predicting Product Returns in E-Commerce: The Contribution of Mahalanobis Feature Extraction	Proceedings of the International Conference on Information Systems 2015, Fort Worth, Texas	A	Published	80%
4	A Customized and Interpretable Neural Network for High-Dimensional Business Data - Evidence from an E-Commerce Application	Proceedings of the International Conference on Information Systems 2016, Dublin, Ireland	A	Submitted	90%
5	An Interpretable Machine Learning Algorithm Based on Randomized Neural Networks	Decision Support Systems	B	Submitted, passed first round of reviews	90%

TABLE 1: Overview of the publications included in this dissertation

rectly related to research question 1.1 and forms the statistical basis of the subsequent chapters.

Chapter 3 develops a prediction model for product returns in online retail. As the dataset related to this business intelligence problem is very high-dimensional, this chapter develops a new technique for feature extraction and dimensionality reduction. The self-developed technique outperforms state-of-the-art dimensionality reduction algorithms and the prediction model is sufficiently accurate for a strategy of prediction and prevention to be feasible. Chapter 3 is related to research questions 1.2 and 2.1.

Chapter 4 extends the concepts developed in chapter 3 in several regards: First, it develops a neural-network-based version of the feature extractor developed in chapter 3. Second, it develops a practical method of how complex interactions between nominal variables can be modeled, thus addressing a key component of research question 1.2. Third, it is possible to customize the approach to different business problems. Fourth, it provides a concept for extracting interpretable constructs from high-dimensional data which can be used to identify

some of the main drivers of product returns. Chapter 4 is related to research questions 1.2, 1.3, 1.4, 2.1 and 2.2.

Chapter	Title	Changes
2	A Unified Framework for Evaluating Predictive Methods	The following previously unpublished material was included: • Estimation of p-value using Bayesian statistics • Generalization of the approach to panel-data structures • Demonstration using example study from the field of information systems
3	Predicting Product Returns in E-Commerce: The Contribution of Mahalanobis Feature Extraction	The following previously unpublished material was included: • Generalization of algorithm to non-linear applications • Multilayer perceptron added to classifiers • Added precision-recall plot
4	A Customized and Interpretable Neural Network for High-Dimensional Business Data - Evidence from an E-Commerce Application	The following previously unpublished material was included: • Added test for forecast encompassing to evaluation
5	An interpretable machine learning algorithm based on randomized neural networks	The following previously unpublished material was included: • Discussion of the relative efficiency of different approaches to calculating gradients • Detailed description of algorithm in pseudocode • More extensive description of indicators • More extensive analysis of results, including additional visualization, more detailed description of the resulting randomized neural networks, evaluation of all iterations

TABLE 2: Changes made to the publications included in this dissertation

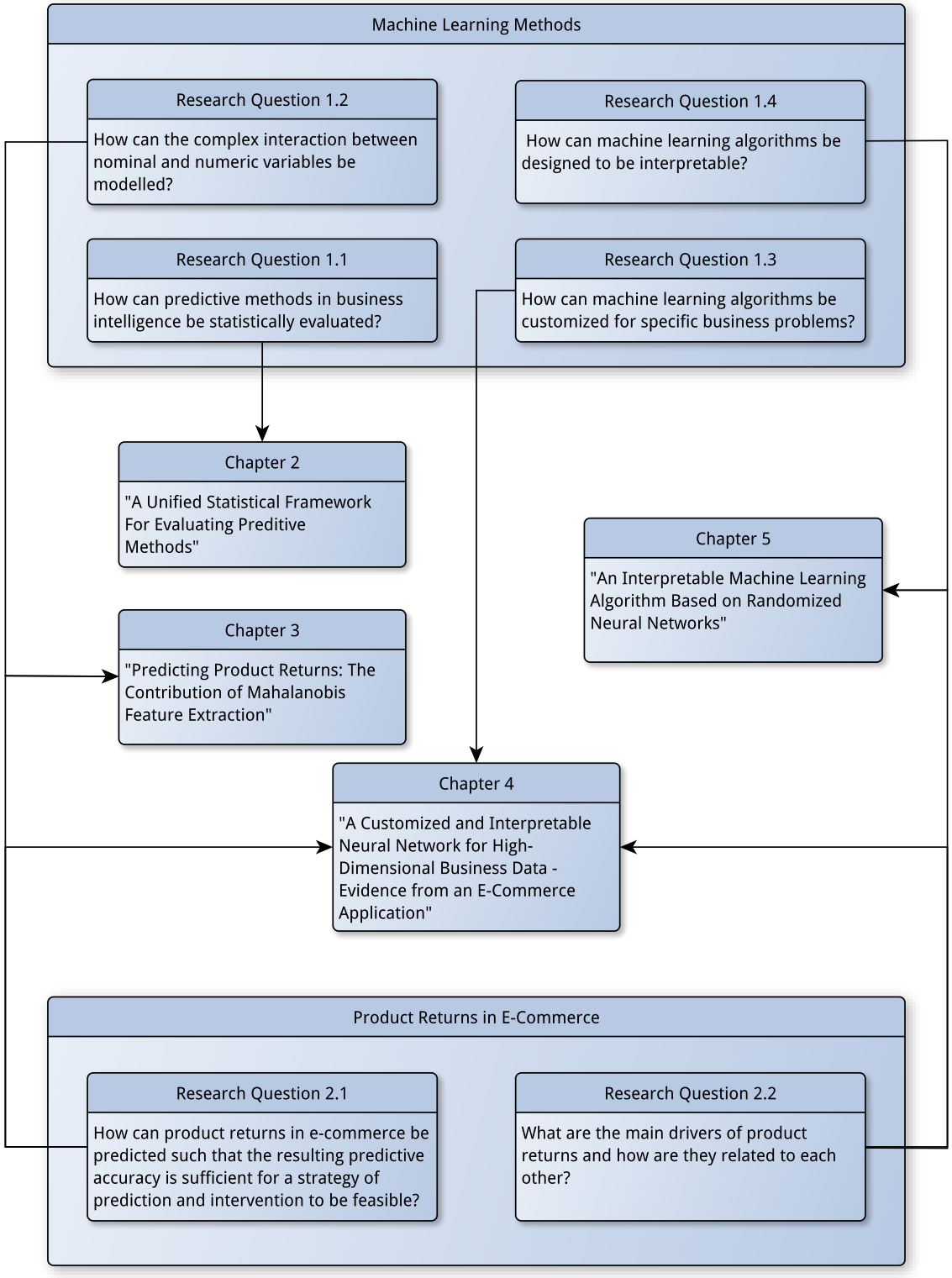


FIGURE 1: Relationship between chapters and research questions

Chapter 5 presents a machine learning algorithm that can be visualized effectively thus providing researchers and practitioners with an approach to extract information rather than predictions from data. This approach is then applied to product returns in online retail. Chapter 5 is related to research questions 1.4 and 2.2.

1.4 Conceptual Relationships Between Chapters

This section explains the conceptual relationships between the chapters. The statistical concepts and algorithms developed in this dissertation are closely related to each other, making this dissertation a cumulative dissertation in the very sense of the word. In the beginning of this section, these relationships are explained. The next subsection details one of the major recurring themes in this dissertation, namely distributed memory parallelization. The section concludes by listing how the methods of the chapters are applied to product returns in e-commerce.

1.4.1 Mathematical Reasoning and Algorithm Development

An overview of the mathematical relationships between the different chapters in this dissertation is provided in Figure 2.

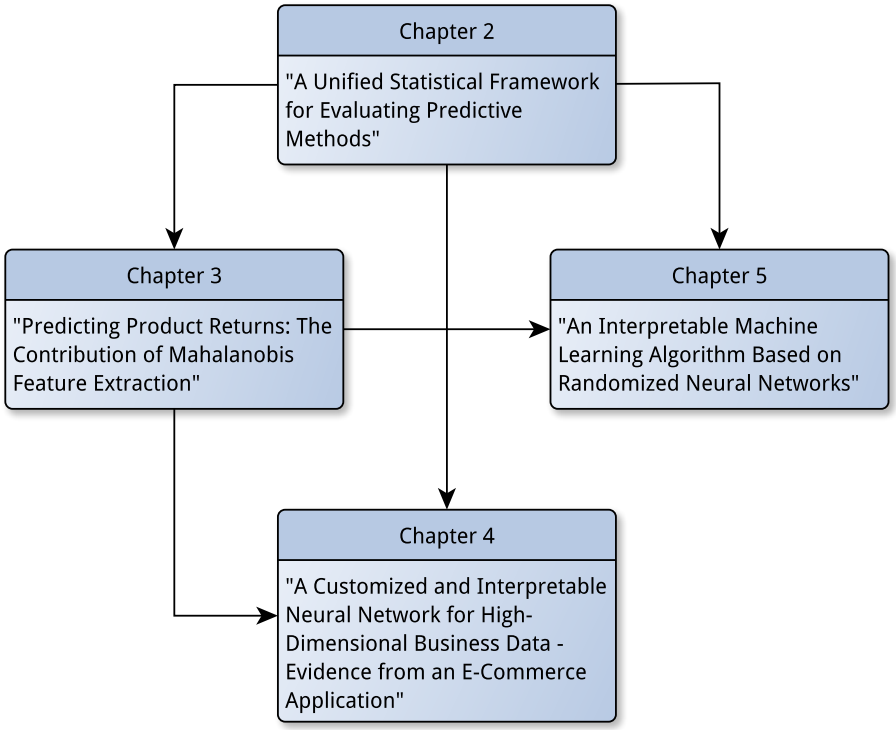


FIGURE 2: Relationship between statistical concepts introduced in the chapters

The statistical framework introduced in chapter 2 constitutes the theoretical basis for the dissertation. All of the subsequent chapters directly build on the concepts developed in this chapter. The statistical framework has originally been developed for the evaluation of predictive methods and is presented as such in chapter 2, however the main context in which it is used in all subsequent chapters is for the development of novel algorithms rather than their evaluation.

Chapter 3 develops an algorithm for linear feature extraction in chapter 3 based on the statistical concept introduced in the preceding chapter. It also uses the framework to evaluate the predictive accuracy of the resulting algorithm.

Chapter 4 develops a neural-network-based version of the algorithm and applies it to a more

high-dimensional dataset which allows for modeling complex interactions between different variables. It also uses the framework to evaluate the predictive accuracy of the resulting algorithm.

Chapter 5 introduces a variation to randomized neural networks (RNN) that allows for a a visualization of complex, non-linear relationships between the different drivers for product returns. To make the randomized neural networks useful for visualization, a combination of a feature selector and a boosting procedure is used, which were developed on the basis of the statistical framework in chapter 2 and the feature extractor developed in chapter 3.

1.4.2 Distributed Memory Parallelization

Distributed memory parallelization is an important topic in modern data science applications. In 2015, many important tech companies have released distributed memory machine learning libraries, for instance Microsoft or Samsung.

All of the concepts and algorithms developed in this dissertation are integrated in a single coherent library, mainly written in C and C++, which is based on OpenMPI , a widely used framework for distributed memory parallelization. The library relies on SWIG (Beazley, 1996) to interface the library to Python. This allows us to combine the ease of use and the wide selection of scientific libraries included in Python with the speed and efficiency of C/C++.

Each of the chapters explains how the developed statistical approach or machine learning algorithm can be efficiently rewritten in terms of the reduce operations needed for an OpenMPI setting.

1.4.3 Product Returns in E-Commerce

Product returns in e-commerce are the most important use case in this dissertation. In order to fill the identified gap of dynamic, customer-oriented strategies (Walsh et al., 2014), this dissertation develops a system that can predict whether a customer is going to return a specific product at the time the product is being ordered and intervene if necessary. Possible intervention strategies include the following:

1. *Limiting payment options* - Previous research has shown that the payment option has great influence on the likelihood of a product being returned particularly in the fashion sector, with a return rate of about 46%, if the customer is billed and about 26%, if the customer pays in advance (Asdecker, 2015). This can be attributed to the fact that returns are less tedious for customers when being billed (Pur et al., 2013). Therefore, a possible intervention strategy to reduce the return rate could be to limit the payment options when the predicted return rate in the basket is deemed too high.

2. *Reducing the period of time in which products can be returned* - As mentioned above, many online retailers offer their customers more than the minimum time period required by law to return their products. A possible intervention strategy might therefore be to reduce that time period to the legally required minimum, if the probability of products being returned is too high.

3. *Moral suasion* - When the probability of a product being returned is deemed too high, a pop-up window appears, reminding the customer of the environmental impact as-

sociated with product returns. Previous studies have demonstrated that moral suasion can effectively influence customer behavior in the context of labeling environmentally friendly products (Aguilar and Cai, 2010; Bjørner et al., 2004a,b; D'Souza et al., 2006).

4. *Rejecting the transaction* - In extreme cases, the transaction can be outright rejected.

The dissertation uses insight gained from the marketing literature as its intellectual foundation to build its data model and extract appropriate indicators. Previous studies have established that impulsive and compulsive consumption patterns are a major cause for product returns (LaRose, 2001). Furthermore, impulse shopping is associated with a number of attitudes and behavioral patterns, such as a hedonistic consumption tendency (Hausman, 2000; Joo Park et al., 2006; Moe, 2003), in-store browsing (Beatty and Ferrell, 1998; Moe, 2003), gender (Dittmar et al., 1995) or visual stimuli (Adelaar et al., 2003; Joo Park et al., 2006). These factors can be indirectly observed in an e-commerce context (Moe, 2003). *Hedonistic consumption tendency* is reflected in the type of goods that the customer purchases. The marketing literature differentiates between *hedonic* and *utilitarian* goods that are associated with distinct consumption patterns (Brookshire et al., 1982; Dhar and Wertenbroch, 2000; Okada, 2005). This implies that knowledge on which goods a customers tends to buy can provide valuable insight into his or her tendency for hedonic consumption. Note that it is not necessary to a priori define which products are hedonic and which are utilitarian for this approach to be feasible. An effective machine learning algorithm is sufficiently adaptable to understand these relationships without prior input. *In-store browsing* occurs when the customer looks at a large number of product before making a purchase decision (Beatty and Ferrell, 1998; Moe, 2003). When a customer places a large number of similar items into the same shopping basket, we interpret this as the e-commerce equivalent of in-store browsing. Even though this dissertation is not based on any personalized information, the customer's *gender* can be indirectly inferred from the products purchased. *Visual stimuli* can be measured by counting the number of pictures a product is advertised with.

For this dissertation, we have established a cooperation with a major German online retailer specializing in fashion. Through this cooperation, we have been able to obtain a sufficiently extensive dataset related to product returns containing a total of 3,637,654 individual transactions.

With the exception of chapter 2, which sets the mathematical basis for all subsequent chapters, all of the studies in this dissertation are related to product returns in online retail.

Chapter 3 develops a predictive model for product returns in online retail. Since there is a gap in the existing literature on product returns regarding dynamic, customer-oriented strategies (Walsh et al., 2014), the chapter introduces such a strategy based on the developed prediction model. It also demonstrates that the strategy is feasible based on the predictive accuracy generated by the model.

Chapter 4 extends this predictive model by modelling the complex interactions between different product characteristics. It also investigates how nominal, sparse and high-dimensional variables can be combined to interpretable constructs that impact the probability of a product being returned.

Chapter 5 investigates how different product return drivers interact with each other to impact the likelihood of a product being returned. It visualizes these interactions using three-dimensional contour plots and generating insight on return behavior.

1.5 Research context and design

1.5.1 Positioning in Information Systems Research

Within the major streams of IS, this research is part of the decision support and design science stream. Researchers that are part of the decision support and design science stream of IS research analyze and develop systems to support human decision makers or improve business processes. The techniques used include mathematical programming, forecasting, simulation or expert systems (Banker and Kauffman, 2004).

This dissertation is part of that research stream as it develops method that help human decision makers gain insight into complex datasets and improve the efficiency of business processes. Specifically, this dissertation develops a method for the systematic prediction of product returns in online retail that helps avoid product returns before they even occur and provides more detailed insight to human decision makers on how different factors influence product returns at the same time. In that, this dissertation makes use of forecasting methods, as many studies in the decision support and design science stream do. At the same time, this dissertation also draws from insight from the fields of computer science and marketing, which are disciplines related to the decision support and design science stream. It uses numerous machine learning methods from the field of computer science as the basis of the forecasting models and builds its data models on the basis of insight gained from the marketing literature on product returns in online retail.

In recent years, the decision support and design science stream has increasingly moved towards statistics and computer science. Agarwal and Dhar (2014) identify data science as a key challenge for and an important opportunity for IS research. For instance, in a study that received the 'Paper of the Year Award' at *MIS Quarterly*, Abbasi et al. (2010) develop a custom-made kernel function combined with a Support Vector Machine to identify fake websites. Sahoo et al. (2012) present a hidden Markov model to develop a recommender system.

In addition, this dissertation builds on criticism of applying classical statistical concept to large datasets, which is often referred to as the "p-value problem". Lin et al. (2013) and Cohen (1992) remind us of the shortcomings of standard statistical concepts such as statistical significance for large datasets. They argue that the concept of statistical significance is becomes increasingly meaningless as the size of datasets increase.

This research relates to these ideas in several ways: The research on dimensionality reduction and analysis of high-dimensional BI datasets (chapters 3 and 4) is directly relevant to the use of machine learning techniques for BI problems. The research on interpretable machine learning (chapter 5) integrate more traditional approaches to BI and decision support systems, such as visualization, with more advanced BI techniques, such as machine learning.

Statistical significance is an inherently linear concept: Factors that have a strongly positive impact in some contexts and a strongly negative impact in others can be deemed statistically insignificant unless the modeller explicitly takes these non-linear relationships into account and linearizes them. If their purpose is to use large datasets to study non-linear relationships between variables, then researchers cannot rely as strongly on the concept of statistical significance as they would in more traditional settings. The concepts proposed in chapter 4 and 5 of this dissertation can therefore be regarded as addressing the p-value problem by replacing the more traditional concept of statistical significance with modern visualization

techniques or tables to interpret neural network nodes.

1.5.2 Positioning in the Philosophy of Science

Shmueli and Koppius (2010) and Shmueli and Koppius (2011) demonstrate that most research in IS that claims to predictive uses statistical tools that were developed for explanation and are, in the view of the authors, not appropriate for predictive purposes. They argue that explanatory statistical modeling and predictive analytics are two distinct concepts that should not be conflated.

According to Shmueli and Koppius (2011), these concepts differ in terms of their goal of analysis, variables of interest, model building, model building constraints and model evaluation: In predictive analytics, the *goal of analysis* is to develop a model that can accurately predict new observations previously unknown to the model. In explanatory statistical modeling, the goal is to test hypotheses reflecting assumptions about causal relationships. In predictive analytics, the *variables of interest* are limited to observed, measurable variables. Explanatory statistical modeling sees these observed variables as representing more abstract, unobserved theoretical constructs and studies the causal relationships between them. In predictive analytics, *model building* aims to minimize the bias and variance of the prediction (in other words to minimize the prediction error). Researchers need to avoid overfitting to a specific dataset. In explanatory modeling, the goal is to minimize the bias of the model and researches need to avoid Type I and Type II errors. Explanatory statistical modeling imposes more rigorous *constraints on model building* than predictive analytics: For explanatory statistical modeling, the model must be interpretable, must support statistical hypothesis testing and must also reflect a theoretical model. Most machine learning algorithms can therefore not be used in this context. By contrast, in predictive analytics, the constraint is that the model can only rely on input variables that are available when the model is to be deployed. For instance, when the task to develop a model that predicts product returns when the customer orders the product, then that model cannot rely on information that can only be gathered after the the customer has ordered the product. Finally, predictive analytics models are *evaluated* using out-of-sample measures such as root mean squared error (RMSE). By contrast, explanatory models are evaluated using goodness-of-fit measures such as R^2 or evaluating the statistical significance of the individual input variables.

This dissertation uses predictive analytics as described above as its conceptual basis. The goal of the methods developed is to maximize predictive accuracy and it does not interpret the observed variables as representing more abstract latent constructs. Even though some of the methods developed in this dissertation are designed to be interpretable, it is technically not necessary for the use case of predicting product returns in online retail. Evaluation is strictly out-of-sample.

Shmueli and Koppius (2011) also argue that predictive methods can be used for theory evaluation: Models can be built based on different, competing theories and their out-of-sample predictive performance can be evaluated. This is view is based on Popper's philosophy of science: Scientific theory differs from myths in that is able to generate falsifiable predictions (Popper, 2005; Straub et al., 2005). It can also be interpreted in the framework provided by Gregor (2006), who differentiates between different kind of theories: The ideas described by Shmueli and Koppius (2010) and Shmueli and Koppius (2011) are related to the concept of P-theories, that is, theories which are can be used for prediction, but not explanation of phenomena.

All of the statistical concepts and algorithms developed in this dissertation are directly related to predictive methods and can therefore be used in the context described above: Chapter 2 provides a statistical framework for evaluating predictive methods. Chapters 3 and 4 provide methods for feature extraction which help researchers and decision makers summarize high-dimensional datasets for further analysis. In chapter 4 and 5, machine learning algorithms are developed that are interpretable to help researchers and decision makers gain insight into complex, non-linear interaction within datasets. This is particularly relevant from a philosophical point of view, as our goal in chapters 4 and 5 is to develop predictive methods that can not only to be used to evaluate P-theories, but are also useful for the development of EP-theories.

1.6 Anticipated Contributions

This dissertation addresses researchers and practitioners alike. To ensure practical relevance, the algorithms developed in this dissertation will concentrate on a single business intelligence use case, namely product returns in online retail. A number of industry stakeholders will stand to benefit from this dissertation:

1. *Online retailers* - The strategies and prediction models developed in this dissertation should be highly relevant to online retailers who hope are looking for effective strategies to reduce their return rate. In addition, the statistical analysis of the interactions of different factors influencing product returns issue can form a basis for the development of new strategies to avoid product returns.

2. *Business practitioners* - Even though all of the approaches developed in this dissertation are evaluated using a specific use case, the approaches are relevant for a wide selection of business practitioners. Many business practitioners require algorithms and approaches that help them model complex interactions between nominal indicators, approaches to gain information rather than just predictions from BI datasets or methods to evaluate predictive methods.

3. *Data scientists* - Those whose profession it is to develop statistical models from large-scale datasets will find the concepts developed in this dissertation useful. In many data science experiments, the first step is to gain a more fundamental understanding of the problem at hand. The approaches developed in chapters 5 and 6 of this dissertation constitute an important contribution in this regard. Moreover, data scientist stand to benefit from the evaluation methods developed in chapter 1 and the dimensionality reduction techniques developed in chapters 2 and 3.

This dissertation contributes to the academic literature in that it develops a set of new statistical approaches and algorithms for evaluating predictive methods, dimensionality reduction and interpretable machine learning. It also contributes to a deeper understanding of the important issue of product returns in online retail.

2 A Unified Statistical Framework for Evaluating Predictive Methods

Abstract

Predictive analytics is an important part of the business intelligence and decision support systems literature and likely to grow in importance with the emergence of big data as a discipline. Despite their importance, the accuracy of predictive methods is often not assessed using statistical hypothesis tests. Furthermore, there is no commonly agreed upon standard as to which questions should be examined when evaluating predictive methods. We fill this gap by defining three requirements that involve the overall and comparative predictive accuracy of the new method. We then develop a unified statistical framework for evaluating predictive methods that can be used to address all three of these questions. The framework is particularly versatile and can be applied to most problems and datasets. An extension to panel data structures that occur in many business analytics settings is also provided. Moreover, it includes a sampling procedure for cases when the assumption of a normal distribution fails.

Keywords: Machine learning, statistical methods, predictive modeling, business intelligence, decision support systems

2.1 Introduction

Predictive analytics has a long tradition in the information systems and computer science literature and has been demonstrated to be applicable to a wide variety of different domains, such as strategic sales management, sales forecasting (Choi et al., 2014; Sun et al., 2008), forecasting wind power output (Wan et al., 2014), analysis of patient outcome (Liu et al., 2015), fraud detection (Abbasi et al., 2012) or recommender systems (Sahoo et al., 2012). It involves the use of quantitative techniques, generally machine learning algorithms, to build predictive models (Shmueli and Koppius, 2010). It plays an important role in the business intelligence (Watson and Wixom, 2007) and decision support systems literature and is highly relevant to both researchers and practitioners.

However, in both fields, newly introduced predictive methods are almost without exception evaluated using the concept of outperformance: A predictive method is considered to be a contribution to the literature if and only if it can be demonstrated to outperform existing state-of-the-art methods according to a chosen measure or loss function.

The purpose of this study is twofold: First, we question the idea that outperformance is the *only* approach to evaluating predictive methods making the case that a rigorous evaluation of a newly introduced predictive method *also* includes a test for the *overall predictive accuracy* and a test for *forecast encompassing*. Second, we present a coherent statistical framework for evaluating predictive methods that unifies the concept of outperformance with the proposed alternative concepts.

In our view, there are three requirements that are of interest when evaluating a predictive method:

Requirement 1: The new predictive method generates statistically significant out-of-sample predictions.

Requirement 2: The out-of-sample predictions of the new predictive method outperform the out-of-sample predictions generated by alternative methods in a statistically significant manner.

Requirement 3: When corrected for the predictions generated by alternative methods, the out-of-sample predictions generated by the new predictive method are still statistically significant.

We argue that the examination of all of these three requirements is necessary for a thorough and rigorous evaluation of a predictive method.

Some might argue that the first requirement is already implied by the second and does not require explicit examination. We disagree for three reasons: First, we might be comparing the new predictive method to a method that is so poor that it does worse than a random walk. In that event, the new method might outperform the old one, even though it does not outperform a random walk. Second, it is in itself interesting to know which of the methods evaluated is actually useful. If the best method cannot be used (maybe because it is computationally too expensive), we would like to know whether there is an appropriate substitute among the alternative methods. Third, assessing predictability is an important part of predictive analytics (Shmueli and Koppius, 2011).

The second requirement is the one that most papers focus on. In predictive analytics, it is common practise to compare the out-of-sample predictions of a newly introduced predictive method with the out-of-sample predictions of state-of-the-art approaches. We agree with the importance of doing so. However, we posit that such comparisons should be supported by statistical hypothesis testing to attain scientific rigour.

To see the importance of the third requirement, consider the following:

Suppose that a new predictive method A and you can be demonstrated to outperform the existing state-of-the-art methods B, C and D. However, when compared to a simple combination of B, C and D, the method does not yield any useful additional predictive value. Could method A then be considered a useful contribution to the literature? Probably not.

Consider another case, in which a new method A does not outperform the state-of-the-art methods B, C and D. However, it can be demonstrated that it can add additional information to the predictions generated by B, C and D that these methods do not capture. Could method A then be considered a useful contribution to the literature? It probably could.

These two examples illustrate that a predictive method that outperforms the state of the art may not constitute a useful contribution and if a predictive method does not outperform the state of the art, this finding does not necessarily imply that the method does not constitute a useful contribution as long as it can be shown that the new method is not encompassed by existing methods. This result underlines the importance of the concept of forecast encompassing.

In this study, we propose a coherent statistical framework that unifies the three requirements presented above. The framework is applicable to a wide variety of different problems, including classification, regression, multilabel and multiclass classification problems, yet all of these problems are based on a common theoretical framework. The framework includes a sampling procedure to be used when the assumption of a normal distribution fails. We also provide an extension of the framework to panel data settings as frequently found in business analytics problems.

We demonstrate that our framework is a more effective tool than solely considering the concept of outperformance for statistical evaluation by using the dataset by Serrano-Cinca and Gutiérrez-Nieto (2013) which has been generously contributed for the purpose of this study. Using our framework, we are able to analyze the authors' results more rigorously and highlight their contribution more effectively than the authors themselves were able to do in their original study.

2.2 Literature Review

A number of hypothesis tests have been proposed for the evaluation of predictive methods in an out-of-sample setting, including tests for equal predictive ability (more commonly known as outperformance in the IS literature) and forecast encompassing.

Tests for equal predictive accuracy test the null hypothesis that the predictive performance of two predictive methods is similar. A widely used test for equal predictive accuracy was proposed by Diebold and Mariano (1995). The test is based on the assumption of a stationary time series with limited memory (in other words, an α-mixing) and assume that values that lie further apart than a certain time period τ are uncorrelated. They propose a test statistic based on the normal distribution that tests the null hypothesis that an appropriately chosen loss functions (such as the squared prediction error or log-loss) assumes similar values for two different predictive methods. They conduct simulations on artificial datasets and find that the test produces reliable results for larger datasets and when the prediction error is normally distributed, but find that heavy-tailed distributions can severely distort the accuracy of the test. West (1996) provides formal conditions under which the Diebold-Mariano test statistic converges to the normal distribution. He uses Monte Carlo simulations based on artificial samples to demonstrate that the test statistic is sufficiently accurate under conditions that are realistic for macroeconomic time series. Mc Cracken (2000) compares alternative loss functions and demonstrates that using the mean absolute error as well as integrating parameter uncertainty into the model can produce more powerful tests that the mean squared error using a dataset of excess returns of the S&P 500 as an example. Corradi et al. (2001) amend these findings by demonstrating that the conditions proposed by West (1996) hold when the size of the testing set grows smaller than the size of the training set (referred to as "prediction period" and "regression period" respectively in the original study). Harvey et al. (1997) criticize the Diebold-Mariano test for its lack of adaptability to small samples and analyse the finding that heavy-tailed distributions seriously distort the Diebold-Mariano test statistic. They develop an alternative method of estimating the variance in the Diebold-Mariano test statistic and propose the use of Student's t-distribution instead of the normal distribution as proposed in the original test. They use Monte Carlo simulations to demonstrate that these modifications increase the accuracy of the test statistics.

The idea of forecast encompassing states that a linear combination of different prediction methods, which has been determined in-sample, will yield no statistically significant reduction in a previously defined loss function when compared to an individual model (Clements and Harvey, 2010). A mathematically equivalent concept, *conditional efficiency*, was originally proposed by Nelson (1972) as well as Granger and Newbold (1973). Harvey et al. (1998) as well as Harvey and Newbold (2000) criticize the naive application of a linear regression to evaluate forecast encompassing, arguing that prediction errors are often not normally distributed and discuss alternative approaches which they argue to be more robust to heavy-tailed prediction errors. Harvey and Newbold (2003) present evidence for the claim

that predicion errors are often not normally distributed in real-world scenarios. Clark and McCracken (2001) provide a new approach for forecast encompassing which they demonstrate to have better convergence conditions on the basis of inflation projections. Clements and Harvey (2010) apply the idea of forecast encompassing to classification problems, develop an adapted version of the concept of forecast encompassing and apply to it predictions for the likelihood of a recession.

We surveyed the current practice of evaluating predictive methods in the IS literature according to the principles described in Shmueli and Koppius (2011). We structured the studies reviewed according to which of the three concepts for evaluating predictive methods identified above have been applied. An overview is provided in Table 3.

The most common proof-of-concept in the literature is outperformance. The predictions generated by the proposed predictive method are compared to to existing state-of-the-art methods to demonstrate its superiority. The paired t-test is by far the most popular statistical hypothesis test used in previous literature (see Table 3). Other tests used are the Wilcoxon signed-rank test, ANOVA, the F-test, the Z-test, Pearson's chi-squared-test and the Diebold-Mariano test (Diebold and Mariano, 1995). All of these hypotheses tests are based on the assumption of a normal distribution. If this assumption fails, the tests are no longer reliable.

Only a single study we surveyed (Serrano-Cinca and Gutiérrez-Nieto, 2013) explicitly examined the accuracy of the predictive methods corrected for alternative methods thus using forecast encompassing to evaluate their predictive method. However, this examination is conducted without using a statistical hypothesis test.

In this study, we integrate these three concepts into a coherent statistical framework. The framework is deliberately designed to be applicable to a wide variety of problems, including regression, single-label, multiclass and multilabel classification problems and panel data settings. It also addresses the issue that forecasting errors are often not normally distributed, which can potentially cause serious distortions to statistical hypothesis tests.

2.3 Calculation

2.3.1 Basic Idea of the Statistical Hypothesis Test

Suppose we have m continuous or discrete random variables X_1, X_2, X_3,..., X_m. The random variables X_a, a = 1, 2,..., m, are drawn without replacement from a dataset of out-of-sample predictions of m predictive methods. Further suppose that we have n continuous or discrete dependent random variables Y_1, Y_2, Y_3,..., Y_n. The variables Y_c, c = 1, 2,..., n, are drawn without replacement from a dataset of the values for the testing set that the predictive methods have been trained to predict. In most classification and regression problems, $n = 1$ holds, but for multiclass and multilabel classification problems the values for n are greater than one. Let x_a^i denote the i^{th} value in the dataset associated with variable X_a. Let y_c^j denote the j^{th} value in the dataset associated with variable Y_c. Let N be the number of instances in the dataset.

Suppose we wanted to compare a set of machine learning algorithms for the purpose of bankruptcy prediction. We would begin by training each of our algorithms on a training set. The variable x_a^i would represent the out-of-sample-prediction of algorithm a for company i. The dummy variable y_c^i would measure whether company i has in fact filed for bankruptcy. Because this is not a multi-label classification problem, there is only one c for y_c^i.

I. Accuracy of predictive methods	t-test	Yen and Hsu (2010)
	Not specified	Schumaker (2013)
	Evaluated without using hypothesis test	Chi et al. (2009); Coussement and Van den Poel (2008); David et al. (2012); Hagenau et al. (2013a); Lee et al. (2011); Papakiriakopoulos et al. (2009); Zhou et al. (2004)
II. Outperformance	Analysis of variance (ANOVA)	Arnott and O'Donnell (2008); Zhao et al. (2011)
	Diebold-Mariano test	Sermpinis et al. (2012)
	F-test	Cao and Parry (2009)
	Pearson's chi-squared-test	Coussement and Van den Poel (2008)
	t-test	Abbasi et al. (2010, 2012); Alfaro et al. (2008); Cao et al. (2012); Carbonneau et al. (2011); Chan and Franklin (2011); Chiang et al. (2006); Ince and Trafalis (2006); Ketter et al. (2012); Khansa and Liginlal (2011); Kim et al. (2011); Lam (2004); Li et al. (2012a); Li and Chen (2013); Oh and Sheng (2011); Sahoo et al. (2012); Yang et al. (2010)
	Wilcoxon signed-rank test	Fang et al. (2013); Kao et al. (2013); Lu et al. (2012, 2009); Saar-Tsechansky and Provost (2007)
	Z-test	Caulkins et al. (2006); Sinha and May (2004)
	Not specified	Bhattacharyya et al. (2011); Cui et al. (2012); Hu et al. (2007); Du Jardin and Séverin (2011); Li et al. (2014); Mangiameli et al. (2004)
	Evaluated without using hypothesis test	Adomavicius et al. (2012); Bai (2011); Bao et al. (2013); Bardhan et al. (2014); Chang et al. (2006a,b); Choi et al. (2011, 2013); Delen (2010); Delen et al. (2013); Fan and Zhang (2009); Gerber (2014); Hagenau et al. (2013b); Karbowski et al. (2005); Ketter et al. (2009); Kisilevich et al. (2013); Lau et al. (2013); Lee et al. (2011, 2012); Li and Wu (2010); Lu et al. (2012); Mehta and Bhattacharyya (2004); Murtagh et al. (2004); Olson and Chae (2012); Serrano-Cinca and Gutiérrez-Nieto (2013); Shen and Loh (2004); Shin et al. (2013); Sun and Li (2008); Su et al. (2012); Thomassey and Fiordaliso (2006); Yolcu et al. (2013); Zhong et al. (2005); Zhou et al. (2004)
III. Forecast encompassing	Evaluated without using hypothesis test	Serrano-Cinca and Gutiérrez-Nieto (2013)

TABLE 3: Evaluation of Statistical Significance in Predictive Analytics

We then define the variable X_c^a as follows:

$$X_c^a = \sum_{i=1}^{N} x_a^i y_c^i. \tag{1}$$

We establish the null hypothesis that the variables X_1, X_2, X_3,..., X_m do not constitute statistically significant predictions for the variables Y_1, Y_2, Y_3,..., Y_m. In other words, our null hypothesis is that the out-of-sample predictions of the predictive methods have no predictive value. We then interpret X_c^a as a random variable that has been constructed by randomly drawing from the two datasets without replacement:

$$E(X_c^a) = \frac{\sum_{i=1}^{N} x_a^i \sum_{j=1}^{N} y_c^j}{N} \forall a, c. \tag{2}$$

There are two interpretations for this approach. The first is related to the correlation coefficient. It can be easily shown that the test statistic proposed is proportional to the correlation coefficient of x_a^i and y_c^j:

$$N\sum_{i=1}^{N} x_a^i y_c^i - \sum_{i=1}^{N} x_a^i \sum_{j=1}^{N} y_c^j \sim X_c^a - E(X_c^a). \tag{3}$$

In fact, taking the formula for the variance-covariance matrix presented in (7), we demonstrate that for the very simple case of only one predictive method, the test statistic is closely related to Pearson's r:

$$\frac{X_c^a - E(X_c^a)}{\sqrt{V(X_c^a)}} = \frac{\sum_{i=1}^{N} x_a^i y_c^i - \frac{\sum_{i=1}^{N} x_a^i \sum_{j=1}^{N} y_c^j}{N}}{\sqrt{\frac{(N\sum_{i=1}^{N} x_a^i x_b^i - \sum_{i=1}^{N} x_a^i \sum_{k=1}^{N} x_b^k)(N\sum_{j=1}^{N} y_c^j y_d^j - \sum_{j=1}^{N} y_c^j \sum_{l=1}^{N} y_d^l)}{N^2(N-1)}}} = \sqrt{N-1} * r. \tag{4}$$

The first interpretation is that we test whether the absolute value of the correlation coefficient or the absolute value of Pearson's r is statistically significantly greater than what we would expect if we were to randomly reshuffle the data.

The second interpretation is related to the squared prediction error:

$$\sum_{i=1}^{N} (x_a^i - y_c^i)^2 - E(\sum_{i=1}^{N} (x_a^i - y_c^i)^2) \tag{5}$$

$$= \sum_{i=1}^{N} (x_a^i)^2 + \sum_{i=1}^{N} (y_c^i)^2 - 2\sum_{i=1}^{N} x_a^i y_c^i - \sum_{i=1}^{N} (x_a^i)^2 - \sum_{i=1}^{N} (y_c^i)^2 + 2\frac{\sum_{i=1}^{N} x_a^i \sum_{j=1}^{N} y_c^j}{N}$$

$$= -2\sum_{i=1}^{N} x_a^i y_c^i + 2\frac{\sum_{i=1}^{N} x_a^i \sum_{j=1}^{N} y_c^j}{N} \sim X_c^a - E(X_c^a).$$

The second is interpretation therefore that we test whether the sum of the squared prediction errors is statistically significantly smaller or greater than what we would expect if we were to randomly reshuffle the data. Both the correlation coefficient and root mean squared error are very common measures for evaluating predictive methods. The two rationales for our approach are thus closely related to standard measures for the quality of predictive methods used in the predictive analytics literature.

Let $\mathbf{X}$ be a vector containing the X_c^a for all a and c. We can either, for simplicity, approximate X_c^a using a normal distribution or use an alternative method which we discuss below. Note that we are not assuming any specific underlying distribution of the variables X_a and Y_c themselves. Let $\mathbf{V}$ be the variance-covariance matrix of $\mathbf{X}$. It follows from our assumptions that we can model $\mathbf{X}$ using a chi-squared distribution:

$$(\mathbf{X} - E(\mathbf{X}))' \mathbf{V}^{-1} (\mathbf{X} - E(\mathbf{X})) \sim \chi^2(n). \tag{6}$$

Returning to our example, we effectively test the null hypothesis that the out-of-sample bankruptcy predictions generated by the machine learning algorithms are as a whole uncorrelated to the dummy variable measuring whether the bank has in fact filed for bankruptcy. More precisely, we test whether the correlations are statistically significantly larger than what we would expect if we were to randomly reshuffle the data.

The reshuffling assumption can be justified on the following grounds: *If* the null hypothesis is true, *then* a random reshuffling of the data should not yield a systematically smaller test statistic then the original, non-reshuffled data set. Therefore, if we can show that a random reshuffling of the data *does* yield a systematically smaller test statistic then the original, non-reshuffled data set, then we have demonstrated that the null hypothesis is not true.

We will discuss the application of this approach to all three requirements below.

Having introduced the basic idea of the statistical hypothesis test, we are left with three questions: First, under what circumstance is the variance-covariance matrix singular and what can we do if it is? Second, how do we calculate the variances and covariances in $\mathbf{V}$? Third, how do we evaluate the statistical significance of a prediction given the predictions of other predictive methods? These questions are addressed in Theorems 1, 2 and 3, respectively.

2.3.2 Proof of Non-Arbitrariness

Suppose that our variables Y_1, Y_2, Y_3,..., Y_n represent a multiclass classification problem (meaning that $Y_1 + Y_2 + Y_3 + ... + Y_n = 1$). Note that our variables X_c^a are constructed such that we permutate through every possible combination of X_1, X_2, X_3,..., X_m and Y_1, Y_2, Y_3,..., Y_m. In such a case, it is straightforward to see that the variables X_c^a are linearly dependent, in which case $\mathbf{V}$ is singular and its inverse could not be found. This problem could be solved by arbitrarily dropping one of the variables Y from our set. More generally speaking, we do not include X_c^a in $\mathbf{X}$ if the inclusion would cause $\mathbf{X}$ to contain a subset of variables that are linearly dependent.

However, if there is a subset of variables that are linearly dependent, we have to eliminate one of these variables arbitrarily. This raises a very important question: If there is no justification which of these variables to eliminate, and we are thus forced to eliminate one of these variables arbitrarily, is our model outcome arbitrary?

Theorem 1 states that the model outcome is not arbitrary, because it does not matter which of these variables is eliminated, the model outcome will always be the same.

Intuitively we can interpret this proposition as follows: We know that a set of variables is linearly dependent, then taking away one variable does not eliminate any information, because the last variable is automatically implied by all the others.

More formally, we can express our proposition as follows:

Theorem 1: *Consider $n+1$ random variables, X_1, X_2, X_3,..., X_{n+1} for which $X_1+X_2+X_3+\ldots+X_{n+1}=0$. Consider an additional set of random variables, Y_1, Y_2, Y_3,..., Y_n that are not linearly dependent. Let $\mathbf{X_i}$ be a vector containing all $\mathbf{X}$ with the exception of X_i. Let $\mathbf{Y}$ be a vector containing all Y_i. Let $\mathbf{V_i}$ and $\mathbf{V_Y}$ be the variance-covariance matrix of $\mathbf{X_i}$ and $\mathbf{Y}$ respectively and let $\sigma_\mathbf{i}$ be the matrix containing their covariances. Then:*

$$\begin{bmatrix} \mathbf{X_i^T} & \mathbf{Y^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_i} & \sigma_\mathbf{i} \\ \sigma_\mathbf{i}^\mathbf{T} & \mathbf{V_Y} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X_i} \\ \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{X_j^T} & \mathbf{Y^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_j} & \sigma_\mathbf{j} \\ \sigma_\mathbf{j}^\mathbf{T} & \mathbf{V_Y} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X_j} \\ \mathbf{Y} \end{bmatrix} \forall i,j. \quad (7)$$

Lemma 1.1:

Let

$$\mathbf{A} = \begin{bmatrix} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{11}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{12}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & \ldots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} \\ (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{21}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{22}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & \ldots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} \\ \ldots & \ldots & \ldots & \ldots \\ (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n1}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n2}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} & \ldots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} \end{bmatrix}. \quad (8)$$

Then:

$$\mathbf{AX_{n+1}} = \mathbf{V_n^{-\frac{1}{2}}}\mathbf{X_n}. \quad (9)$$

Proof:

$$\mathbf{AX_{n+1}} = \quad (10)$$

$$\begin{bmatrix} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{11}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{12}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & \ldots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} \\ (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{21}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{22}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & \ldots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} \\ \ldots & \ldots & \ldots & \ldots \\ (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n1}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n2}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} & \ldots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \ldots \\ -\sum_{a=1}^{n} X_a \end{bmatrix}.$$

$$= \begin{bmatrix} \sum_{a=1}^{n-1} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1a}} X_a - \sum_{a=1}^{n-1} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} X_a + \sum_{a=1}^{n} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} X_a \\ \sum_{a=1}^{n-1} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2a}} X_a - \sum_{a=1}^{n-1} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} X_a + \sum_{a=1}^{n} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} X_a \\ \ldots \\ \sum_{a=1}^{n-1} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{na}} X_a - \sum_{a=1}^{n-1} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} X_a + \sum_{a=1}^{n} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} X_a \end{bmatrix}.$$

$$= \begin{bmatrix} \sum_{a=1}^{n} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1a}} X_a \\ \sum_{a=1}^{n} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2a}} X_a \\ \ldots \\ \sum_{a=1}^{n} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{na}} X_a \end{bmatrix} = \mathbf{V_n^{-\frac{1}{2}}}\mathbf{X_n}$$

q.e.d.

Lemma 1.2:

$$\mathbf{A}\sigma_{\mathbf{n+1}} = \mathbf{V_n^{-\frac{1}{2}}}\sigma_\mathbf{n}. \quad (11)$$

Proof:

$$\mathbf{A}\sigma_{\mathbf{n+1}} = \begin{bmatrix} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{11}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{12}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & \cdots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} \\ (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{21}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{22}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & \cdots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} \\ \cdots & \cdots & \cdots & \cdots \\ (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n1}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n2}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} & \cdots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} (\sigma_{\mathbf{n}})_{\mathbf{11}} & (\sigma_{\mathbf{n}})_{\mathbf{12}} & \cdots & (\sigma_{\mathbf{n}})_{\mathbf{1m}} \\ (\sigma_{\mathbf{n}})_{\mathbf{21}} & (\sigma_{\mathbf{n}})_{\mathbf{22}} & \cdots & (\sigma_{\mathbf{n}})_{\mathbf{2m}} \\ \cdots & \cdots & \cdots & \cdots \\ -\sum_{a=1}^{n-1}(\sigma_{\mathbf{n}})_{\mathbf{a1}} & -\sum_{a=1}^{n-1}(\sigma_{\mathbf{n}})_{\mathbf{a2}} & \cdots & -\sum_{a=1}^{n-1}(\sigma_{\mathbf{n}})_{\mathbf{am}} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{a=1}^{n}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1a}}(\sigma_{\mathbf{n}})_{\mathbf{a1}} & \sum_{a=1}^{n}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1a}}(\sigma_{\mathbf{n}})_{\mathbf{a2}} & \cdots & \sum_{a=1}^{n}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1a}}(\sigma_{\mathbf{n}})_{\mathbf{am}} \\ \sum_{a=1}^{n}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2a}}(\sigma_{\mathbf{n}})_{\mathbf{a1}} & \sum_{a=1}^{n}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2a}}(\sigma_{\mathbf{n}})_{\mathbf{a2}} & \cdots & \sum_{a=1}^{n}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2a}}(\sigma_{\mathbf{n}})_{\mathbf{am}} \\ \cdots & \cdots & \cdots & \cdots \\ \sum_{a=1}^{n}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{na}}(\sigma_{\mathbf{n}})_{\mathbf{a1}} & \sum_{a=1}^{n}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{na}}(\sigma_{\mathbf{n}})_{\mathbf{a2}} & \cdots & \sum_{a=1}^{n}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{na}}(\sigma_{\mathbf{n}})_{\mathbf{am}} \end{bmatrix} =$$

$$\mathbf{V_n^{-\frac{1}{2}}}\sigma_{\mathbf{n}}.$$

q.e.d.

Lemma 1.3:

$$\mathbf{A^T A} = \mathbf{V_{n+1}^{-1}}. \tag{13}$$

Proof:

$$\mathbf{V_{n+1}A^TA} = \begin{bmatrix} (\mathbf{V_n})_{\mathbf{11}} & (\mathbf{V_n})_{\mathbf{12}} & \cdots & -\sum_{\mathbf{a=1}}^{\mathbf{n}}(\mathbf{V_n})_{\mathbf{1a}} \\ (\mathbf{V_n})_{\mathbf{21}} & (\mathbf{V_n})_{\mathbf{22}} & \cdots & -\sum_{\mathbf{a=1}}^{\mathbf{n}}(\mathbf{V_n})_{\mathbf{2a}} \\ \cdots & \cdots & \cdots & \cdots \\ -\sum_{\mathbf{a=1}}^{\mathbf{n}}(\mathbf{V_n})_{\mathbf{a1}} & -\sum_{\mathbf{a=1}}^{\mathbf{n}}(\mathbf{V_n})_{\mathbf{a2}} & \cdots & \sum_{\mathbf{a=1}}^{\mathbf{n}}\sum_{\mathbf{b=1}}^{\mathbf{n}}(\mathbf{V_n})_{\mathbf{ab}} \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{11}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{21}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & \cdots & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n1}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} \\ (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{12}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{22}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & \cdots & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n2}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} \\ \cdots & \cdots & \cdots & \cdots \\ -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & \cdots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} \end{bmatrix}$$

$$\begin{bmatrix} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{11}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{12}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & \cdots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} \\ (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{21}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{22}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & \cdots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} \\ \cdots & \cdots & \cdots & \cdots \\ (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n1}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n2}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} & \cdots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{a=1}^{n}(\mathbf{V_n})_{\mathbf{1a}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1a}} & \cdots & \sum_{a=1}^{n}(\mathbf{V_n})_{\mathbf{1a}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{na}} \\ \sum_{a=1}^{n}(\mathbf{V_n})_{\mathbf{2a}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1a}} & \cdots & \sum_{a=1}^{n}(\mathbf{V_n})_{\mathbf{2a}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{na}} \\ -\sum_{a=1}^{n}\sum_{b=1}^{n}(\mathbf{V_n})_{\mathbf{ba}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1a}} & \cdots & -\sum_{a=1}^{n}\sum_{b=1}^{n}(\mathbf{V_n})_{\mathbf{ba}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{na}} \end{bmatrix}$$

$$\begin{bmatrix} (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{11}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{12}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} & \dots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{1n}} \\ (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{21}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{22}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} & \dots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{2n}} \\ \dots & \dots & \dots & \dots \\ (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n1}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} & (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{n1}} - (\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} & \dots & -(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{nn}} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{a=1}^{n}\sum_{b=1}^{n}(\mathbf{V_n})_{\mathbf{1a}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{ba}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{b1}} & \dots & 0 \\ \sum_{a=1}^{n}\sum_{b=1}^{n}(\mathbf{V_n})_{\mathbf{2a}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{ba}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{b1}} & \dots & 0 \\ \dots & \dots & \dots \\ -\sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}(\mathbf{V_n})_{\mathbf{ba}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{ca}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{c1}} & \dots & 0 \end{bmatrix}$$

$$- \begin{bmatrix} \sum_{a=1}^{n}\sum_{b=1}^{n}(\mathbf{V_n})_{\mathbf{1a}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{ba}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{bn}} & \dots \\ \sum_{a=1}^{n}\sum_{b=1}^{n}(\mathbf{V_n})_{\mathbf{1a}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{ba}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{bn}} & \dots \\ \dots & \dots \\ -\sum_{a=1}^{n}\sum_{b=1}^{n}\sum_{c=1}^{n}(\mathbf{V_n})_{\mathbf{ba}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{ca}}(\mathbf{V_n^{-\frac{1}{2}}})_{\mathbf{cn}} & \dots \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ -1 & -1 & \dots & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ -1 & -1 & \dots & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} = \mathbf{I}$$

$$\Rightarrow \mathbf{A^T A} = \mathbf{V_{n+1}^{-1}}$$

q.e.d.

Let $\mathbf{V}$ be defined as follows:

$$\mathbf{V} = \mathbf{V_Y} - \sigma_{\mathbf{n+1}}^{\mathbf{T}}\mathbf{V_{n+1}^{-1}}\sigma_{\mathbf{n+1}} \tag{15}$$

Then, the following holds true:

$$\begin{bmatrix} \mathbf{X_{n+1}^T} & \mathbf{Y^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_{n+1}} & \sigma_{\mathbf{n+1}} \\ \sigma_{\mathbf{n+1}}^{\mathbf{T}} & \mathbf{V_Y} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X_{n+1}} \\ \mathbf{Y} \end{bmatrix} \tag{16}$$

$$= \begin{bmatrix} \mathbf{X_{n+1}^T} & \mathbf{Y^T} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{V_{n+1}^{-1}} + \mathbf{V_{n+1}^{-1}}\sigma_{\mathbf{n+1}}\mathbf{V^{-1}}\sigma_{\mathbf{n+1}}^{\mathbf{T}}\mathbf{V_{n+1}^{-1}} & -\mathbf{V_{n+1}^{-1}}\sigma_{\mathbf{n+1}}\mathbf{V^{-1}} \\ -\mathbf{V^{-1}}\sigma_{\mathbf{n+1}}^{\mathbf{T}}\mathbf{V_{n+1}^{-1}} & \mathbf{V^{-1}} \end{bmatrix} \begin{bmatrix} \mathbf{X_{n+1}} \\ \mathbf{Y} \end{bmatrix}$$

$$= \mathbf{X_{n+1}^T V_{n+1}^{-1} X_{n+1}} + (\mathbf{Y} - \sigma_{\mathbf{n+1}}^{\mathbf{T}}\mathbf{V_{n+1}^{-1}}\mathbf{X_{n+1}})^{\mathbf{T}}\mathbf{V^{-1}}(\mathbf{Y} - \sigma_{\mathbf{n+1}}^{\mathbf{T}}\mathbf{V_{n+1}^{-1}}\mathbf{X_{n+1}})$$

$$\mathbf{V} = \mathbf{V_Y} - \sigma_{\mathbf{n+1}}^{\mathbf{T}}\mathbf{V_{n+1}^{-1}}\sigma_{\mathbf{n+1}} = \mathbf{V_Y} - \sigma_{\mathbf{n+1}}^{\mathbf{T}}\mathbf{A^T A}\sigma_{\mathbf{n+1}} = \mathbf{V_Y} - \sigma_{\mathbf{n}}^{\mathbf{T}}\mathbf{V_n^{-1}}\sigma_{\mathbf{n}}$$

$$\Rightarrow \mathbf{X_{n+1}^T V_{n+1}^{-1} X_{n+1}} + (\mathbf{Y} - \sigma_{\mathbf{n+1}}^{\mathbf{T}}\mathbf{V_{n+1}^{-1}}\mathbf{X_{n+1}})^{\mathbf{T}}\mathbf{V^{-1}}(\mathbf{Y} - \sigma_{\mathbf{n+1}}^{\mathbf{T}}\mathbf{V_{n+1}^{-1}}\mathbf{X_{n+1}})$$

$$= \mathbf{X_{n+1}^T A^T A X_{n+1}} + (\mathbf{Y} - \sigma_{\mathbf{n+1}}^{\mathbf{T}} \mathbf{A^T A X_{n+1}})^{\mathbf{T}} \mathbf{V^{-1}} (\mathbf{Y} - \sigma_{\mathbf{n+1}}^{\mathbf{T}} \mathbf{A^T A X_{n+1}})$$

$$= \mathbf{X_n^T V_n^{-1} X_n} + (\mathbf{Y} - \sigma_{\mathbf{n}}^{\mathbf{T}} \mathbf{V_n^{-1} X_n})^{\mathbf{T}} \mathbf{V^{-1}} (\mathbf{Y} - \sigma_{\mathbf{n}}^{\mathbf{T}} \mathbf{V_n^{-1} X_n})$$

$$= \begin{bmatrix} \mathbf{X_n^T} & \mathbf{Y^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_n} & \sigma_{\mathbf{n}} \\ \sigma_{\mathbf{n}}^{\mathbf{T}} & \mathbf{V_Y} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X_n} \\ \mathbf{Y} \end{bmatrix}$$

q.e.d.

2.3.3 Calculating the Elements of V

In order to calculate the variance-covariance matrix $\mathbf{V}$ of $\mathbf{X}$, we need to calculate the variances under the null hypothesis for every element X_c^a that is part of $\mathbf{X}$ and their respective covariances. Recall that we have interpreted X_c^a as a random variable that has been constructed by randomly drawing from the dataset without replacement. We can therefore interpret this problem as a generalization of the hypergeometric distribution.

Theorem 2: *Suppose that X_c^a and X_d^b are random variables that have been constructed by randomly drawing from a dataset without replacement. Then the following holds true:*

$$cov(X_c^a, X_d^b) = \frac{(N\sum_{i=1}^N x_a^i x_b^i - \sum_{i=1}^N x_a^i \sum_{k=1}^N x_b^k)(N\sum_{j=1}^N y_c^j y_d^j - \sum_{j=1}^N y_c^j \sum_{l=1}^N y_d^l)}{N^2(N-1)} \forall a,b,c,d. \tag{17}$$

Proof:

Lemma 2.1:

$$E(x_a^i y_c^j x_b^k y_d^l \mid k \neq i, l \neq j) = \frac{(\sum_{i=1}^N x_a^i \sum_{k=1}^N x_b^k - \sum_{i=1}^N x_a^i x_b^i)(\sum_{j=1}^N y_c^j \sum_{l=1}^N y_d^l - \sum_{j=1}^N y_c^j y_d^j)}{N^2(N-1)^2}. \tag{18}$$

Proof for Lemma 2.1:

$$E(x_a^i y_c^j x_b^k y_d^l \mid k \neq i, l \neq j) = \frac{\sum_{i=1}^N x_a^i \sum_{k=1,k\neq i}^N x_b^k}{N(N-1)} \frac{\sum_{j=1}^N y_c^j \sum_{l=1,l\neq j}^N y_d^l}{N(N-1)} \tag{19}$$

$$= \frac{(\sum_{i=1}^N x_a^i \sum_{k=1}^N x_b^k - \sum_{i=1}^N x_a^i x_b^i)(\sum_{j=1}^N y_c^j \sum_{l=1}^N y_d^l - \sum_{j=1}^N y_c^j y_d^j)}{N^2(N-1)^2}.$$

q.e.d.

$$cov(X_c^a, X_d^b) = E(X_c^a X_d^b) - E(X_d^b)E(X_d^b) \tag{20}$$

$$= N * E(x_a^i y_c^j x_b^i y_d^j)$$

$$+N(N-1)E(x_a^i y_c^j x_b^k y_d^l \mid k \neq i, l \neq j) - E(X_d^b)E(X_d^b)$$

$$= N * \frac{\sum_{i=1}^N x_a^i x_b^i \sum_{j=1}^N y_c^j y_d^j}{N^2}$$

$$+N(N-1)\frac{(\sum_{i=1}^N x_a^i \sum_{k=1}^N x_b^k - \sum_{i=1}^N x_a^i x_b^i)(\sum_{j=1}^N y_c^j \sum_{l=1}^N y_d^l - \sum_{j=1}^N y_c^j y_d^j)}{N^2(N-1)^2}$$

$$-\frac{\sum_{i=1}^N x_a^i \sum_{k=1}^N x_b^k \sum_{j=1}^N y_c^j \sum_{l=1}^N y_d^l}{N^2}$$

$$
\begin{aligned}
&= \frac{N(N-1)\sum_{i=1}^{N} x_a^i x_b^i \sum_{j=1}^{N} y_c^j y_d^j}{N^2(N-1)} \\
&+ \frac{N(\sum_{i=1}^{N} x_a^i \sum_{k=1}^{N} x_b^k - \sum_{i=1}^{N} x_a^i x_b^i)(\sum_{j=1}^{N} y_c^j \sum_{l=1}^{N} y_d^l - \sum_{j=1}^{N} y_c^j y_d^j)}{N^2(N-1)} \\
&- \frac{(N-1)\sum_{i=1}^{N} x_a^i \sum_{k=1}^{N} x_b^k \sum_{j=1}^{N} y_c^j \sum_{l=1}^{N} y_d^l}{N^2(N-1)} \\
&= \frac{N^2 \sum_{i=1}^{N} x_a^i x_b^i \sum_{j=1}^{N} y_c^j y_d^j - N \sum_{i=1}^{N} x_a^i x_b^i \sum_{j=1}^{N} y_c^j \sum_{l=1}^{N} y_d^l}{N^2(N-1)} \\
&+ \frac{-N \sum_{i=1}^{N} x_a^i \sum_{k=1}^{N} x_b^k \sum_{j=1}^{N} y_c^j y_d^j + \sum_{i=1}^{N} x_a^i \sum_{k=1}^{N} x_b^k \sum_{j=1}^{N} y_c^j \sum_{l=1}^{N} y_d^l}{N^2(N-1)} \\
&= \frac{(N \sum_{i=1}^{N} x_a^i x_b^i - \sum_{i=1}^{N} x_a^i \sum_{k=1}^{N} x_b^k)(N \sum_{j=1}^{N} y_c^j y_d^j - \sum_{j=1}^{N} y_c^j \sum_{l=1}^{N} y_d^l)}{N^2(N-1)}.
\end{aligned}
$$

q.e.d.

Note that this formula is also applicable to cases where $a = b$ and/or $c = d$. When $a = b$ and $c = d$ the above formula is equivalent to calculating the variance of X_c^a. Note further that in the special case in which x_a^i and y_c^j can only assume the values 0 or 1, a equals b and c equals d, this formula reduces to the standard formula for the variance of the hypergeometric distribution. When a equals b *or* c equals d, it reduces to the standard formula for the covariance of the hypergeometric distribution. This is consistent with our interpretation of the problem as a generalization of the hypergeometric distribution. Finally, note that this is not an estimate of the variance-covariance matrix under the null hypothesis, but that it actually is the variance-covariance matrix under the null hypothesis.

2.3.4 Evaluating the Statistical Significance Corrected for Other Predictive Methods

Theorem 3: *Suppose* $\mathbf{X_1}$ *and* $\mathbf{X_2}$ *are two multivariate random variables of length* n_1 *and* n_2 *respectively. Further suppose that their expected values are* 0*, their variance-covariance-matrices are* $\mathbf{V_1}$ *and* $\mathbf{V_2}$ *respectively and their mutual covariance matrix is* σ_{12}*.*

$$
\begin{bmatrix} \mathbf{X_1^T} & \mathbf{X_2^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_1} & \sigma_{12} \\ \sigma_{12}^{\mathbf{T}} & \mathbf{V_2} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \end{bmatrix} - \mathbf{X_1^T V_1^{-1} X_1}. \tag{21}
$$

Then (21) is a squared Mahalanobis distance. Furthermore, assume that $\mathbf{X_1}$ *and* $\mathbf{X_2}$ *are multivariate normal. Then the following holds true:*

$$
\begin{bmatrix} \mathbf{X_1^T} & \mathbf{X_2^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_1} & \sigma_{12} \\ \sigma_{12}^{\mathbf{T}} & \mathbf{V_2} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \end{bmatrix} - \mathbf{X_1^T V_1^{-1} X_1} \sim \chi^2(n_2). \tag{22}
$$

Proof:

Define $\mathbf{V_{12}}$ as follows:

$$
\mathbf{V_{12}} := \mathbf{V_2}\text{-}\sigma_{12}^{\mathbf{T}} \mathbf{V_1^{-1}} \sigma_{12}. \tag{23}
$$

Then $\begin{bmatrix} \mathbf{X_1^T} & \mathbf{X_2^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_1} & \sigma_{12} \\ \sigma_{12}^{\mathbf{T}} & \mathbf{V_2} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \end{bmatrix}$ can be reduced as follows:

$$\begin{bmatrix} \mathbf{X_1^T} & \mathbf{X_2^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_1} & \sigma_{12} \\ \sigma_{12}^{\mathbf{T}} & \mathbf{V_2} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \end{bmatrix} = \tag{24}$$

$$\begin{bmatrix} \mathbf{X_1^T} & \mathbf{X_2^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_1^{-1}} + \mathbf{V_1^{-1}}\sigma_{12}\mathbf{V_{12}^{-1}}\sigma_1^{\mathbf{T}}\mathbf{V_1^{-1}} & -\mathbf{V_1^{-1}}\sigma_{12}\mathbf{V_{12}^{-1}} \\ -\mathbf{V_{12}^{-1}}\sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}} & \mathbf{V_{12}^{-1}} \end{bmatrix} \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \end{bmatrix}$$

$$= \mathbf{X_1^T V_1^{-1} X_1} + \mathbf{X_1^T V_1^{-1}}\sigma_{12}\mathbf{V_{12}^{-1}}\sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}X_1}$$

$$-\mathbf{X_2^T V_{12}^{-1}}\sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}X_1} - \mathbf{X_1^T V_1^{-1}}\sigma_{12}\mathbf{V_{12}^{-1}X_2} + \mathbf{X_2^T V_{12}^{-1}X_2}$$

$$= \mathbf{X_1^T V_1^{-1} X_1} + (\mathbf{X_2} - \sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}X_1})^{\mathbf{T}}\mathbf{V_{12}^{-1}}(\mathbf{X_2} - \sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}X_1}).$$

Define $\mathbf{X_{12}}$ as follows:

$$\mathbf{X_{12}} = \mathbf{X_2} - \sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}X_1}. \tag{25}$$

Then $E(\mathbf{X_{12}^T X_{12}})$can be written as follows:

$$E(\mathbf{X_{12}X_{12}^T}) = E((\mathbf{X_2} - \sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}X_1})(\mathbf{X_2} - \sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}X_1})^{\mathbf{T}}) \tag{26}$$

$$= E(\mathbf{X_2X_2^T}) - E(\sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}X_1X_2^T}) - E(\mathbf{X_2X_1}^{\mathbf{T}}\mathbf{V_1^{-1}}\sigma_{12}) + E(\sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}X_1X_1}^{\mathbf{T}}\mathbf{V_1^{-1}}\sigma_{12})$$

$$= \mathbf{V_2} - 2\sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}}\sigma_{12} + \sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}V_1V_1^{-1}}\sigma_{12} = \mathbf{V_2} - \sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}}\sigma_{12} = \mathbf{V_{12}}$$

Therefore $\mathbf{X_{12}^T V_{12}^{-1} X_{12}} = (\mathbf{X_2} - \sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}X_1})^{\mathbf{T}}\mathbf{V_{12}^{-1}}(\mathbf{X_2} - \sigma_{12}^{\mathbf{T}}\mathbf{V_1^{-1}X_1})$ is a squared Mahalanobis distance. Combining (24), (25) and (26), we get (21):

$$\begin{bmatrix} \mathbf{X_1^T} & \mathbf{X_2^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_1} & \sigma_{12} \\ \sigma_{12}^{\mathbf{T}} & \mathbf{V_2} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \end{bmatrix} = \mathbf{X_1^T V_1^{-1} X_1} + \mathbf{X_{12}^T V_{12}^{-1} X_{12}} \tag{27}$$

$$\Leftrightarrow \mathbf{X_{12}^T V_{12}^{-1} X_{12}} = \begin{bmatrix} \mathbf{X_1^T} & \mathbf{X_2^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_1} & \sigma_{12} \\ \sigma_{12}^{\mathbf{T}} & \mathbf{V_2} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \end{bmatrix} - \mathbf{X_1^T V_1^{-1} X_1}$$

Finally, if we assume $\mathbf{X_1}$ and $\mathbf{X_2}$ to be normally distributed, (22) is proven as well:

$$\Leftrightarrow \mathbf{X_{12}^T V_{12}^{-1} X_{12}} = \begin{bmatrix} \mathbf{X_1^T} & \mathbf{X_2^T} \end{bmatrix} \begin{bmatrix} \mathbf{V_1} & \sigma_{12} \\ \sigma_{12}^{\mathbf{T}} & \mathbf{V_2} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \end{bmatrix} - \mathbf{X_1^T V_1^{-1} X_1} \sim \chi^2(n_2) \tag{28}$$

q.e.d.

There is a simple interpretation to Theorem 3: The difference between the two chi-squared distributions is itself chi-squared distributed and can be interpreted as the statistical significance of $\mathbf{X_2}$ corrected for $\mathbf{X_1}$. Note that this result is non-trivial as the difference between two chi-squared distribution is generally not chi-squared distributed.

2.3.5 Monte Carlo Sampling

In cases where the assumption of a normal distribution is is too strong, we reinterpret the test statistics defined in equations (6) and (22) as the squared Mahalanobis distance of the observation from the expected value under the null hypothesis. Essentially, a chi-squared test is designed to calculate the probability that a random procedure would produce an observation with a Mahalanobis distance that is greater than or equal to a given value.

In the case of our framework, the chi-squared test statistic and the associated p-value can be interpreted as an approximation to a Monte Carlo procedure, in which the data is randomly reshuffled. On the basis of this interpretation, we construct a new test based on Monte Carlo sampling: For this experiment, we reshuffle the values y_c^j randomly such that $y_c^{j_{new}} = y_c^{j_{old}}$ and $y_d^{j_{new}} = y_d^{j_{old}}$ for all c, d .

We now know that the null hypothesis (H_0) is true and calculate the test statistic as defined in equation (6) for each of the different predictive methods. We repeat this procedure a large number of times and record the test statistics generated.

Because the Monte Carlo reshuffling and the original hypothesis test are based on the same mathematical consideration and the p-value associated with the chi-squared test statistic can be interpreted as an approximation to a Monte Carlo simulation, the two p-values are directly comparable and it is possible to directly quantify the impact of a possible distortion.

Note that $E(X_c^a)$ and $\mathbf{V}$ are unaffected by the random reshufflings. We only need to recalculate X_c^a as defined in equation (1). Since this is computationally very cheap, the Monte Carlo method is feasible, even for large datasets and a large number of iterations.

Finally, we can calculate a credible interval in which the "true" p-value is likely to be located based on the beta distribution as the conjugate prior to the binomial and the assumption of a flat prior.

2.3.6 Application of the framework to assess the three requirements

Applying the framework to assess the first requirement is achieved by a straight-forward application of equations (1), (2), (6) and (7).

The second requirement can be examined by slightly modifying the approach defined above: We want to assess whether the difference in the test statistic for X_c^a and X_c^b is statistically significant, in other words, whether there is a statistically significant difference in the predictive accuracy of predictive methods a and b. As we demonstrated in (4), our test statistic is directly proportional to Pearson's r. Testing whether there is a statistically significant difference between the calculated test statistics of two different is identical to testing whether there is a statistically significant difference between their respective values for Pearson's r. Thus, we calculate $\frac{x_a^i y_c^i - E(X_c^a)}{\sqrt{V(X_c^a)}}$ and $\frac{x_b^i y_c^i - E(X_c^b)}{\sqrt{V(X_c^b)}}$ for all i and create a a set of values $x_{a^T}^i$ of length $N^T = 2N$ containing $\frac{x_a^i y_c^i - E(X_c^a)}{\sqrt{V(X_c^a)}}$ and $\frac{x_b^i y_c^i - E(X_c^b)}{\sqrt{V(X_c^b)}}$ for all i. We then create a dummy variable $y_{c^T}^i$ that assumes the value of 1 if $x_{a^T}^i$ has been generated by X_c^b and 0 otherwise. We can then apply equations (1), (2), (6) and (7) to N^T, $x_{a^T}^i$ and $y_{c^T}^i$ in the same manner as proposed above. We effectively test whether there is a statistically significant correlation between the set of values $x_{a^T}^i$ and $y_{c^T}^i$ which is the case if only if there is a statistically significant difference

between $\frac{x_a^i y_c^i - E(X_c^a)}{\sqrt{V(X_c^a)}}$ and $\frac{x_b^i y_c^i - E(X_c^b)}{\sqrt{V(X_c^b)}}$.

The third requirement can be examined using equations (1), (2), (6), (7) and (22). We want to assess whether a predictive method generates statistically significant predictions when corrected for a set of other predictive methods. We first calculate the test statistic as defined in equation (7) for the combined predictions of the new predictive model(s) and the predictive model(s) we would like to correct for. We then do the same for only the predictive model(s) we would like to correct for. By equation (22), the difference between the two test statistics is itself chi-squared distributed under the null hypothesis.

2.3.7 Generalization of the approach to panel datasets

Consider the following two business cases, both of which are important applications of predictive analytics:

1) Customer retention prediction: A company has a set of customers that it observes over a period of time. Using accumulated transaction data accumulated, the task is to predict whether the customer is going to switch to another company.

2) Bankruptcy prediction: Given balance sheet data on a set of companies observed over a period of time, the task is to predict whether a company is going to file for bankruptcy.

Both of these problems are examples of panel data. In these settings, the probability of a customer switching to another company or the probability of a company filing for bankruptcy at time t cannot reasonably be assumed to be uncorrelated to the probability at time t+1.

However, one of the assumptions of the basic framework introduced in this study is that the predictions are identically independently distributed. We therefore modify the framework to be suitable for panel data settings.

Mathematically, the approach is straightforward: Assuming that the rows in the panel are serially correlated, but columns are not, let $x_a^{r_1,i}$ denote the i^{th} value in row r_1 of the dataset associated with variable X_a. Let $y_c^{r_2,j}$ denote the j^{th} value in row r_2 of the dataset associated with variable Y_c. Then, we can calculate $\mathbf{V}$ as follows:

$$cov(X_c^a, X_d^b) = cov(\sum_{r_1}\sum_{i=1}^{N} x_a^{r_1,i} y_c^{r_1,i}, \sum_{r_2}\sum_{i=1}^{N} x_b^{r_2,i} y_d^{r_2,i}) \tag{29}$$

$$= \sum_{r_1}\sum_{r_2} cov(\sum_{i=1}^{N} x_a^{r_1,i} y_c^{r_1,i}, \sum_{i=1}^{N} x_b^{r_2,i} y_d^{r_2,i}).$$

Equation (7) is readily applicable to the calculation of the row-wise covariances in (29). The expected value is calculated as follows:

$$E(X_c^a) = \sum_{r_1} E(\sum_{i=1}^{N} x_a^{r_1,i} y_c^{r_1,i}) = \sum_{r_1} \frac{\sum_{i=1}^{N} x_a^{r_1,i} \sum_{j=1}^{N} y_c^{r_1,i}}{N} \forall a, c \tag{30}$$

The Monte Carlo sampling approach is can also be applied to panel data: Since we have assumed that there is at least one dimension for which there is no serial correlation, we can randomly reshuffle the data along that dimension while keeping the orders for the other dimensions intact: If, for instance, we have generated predictions for a set of companies over a given period of time, we can reshuffle predictions for individual companies while leaving the order of the time dimension intact: Our prediction for company A at time t_1 could become a prediction for company B at time t_1 as a result of the reshuffling, but it could not become a prediction for either company A or B at time t_2.

2.3.8 Accounting for missing data

So far, the framework proposed above is unrealistic. We have implicitly assumed that data on all customers and all companies is available for the entire observation period. In practise, this is not the case: Over the observation period, new customers could have been acquired and others have switched to another company. Likewise, taking the example of bankruptcy prediction, balance sheet data is not usually not available after the company has gone bankrupt. This implies that we need to develop an approach that is robust to missing data.

This can be easily achieved by further modifying the framework in the manner already discussed with respect to our second question: We simply define a set of variables $x^i_{a'}$ which contain all $\frac{x^i_a y^i_c - E(X^a_c)}{\sqrt{V(X^a_c)}}$ (or another measure of accuracy such as the squared prediction error) for both the predictive method and a naive benchmark (such as the average of the predicted value in the training set). $x^i_{a'}$ is thus a measure of the predictive accuracy x^i_a or the benchmark have for y^i_c. We then create a dummy variable $y^i_{c'}$ that assumes the value of 1 if $x^i_{a'}$ has been generated by X^b_c and 0 if it has been generated by the benchmark.

We let all fields for which we have no data because the respective trips have already ended assume the value of 0. Since we do this for both the predictive method as well as the benchmark, this will not distort our results.

We can now apply the framework to this dataset by slightly modifying requirements 1 and 3:

Requirement 1 (modified): The predictive methods outperform a naive benchmark prediction in a statistically significant manner.

Requirement 3 (modified): When corrected for the outperformances of the predictions generated by alternative methods, the predictive method outperforms a naive benchmark prediction in a statistically significant manner.

We effectively test whether there is a statistically significant correlation between the set of values $x^i_{a'}$ and $y^i_{c'}$ which is the case if and only if there is a statistically significant difference between the predictive performance of our predictive method x^i_a and the benchmark. Note that is not necessary to assume stationarity, because the expected values are calculated for every individual row.

2.4 Methods

We implemented the framework in C++ and developed an interface to Python using Cython. We generated the pseudo-random numbers needed for the Monte Carlo simulation using the Mersenne twister originally developed by Matsumoto and Nishimura (1998) as implemented in the Boost C++ library. The Mersenne twister has been demonstrated to be a highly effective pseudo-random number generator, passing the so-called diehard tests (Matsumoto and Nishimura, 1998).

To illustrate how our framework can be used to generate additional insight, we used the dataset provided by Serrano-Cinca and Gutiérrez-Nieto (2013) for the purposes of this study.

We then compared our own framework to standard methods of evaluating predictive methods that are based simply on the concept of outperformance using the data provided by the authors.

	RMSE	t	p-value
Linear Discriminant Analysis (full)	0.1915	-0.06187	0.9507
Logistic Regression (full)	0.1918	-0.7414	0.4585
Logistic Regression (step-wise)	0.1964	-1.879	0.06031*
Multilayer Perceptron	0.2075	-1.803	0.07135*
K-Nearest Neighbour	0.2175	-0.1584	0.8742
Linear Discriminant Analysis (step-wise)	0.2184	-0.1446	0.8850
Logistic Regression	0.2192	-0.08245	0.9343
Bagging (C4)	0.2197	-0.2591	0.7955
Linear Discriminant Analysis	0.2212	-0.6330	0.5267
Naive Bayes	0.2252	-2.429	0.01516**
Bagging (Random Tree)	0.2400	-1.737	0.08243*
Boosting (Random Tree)	0.2484	-1.769	0.07696*
AdaBoost (C4)	0.2553	-5.451	5.12e-08***
Decision Tree (C4)	0.2862	-1.102	0.2706
Support Vector Machine	0.2917	-2.725	0.006437
Decision Tree (Random Tree)	0.3072	-29.45	1.21e-181***
Partial Least Squares Discriminant Analysis	0.4381	-	-

Note: *p < 0.1, **p < 0.05, ***p < 0.01

TABLE 4: Results for Serrano-Cinca and Gutiérrez-Nieto 2013 (t-test)

2.5 Application

Serrano-Cinca and Gutiérrez-Nieto (2013) analyze balance sheet data for bankruptcy prediction. Their main contribution to the literature is introducing partial least squares discriminant analysis and analyzing the correlation between predictions of a set of selected algorithms using cluster analysis, principal component analysis and by calculating the correlation coefficients. This approach is akin to the third requirement proposed in this study, but their methodology does not generate insight into the statistical significance of their findings.

To illustrate the usefulness of our framework, we compare it to standard methods of evaluating predictive methods, based on our survey of the relevant literature conducted above: We calculated the RMSE based on the probabilistic out-of-sample predictions generated by each of the algorithms used in the original study. We sorted the algorithms in order of their predictive performance as measured by RMSE and used Welch's t-test to determine whether the difference of the squared prediction errors for each of these algorithms in comparison to the next best algorithm is statistically significant. Results are reported in Table 4.

We find that the outperformance of each of the algorithms in comparison to the next best algorithm is not statistically significant for most cases. With the exception of partial least squares discriminant analysis, the differences between the RMSEs are not large. However, this analysis does not give us any indication about the absolute predictive accuracy: We do

	r	χ^2 (p- value)	95%-credible interval
Combination of all models	-	1565.88 (0.00***)	(0.00, 0.00)
Logistic Regression (full)	0.4234	1436.34 (0.00***)	(0.00, 0.00)
Logistic Regression (step-wise)	0.4226	1431.08 (0.00***)	(0.00, 0.00)
Linear Discriminant Analysis (full)	0.4098	1345.63 (0.00***)	(0.00, 0.00)
Multilayer Perceptron	0.4089	1339.53 (0.00***)	(0.00, 0.00)
Logistic Regression	0.3924	1233.74 (0.00***)	(0.00, 0.00)
Naive Bayes	0.3888	1210.92 (0.00***)	(0.00, 0.00)
Support Vector Machine	0.3740	1120.51 (0.00***)	(0.00, 0.00)
Linear Discriminant Analysis (step-wise)	0.3711	1103.6 (0.00***)	(0.00, 0.00)
K-Nearest Neighbour	0.3670	1079.45 (0.00***)	(0.00, 0.00)
Bagging (C4)	0.3539	1003.64 (0.00***)	(0.00, 0.00)
Linear Discriminant Analysis	0.3529	997.806 (0.00***)	(0.00, 0.00)
AdaBoost (C4)	0.3312	878.728 (0.00***)	(0.00, 0.00)
Boosting (Random Tree)	0.3296	870.471 (0.00***)	(0.00, 0.00)
Bagging (Random Tree)	0.3164	801.997 (0.00***)	(0.00, 0.00)
Decision Tree (C4)	0.2482	493.528 (0.00***)	(0.00, 0.00)
Decision Tree (Random Tree)	0.2203	388.965 (0.00***)	(0.00, 0.00)
Partial Least Squares Discriminant Analysis	0.1766	249.779 (0.00***)	(0.00, 0.00)

Note: *p < 0.1, **p < 0.05, ***p < 0.01

TABLE 5: Results for Serrano-Cinca and Gutiérrez-Nieto 2013: Accuracy of Predictive Method.

not know whether the algorithms are equally good or equally poor: All we can say is that their predictive accuracy is better than partial least squares discriminant analysis. This might be because partial least squares discriminant analysis is a particularly poor algorithm, whereas the others are mediocre, or it might be because all of the algorithms are highly accurate, but partial least squares discriminant analysis is less so. Moreover, we find that partial least squares discriminant analysis, the algorithm originally introduced by the authors, is by far the algorithm with the worst predictive accuracy.

These results would therefore lead us to conclude that the algorithm should not be used for bankruptcy prediction as Serrano-Cinca and Gutiérrez-Nieto (2013) propose.

We then contrast these results with the insights gained from our own framework: We calculate Pearson's r to measure the predictive performance of each of the algorithms and sort the algorithms accordingly. As demonstrated in (4), given any sample size N, the proposed test statistic is directly proportional to Pearson's r, therefore sorting by Pearson's r is equivalent to sorting by statistical significance according to the proposed test.

We calculate whether the combined and individual predictive accuracies (first requirement) and whether the difference in Pearson's r between an algorithm and the next best algorithm,

	r	χ^2 (p- value)	95%-credible interval
Logistic Regression (full)	0.4234	2.236e-03 (0.988)	(0.987, 0.989)
Logistic Regression (step-wise)	0.4226	6.395e-02 (0.800)	(0.796, 0.801)
Linear Discriminant Analysis (full)	0.4098	3.423e-03 (0.985)	(0.984, 0.986)
Multilayer Perceptron	0.4089	0.1117 (0.738)	(0.737, 0.743)
Logistic Regression	0.3924	5.765e-03 (0.939)	(0.937, 0.940)
Naive Bayes	0.3888	9.943e-02 (0.753)	(0.749, 0.755)
Support Vector Machine	0.3740	3.910e-03 (0.950)	(0.948, 0.951)
Linear Discriminant Analysis (step-wise)	0.3711	8.231e-03 (0.929)	0.926, 0.930)
K-Nearest Neighbour	0.3670	8.532e-02 (0.771)	(0.765, 0.770)
Bagging (C4)	0.3539	5.376e-04 (0.982)	(0.980, 0.981)
Linear Discriminant Analysis	0.3529	0.2658 (0.606)	(0.601, 0.607)
AdaBoost (C4)	0.3312	1.473e-03 (0.969)	(0.968, 0.970)
Boosting (Random Tree)	0.3296	0.1037 (0.747)	(0.742, 0.748)
Bagging (Random Tree)	0.3164	3.014 (0.082*)	(0.083, 0.086)
Decision Tree (C4)	0.2482	0.5977 (0.439)	(0.437, 0.443)
Decision Tree (Random Tree)	0.2203	2.376 (0.123)	(0.121, 0.125)
Partial Least Squares Discriminant Analysis	0.1766	-	-

Note: *p < 0.1, **p < 0.05, ***p < 0.01

TABLE 6: Results for Serrano-Cinca and Gutiérrez-Nieto 2013: Statistical significance at which predictive method outperforms next best method.

as measured by Pearson's r, is statistically significant (second requirement). We also evaluate the statistical significance of every algorithm using out-of-sample predictions generated by all other algorithms as control variables (third requirement).

For each of the requirements, we calculate the p-value both on the basis of the chi-squared distribution as well as the Monte Carlo simulation. For the latter, we provide the 95%-credible interval, that is the interval in which the "true" p-value is likely to be located. Results are reported in Tables 5, 6 and 7.

Based on the results presented in Table 5, we find that the out-of-sample predictions generated by all algorithms are statistically highly significant. We can therefore conclude that Partial Least Squares Discriminant Analysis, despite being the worst measure both by RMSE and Pearson's r, generates statistically highly significant out-of-sample predictions and that this problem is predictable.

The results presented in Table 6, are most similar to results generated from more standard methods of analysis: We find that the differences in Pearson's r are not statistically significant for most cases (as we found that the differences in RMSE are mostly not significant either) and therefore conclude that we shouldn't place to much emphasis on our finding that the order of predictive accuracy according to RMSE is different thant the order of predictive

	r	χ^2 (p-value)	95%-credible interval
Logistic Regression (full)	0.4234	8.830 (2.963e-03***)	(4.381e-03, 5.238e-03)
Logistic Regression (step-wise)	0.4226	5.253 (2.190e-02**)	(2.084e-02, 2.264e-02)
Linear Discriminant Analysis (full)	0.4098	10.09 (1.490e-03***)	(1.583e-03, 2.115e-03)
Multilayer Perceptron	0.4089	2.746 (9.747e-02*)	(9.380e-02, 9.745e-02)
Logistic Regression	0.3924	2.47771 (0.115471)	(0.1058, 0.1097)
Naive Bayes	0.3888	19.66 (9.245e-06***)	(2.422e-06, 5.571e-05)
Support Vector Machine	0.3740	5.244 (2.208e-02**)	(2.217e-02, 2.403e-02)
Linear Discriminant Analysis (step-wise)	0.3711	3.445 (6.344e-02*)	(6.254e-02, 6.557e-02)
K-Nearest Neighbour	0.3670	2.891e-02 (0.8649)	(0.8639, 0.8681)
Bagging (C4)	0.3539	1.8407 (0.1749)	(0.1714, 0.1761)
Linear Discriminant Analysis	0.3529	0.140574 (0.7077)	(0.7044, 0.7101)
AdaBoost (C4)	0.3312	1.367 (0.2424)	(0.2373, 0.2426)
Boosting (Random Tree)	0.3296	1.987 (0.1586)	(0.1536, 0.1581)
Bagging (Random Tree)	0.3164	0.4585 (0.4983)	(0.4958, 0.5020)
Decision Tree (C4)	0.2482	3.002 (8.315e-02*)	(8.086e-02, 8.427e-02)
Decision Tree (Random Tree)	0.2203	5.254 (2.19e-02**)	(2.084, 2.264)
Partial Least Squares Discriminant Analysis	0.1766	7.116 (7.639e-03***)	(8.143e-03, 9.295e-03)

Note: *p < 0.1, **p < 0.05, ***p < 0.01

TABLE 7: Results for Serrano-Cinca and Gutiérrez-Nieto 2013: Accuracy of predicted method corrected for other methods.

accuracy according to Pearson's r.

Finally, from the results presented in Table 7, we find that the out-of-sample predictions for some of the algorithms used in the study are still significant even when corrected for all of the other algorithms used. These algorithms are not necessarily the strongest overall performers. In fact, partial least squares discriminant analysis, the algorithm originally introduced to the literature by Serrano-Cinca and Gutiérrez-Nieto (2013), is the weakest performer of all of the algorithms considered, both in terms of RMSE and Pearson's r. However, unlike most other algorithms, its out-of-sample predictions remain statistically significant at the 1% level when corrected for all other algorithms.

These findings imply that partial least squares discriminant analysis is able to compensate for weaknesses of other algorithms and thus constitutes a contribution to the literature despite being the weakest overall performer.

2.6 Discussion

In this study, we proposed a unified statistical framework for evaluating predictive methods in IS research that addresses several shortcomings in the prior predictive analytics literature. By incorporating the concepts of overall predictive accuracy, outperformance and forecast

encompassing into a single, coherent framework, we contribute to a more effective demonstration of results in the predictive analytics literature.

The usefulness of the framework has been illustrated using data from a previous study. Serrano-Cinca and Gutiérrez-Nieto (2013) introduced a new method (partial least squares discriminant analysis) to the bankruptcy prediction literature and analyzed the correlation between predictions of a set of selected algorithms. Using more established of evaluation common in the IS literature, we were not able to properly assess the predictive accuracy of the algorithms. Based on the results, we could only conclude that partial least squares discriminant analysis is the worst performer of all the algorithms considered. The tests did not allow assessing the overall predictive accuracy of the algorithms. However, using the framework proposed in this study, we were able to show that partial least squares discriminant analysis is a valuable contribution to the literature, more effectively demonstrating the author's contribution than the authors themselves. We proved that the predictions generated by the algorithm are statistically highly significant. We then showed that the predictions remain statistically significant at the 1% level, even when corrected for all of the other algorithms considered in the original study. This led us to conclude that partial least squared discriminant analysis is a valuable contribution to the bankruptcy prediction literature because it is relatively uncorrelated to other algorithms and compensates for their weaknesses.

For decision makers interested in bankruptcy predictions such as investors or regulatory agencies, these results indicate, that partial least squares discriminant analysis should be used as part of a set of predictive methods used to monitor financial institutions. Hypothesis tests commonly used in the IS literature for evaluating predictive methods would have strongly suggested otherwise. This illustrates that the framework can be used to present IS research in a more rigorous and effective manner. Moreover, the statistical test we developed does not assume data to be normally distributed. Thus, it alleviates future research from making this critical and often not accurate assumption.

The most important limitation of the framework is its lack of applicability to single time series. If the tasks is to predict one time series, such as the S&P 500 or the EUR/USD exchange rate, the framework cannot be applied. If, however, the task is to predict a larger number of time series (such as the 500 stocks than form the S&P 500), the panel data model can be applied as discussed above. We believe that this limitation is not very relevant to IS research.

We also stress the point that the requirements a predictive method has to pass strongly depend on the business context: In contexts, where decision makers have to rely on only a single predictive method, especially where decisions are made by a computer program, our line of reasoning, which is appropriate for bankruptcy prediction, may not be applicable.

However, in such contexts, in may be useful to demonstrate that the introduced method outperforms and is not encompassed by the state-of-the-art. If a simple combination of extant predictive methods achieve the same results as the new proposed method, then the new proposed method may not be considered to be a contribution to the literature.

A more difficult test to pass is to demonstrate that the new predictive method does not only outperform but also encompasses the state-of-the-art: This implies that it is not only better than the state-of-the-art, but that their is no subset of the dataset in which any state-of-the-art method can provide additional information. Such a finding indicates a very significant contribution.

Based on these considerations, we conclude that the concept of forecast encompassing is very useful to assess and demonstrate the contribution of predictive methods rigorously and effectively. Due to the theoretical advantages of the framework proposed in this study and its coherence, we believe that it constitutes a useful contribution for evaluating predictive methods in the IS literature.

Our framework can be easily applied to large-scale distributed memory architectures (Big Data): The sufficient statistics needed to calculate the test statistics given above are simple sums over the dataset. Using a message-passing interface, Apache Hadoop or Apache Spark, the framework can therefore be easily extended to such settings.

2.7 Conclusion

The objective of this study was to introduce a comprehensive framework for evaluating predictive methods particularly with regards to the concept of forecast encompassing. We used a dataset from the previous IS literature to illustrate how our framework can be used to generate additional insight to concepts traditionally used in the IS literature. Because of its versatility the framework could be of interest to most researchers in the field of predictive analytics. In addition, researchers could make use of the theoretical benefit of the framework, namely that it provides a "fail-safe" option for cases in which the assumption of a normal distribution is not applicable. This issue has been all but ignored in previous IS literature. Most importantly, we hope that this study is seen as an encouragement to use the additional concepts to evaluate predictive methods in order to demonstrate contributions more effectively and rigorously.

3 Predicting Product Returns in E-Commerce: The Contribution of Mahalanobis Feature Extraction

Abstract

Product returns are a major challenge in e-commerce that severely affect the economic and ecological sustainability of the industry. While many static one-size-fits-all approaches to limit product returns have been proposed, there is a gap in the literature regarding strategies based on individual consumption patterns. We introduce a decision support system for the prediction of product returns, including a new approach for large-scale feature extraction. This system can be used as the basis for a returns strategy that allows online retailers to intervene before problematic transactions even take place. Using a dataset containing 1,149,262 purchases obtained from a major German online retailer, we demonstrate that our decision support system can identify consumption patterns associated with a high product return rate at sufficient accuracy for such a strategy to be feasible. We also show that the system outperforms a wide selection of state-of-the art classification and dimensionality reduction algorithms.

Keywords: Machine learning, e-commerce, online retail, product returns, statistical methods, business intelligence, decision support systems,

3.1 Introduction

Even though e-commerce has become an important industry that is growing rapidly, many online retailers fail to be profitable (Rigby, 2014). One important reason for this are product returns, which constitute a considerable cost factor. A significant portion of online retailers report that lowering the rate of product returns by 10% could increase profitability by over 20% (Pur et al., 2013). The overall return rate varies, but for online retailers specializing in fashion, it is often higher than 50% of all purchases (Asdecker, 2015).

Product returns are also an important factor contributing to the carbon footprint of e-commerce as a business model. Several studies have evaluated whether e-commerce is environmentally friendlier than traditional retailers, with inconclusive results (Fuchs, 2008; Weber et al., 2008; Williams and Tagami, 2002). It is widely recognized that the "last mile" of the delivery chain (delivering the product to the customer's doorstep) is the most energy intensive (Browne et al., 2008; Halldórsson et al., 2010; Song et al., 2009). Therefore, reducing product returns will have significant impact on the sustainability of e-commerce, both economically and ecologically.

However, simply prohibiting product returns is not an option. Many countries require online retailers to give their customers the right to return products within a certain period of time after purchase. For instance, the minimum required by law in all member states of the European Union is 14 days.[2] In addition, product returns are inherently part of online retailers' business model. Theoretical and empirical studies have demonstrated that a more lenient return policy is associated with a positive impact on customer satisfaction (Cassill,

[2]Directive 2011/83/EU of the European Parliament and of the Council of 25 October 2011, http://eur-lex.europa.eu/Surpriser/Surpriser.do?uri=OJ:L:2011:304:0064:0088:en:PDF, retrieved 2015-03-16

1998), purchase rates (Wood, 2001), future buying behavior (Petersen and Kumar, 2009) or customers' emotional responses (Suwelack et al., 2011). Many researchers have tried to identify the optimal rate of return (Ketzenberg and Zuidwijk, 2009; Padmanabhan and Png, 1997), in other words the return policy that offers the ideal trade-off between the costs associated with product returns and the beneficial impact they have.

In this study, we take a different approach: There is a gap in the current literature on product returns which focuses on optimizing a static one-size-fits-all (i.e. Bonifield et al. 2010; Petersen and Kumar 2009) return policy rather than a dynamic, customer-specific return strategy (Walsh et al., 2014). We fill this gap by proposing a system of prediction and targeted intervention that is able to identify consumption patterns associated with an extremely high rate of product returns and prevent such transactions from taking place. Such consumption patterns might be associated with fraud or impulse shopping. Fraud (such as ordering an item of clothing, wearing it for a special occasion and then returning it to the retailer) has been identified as an important factor for product returns (Wachter et al., 2012) with about 8% of all product returns estimated to be fraudulent (Speights and Hilinski, 2005). In addition, online shopping has been shown to encourage impulsive or even compulsive consumption patterns, which leads to increased return rates (LaRose, 2001). In the absence of a strategy targeting customers who excessively return products, they will effectively be cross-subsidized by those who are more responsible.

We envision a system that is able to assess the likelihood of a product return in real-time while customers put together their shopping basket. Before they even hit the "order" button, the system will be able to predict which products are most likely to be returned. If the likelihood of a product return is too high, the system can intervene. Such a strategy is known as demarketing (Kotler and Levy, 1971). In developing this system, we recognize that product returns are inherently part of online retailers' business model and therefore concentrate on extreme cases with a very high likelihood of a product return. The large majority of customers would not be affected by this strategy. We will briefly discuss several possibilities for such interventions:

1. *Limiting customers' payment options* - Customers who try to make purchases that are very likely to be returned are requested to pay in advance, via credit card or to use an online payment service. This strategy is motivated by the fact that customers who pay after delivery are about twice as likely to return an item they purchased than those who pay in advance (Asdecker, 2015). The reason for this phenomenon is that when returning products, customers who pay after delivery do not have to make sure that their money is refunded. Instead, they simply never transfer the money in the first place. This suggests that requiring advance payment or direct debit could be an effective strategy against excessive product returns. Pay-after-delivery is the a common means of payment in many countries [3].

2. *Artificially increasing delivery time* - Customers are told that products for which the system predicts a high likelihood of being returned are currently out of stock in the hope that customers will then remove them from their shopping basket. Of course, such a policy can adversely affect customer loyalty, should be used with care and only for very extreme cases.

3. *Moral suasion* - When a customer tries to make purchase that is very likely to be returned, a pop-up window reminds him or her of the environmental impact associated with product returns. Environmental labeling has been shown to effectively influence consumer behavior (Aguilar and Cai, 2010; Bjørner et al., 2004a,b; D'Souza et al., 2006).

[3] https://www.about-payments.com/knowledge-base/methods, retrieved 2015-04-08

4. *Rejecting the transaction* - In very extreme cases, online retailers may choose to outright reject the order. Consumer protection laws protect customers' rights after a transaction takes place, but they cannot force companies to accept the transaction in the first place.

The purpose of this study is to develop the most important prerequisite for the described strategy, namely a method for accurately predicting product returns. To the best of our knowledge, this study is the first in the academic literature to use machine learning techniques for the identification of consumption patterns associated with a high product return rate. So far, machine learning has only been applied to product returns in very different settings such as forecasting returns of end-of-life goods (Clottey et al., 2012; Toktay et al., 2004). In addition, surveys among online retailers have shown that dynamic predictive methods are not yet widespread in practice and mainly based on very simple indicators such a customer's past return rate (Pur et al., 2013).

Our approach is based on a large dataset containing 1,149,262 purchases obtained from a major German online retailer specializing in fashion. We develop a decision support system including a new algorithm for linear dimensionality reduction, called *Mahalanobis feature extraction*. We demonstrate that Mahalanobis feature extraction is effective for predicting product returns when combined with an adaptive boosting classification algorithm and show that this combination outperforms a wide selection of benchmarks.

The remainder of this study is organized as follows: In the next section we will review related literature. We then introduce our dataset establishing the need for a dimensionality reduction algorithm. After that, we develop Mahalanobis feature extraction. In the subsequent section we introduce our research methodology and research hypothesis. We then present the results of our evaluation. In the final two section, we discuss our results and conclude.

3.2 Literature Review

The issue of product returns has been investigated in numerous previous studies. These studies have demonstrated that allowing customers to return products should be understood as being inherently part of a retailer's business model and tried to identify optimal return policies. A number of previous studies have also suggested that the way products are presented can have significant impact on the likelihood of a product return. Finally, the issue of fraudulent behavior has been thoroughly investigated.

3.2.1 Product Returns in E-Commerce

Many authors have demonstrated the importance of product returns to online retailers' business models. It has been shown that a more lenient product return policy has a positive impact on customers' willingness to purchase products (Autry, 2005; Bower and Maxham III, 2012; Petersen and Kumar, 2009; Stock et al., 2006). A more lenient policy towards product returns can also have a positive impact on customer satisfaction (Cassill, 1998; Dissanayake and Singh, 2008), customer loyalty (Mollenkopf et al., 2007) or perceived fairness (Pei et al., 2014). There are two theoretical explanations for this phenomenon. First, a lenient return policy reduces the risk customers take when purchasing products online (Che, 1996; Heiman et al., 2001; Li et al., 2013; Schmidt et al., 1999; Wood, 2001) and can be interpreted through the lens of finance theory as an option (Anderson et al., 2009; Heiman et al., 2002). Second, allowing customers to return products purchased online can be interpreted as a signal for higher product quality (Bonifield et al., 2010).

Theoretical research has demonstrated that there is an optimal rate of return, where the positive impact and the costs associated with product return are in balance (Yan, 2009) and several theoretical models have been proposed to identify the ideal level of leniency. Such an optimal rate of return exists even when taking competition into account (Li et al., 2012b). In competitive environments with limited self space, offering generous return policies even for perishable goods is the only Nash equilibrium of the game (Bandyopadhyay and Paul, 2010). When demand is known, return policies increase competition (Padmanabhan and Png, 1997). The optimal return policy based on a straight-forward two-period model for customer preferences for prices and return policies can be demonstrated to be relatively insensitive to parameter assumptions (Ketzenberg and Zuidwijk, 2009). Specific practical insight gained from such models includes the recommendation that marketing campaigns that decrease price sensitiveness of customers will lead to higher profits (Mukhopadhyay and Setoputro, 2004).

Other authors take a more practical approach to the issue of product returns, investigating the impact of specific product return strategies. We can differentiate between monetary strategies such as offering discounts for not returning product, procedure instruments such safety packaging and customer-centric strategies which focus on individual customers such as giving product advice (Walsh et al., 2014). A widely discussed example for a monetary strategy is to charge for the shipping costs associated with product returns (the common practice is to offer free returns). Such a strategy can be shown to lower prices (Ancarani et al., 2009; Chen and Bell, 2009). It also is recognized that information provision can have significant impact on the likelihood of a product return (Zhou et al., 2006). For instance, presenting items of clothing in an emotionally charged way (such as attractive models against a beautiful background) rather than a neutral way will increase the likelihood of a product return (De et al., 2012). On the other hand, positively framing the brands can decrease the likelihood of a product return (Bechwati and Siegal, 2005). Sellers can also use the price and restocking fee as a means of targeting different customer groups (Shulman et al., 2009).

There is empirical evidence that online shopping encourages impulsive (spontaneous and unplanned) or even compulsive (addictive) consumption patterns or can overwhelm customers which leads to an increase in product returns (LaRose, 2001; Rabinovich et al., 2011). Another problem is the issue of fraud: Customers sometimes abuse consumer protection rights (Heiman et al., 2001) or violate social norms (Autry et al., 2007). A typical example would be to purchase an item of clothing, wear it on one or two occasions and then return it to the retailer (Harris, 2010; King and Dennis, 2006; Wachter et al., 2012). This is the most frequent form of fraud (Speights and Hilinski, 2005). In more serious cases, the returned item is not the item that the customer originally ordered or the item was broken by the customer (Harris, 2010; King and Dennis, 2006; Wachter et al., 2012).

3.2.2 Predictive Analytics in Information Systems Research

Predictive analytics has been recognized as an important part of information systems research (Agarwal and Dhar, 2014; Shmueli and Koppius, 2011). Machine learning algorithms have been successfully applied to a wide variety of problem domains such as detecting financial fraud (Abbasi et al., 2012), identifying fake websites (Abbasi et al., 2010), detecting credit card fraud (Bhattacharyya et al., 2011), sales forecasting (Choi et al., 2011), recommender systems (Sahoo et al., 2012) or credit scoring (Zhang et al., 2010).

However, few authors have suggested the use of data analysis to address the issue of product

returns. Yu and Wang (2008) propose a hybrid mining approach to divide customers into different segments and offering different return policies to these segments. Some authors propose methods of forecasting returns of end-of-life goods (Clottey et al., 2012; Toktay et al., 2004), which is particularly relevant to electronic products or product that can be environmentally hazardous.

3.2.3 Implications

In summary, there are two important lessons to be drawn from the existing literature which are highly relevant for the development of a decision support system for product returns and its subsequent evaluation.

1. Product returns are inherently part of online retailers' business models. In order to stay competitive with more traditional forms of retail, online retailers, particularly in the fashion industry, must give customers the right to return products if they find they do not like them after purchase.

2. Some customers abuse these rights. Due to impulsive, compulsive or even fraudulent behavior, we often observe transactions that are not profitable.

We argue that there is an important gap in the academic literature in that there currently is no strategy that is both practical and compatible with online retailers' business model. In reality, it may be difficult to implement an optimal return policy based on very abstract mathematical models such as the ones proposed by Bandyopadhyay and Paul (2010); Ketzenberg and Zuidwijk (2009); Li et al. (2012b); Mukhopadhyay and Setoputro (2004); Padmanabhan and Png (1997); Shulman et al. (2009, 2011) or Yan (2009). We also believe that a strategy of charging for the shipping costs associated with returns, as frequently proposed in the literature (Ancarani et al., 2009; Chen and Bell, 2009), will punish customers who have legitimate reasons for returning products and is not compatible with online retailers' business model. Information provision may have an impact on product returns, but the statistical significance is only due to the size of the dataset and its impact from a business perspective is actually quite low (De et al., 2012).

A successful and practical strategy should focus on identifying consumption patterns associated with a very high return rate *before the transaction even takes place*. To date there is no such strategy in the academic literature. In fact, there is very little research of customer-based return management strategies (Walsh et al., 2014). Also, there are very few studies that propose the use of machine learning techniques to address the issue of product returns. We believe such a strategy to be particularly promising as machine learning has been successfully applied to many different business intelligence problems.

3.3 Data Collection

We cooperated with a major German online retailer that specializes in fashion. For the purposes of this study, we analyzed a dataset consisting of a total of 1,149,262 product purchase from July 2014 to November 2014. The return rate over that period of time was 57.3%.

We separate the indicators contained in our dataset into two categories: Numeric indicators (such as a customer's past return rate or the total number of products in a basket), and nominal indicators (such as the product's brand or the sales channel). Nominal indicators differ

from numeric indicators in that they need to be encoded using dummy variables. From a technical point of view, it is beneficial both in terms of computing memory and CPU time to represent our dataset in CSR matrix format, which only stores the non-zero entries while being mathematically equivalent to standard encoding techniques traditionally used in machine learning.

We also differentiate between three levels, namely indicators on the product level (such as a product's brand or color), which may be different for every product in a basket, the basket level (such as the time of the purchase or the sales channel), which are similar for all products in a basket and the customer level (such as a customer's past return rate), which is attributed to a particular customer.

For reasons of data privacy, we abstain from analyzing any personal information, relying on completely pseudonumized data instead. When calculating a customer's past return rate, we included an additional dataset of purchases and returns between July 2013 and June 2014 (the twelve months prior to our actual dataset). When creating these features, we were guided by the assumption that a customer's past return rate will be a good predictor for his or her future return rate and that compulsive shoppers tend to accumulate a large number of similar products.

An overview of all indicators used is provided in Table 8.

In total, our dataset consists of 5868 dummy variables, 10 numeric features and 1,149,262 samples. The overall density of the dataset is roughly 0.33%, demonstrating that the use of sparse matrix format reduces the memory requirement by over 99%.

3.4 Mahalanobis Feature Extraction

Most machine learning algorithms do not scale to datasets like the one used in this study or are not applicable to sparse matrices. In this section, we develop a new approach for dimensionality reduction that is particularly useful for large-scale sparse matrices.

3.4.1 Motivation

Suppose we have an $(I \times J)$-matrix $\mathbf{X}$, I being the number of samples, J being the number of features and $I > J$. Suppose further that we have a $(I \times J^{ext})$-matrix $\mathbf{Y}$ our goal being to train algorithms to predict $\mathbf{Y}$ using the data contained in $\mathbf{X}$. $\mathbf{Y}$ can be continuous or discrete, therefore the framework is equally applicable to classification and regression problems. Instead of training our algorithms directly on $\mathbf{X}$, we first transform $\mathbf{X}$ and reduce it to a $(I \times J^{ext})$-matrix $\mathbf{X^{ext}}$ on the basis of a weight matrix $\mathbf{W}$.

We describe these transformations at further detail: Let x_{ij}, y_{ij} and x_{ij}^{ext} be the element in the i^{th} row and j^{th} column of $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{X^{ext}}$ respectively. Let $\mathbf{x_i}$ be the i^{th} row in $\mathbf{X}$ and $\mathbf{w_{:j}}$ be the j^{th} column in $\mathbf{W}$. Let $f_j(...)$ be the j^{th} transformation function. Then, $x_{ij}^{ext} = f_j(\mathbf{x_i}, \mathbf{w_{:j}})$.

Our goal is to optimize $\mathbf{W}$, such that we maximize the predictive power $\mathbf{X^{ext}}$ has for $\mathbf{Y}$.

Indicator	**Category**	**Number of features**
Product level		
Product *brand*	nominal	672
Product *color*	nominal	2055
Product *size*	nominal	1053
Product *target group*	nominal	6
Activity product is designed for	nominal	51
Product *category*	nominal	65
Product *subcategory*	nominal	783
Sales channel from which customer reached product	nominal	882
Number of times the *exact same product* is in basket	numeric	1
Number of products in basket with *same category* and *target group* in basket	numeric	1
Number of products in basket with same *subcategory* and target group in basket	numeric	1
Number of products in basket with *same brand* and *target group* in basket	numeric	1
Number of products in basket that share *identical characteristics* with the *exception of product size*	numeric	1
Number of products in basket that share *identical characteristics* with the *exception of product color*	numeric	1
Number of times a *similar product*, differing only in size and/or color, is in basket (on the basis of product ID)	numeric	1
Basket level		
Hour of the day at which basket was ordered	nominal	24
Platform from which basket was ordered	nominal	6
Device from which basket was ordered	nominal	32
Operating system from which basket was ordered	nominal	50
Web browser from which basket was ordered	nominal	185
Payment method	nominal	4
Total number of products in basket	numeric	1
Customer level		
Customer's past return rate	numeric	1
Number of products customer previously purchased	numeric	1
Total number of dummy variables		**5868**
Total number of numeric features		**10**
Total number of features		**5878**

TABLE 8: Overview of the indicators used

The predictive power $\mathbf{X}^{\mathbf{ext}}$ has for $\mathbf{Y}$ is measured using a statistical approach originally proposed by Urbanke et al. (2014). This approach was originally developed for the evaluation of predictive methods. It measures the probability of a set of correlation coefficients under the null hypothesis that the correlation coefficients are a product of a random reshuffling of the data. The test statistic is the squared Mahalanobis distance of the observation from the

expected value under the null hypothesis. Under certain assumptions it can be interpreted as being chi-squared distributed.

The approach proposed in this study is to maximize the Mahalanobis distance of the observation from the expected value under the null hypothesis, thus minimizing the probability that the null hypothesis is true. In that, we diverge from the usual machine learning practice of using Bayesian statistics and maximum likelihood, relying on frequentist statistics instead.

3.4.2 Basic Idea

Let z_j be defined as follows:

$$z_j = \sum_{i=1}^{I} x_{ij}^{ext} y_{ij}. \tag{31}$$

Let $\mathbf{Z}$ be a vector of length J^{ext} containing all z_j. We then establish the null hypothesis that $\mathbf{X^{ext}}$ has no predictive power for $\mathbf{Y}$ and is the product of a random reshuffling of $\mathbf{Y}$. Let E($\mathbf{Z}$) be the expected value of $\mathbf{Z}$ and $\mathbf{V}$ be the variance-covariance-matrix under the null hypothesis. E($\mathbf{Z}$) can then be calculated as follows (Urbanke et al., 2014):

$$E(z_j) = \frac{\sum_{i=1}^{I} x_{ij}^{ext} \sum_{i=1}^{I} y_{ij}}{I}. \tag{32}$$

$\mathbf{V}$ can be calculated as follows (Urbanke et al., 2014):

$$cov(z_j, z_{j'}) = \frac{\left(I\sum_{i=1}^{I} x_{ij}^{ext} x_{ij'}^{ext} - \sum_{i=1}^{I} x_{ij}^{ext} \sum_{i=1}^{I} x_{ij'}^{ext}\right)\left(I\sum_{i=1}^{I} y_{ij} y_{ij'} - \sum_{i=1}^{I} y_{ij} \sum_{i=1}^{I} y_{ij'}\right)}{I^2(I-1)}. \tag{33}$$

Our goal is to optimize $\mathbf{W}$ in order to maximize $(\mathbf{Z} - E(\mathbf{Z}))'\mathbf{V^{-1}}(\mathbf{Z} - E(\mathbf{Z}))$. Note that the dimensionality of $\mathbf{V}$ depends on the number of *extracted* features, rather than the number of *original* features. Since the number of extracted features is, by the definition of dimensionality reduction, a lot smaller than the number of original features, the approach can be applied to very large-scale problems.

Because maximizing $(\mathbf{Z} - E(\mathbf{Z}))'\mathbf{V^{-1}}(\mathbf{Z} - E(\mathbf{Z}))$ is equivalent to maximizing the Mahalanobis distance from $\mathbf{Z}$ to E($\mathbf{Z}$) under the null hypothesis, we call our new algorithm *Mahalanobis feature extraction*.

3.4.3 Numerical Approximation

The technique can be optimized using any numerical, gradient-based approach with mini-batch updating. In every iteration, $\mathbf{W}$ is updated using the following gradient. This necessitates the calculation of $\frac{\partial(\mathbf{Z}-E(\mathbf{Z}))'\mathbf{V^{-1}}(\mathbf{Z}-E(\mathbf{Z}))}{\partial w_{cd}}$.

Recall that $\mathbf{Z}$ is defined in (37). The partial derivative of any element in $\mathbf{Z}$ with respect to a single element w_{cd} is calculated as follows:

$$\frac{\partial z_j}{\partial w_{cd}} = 0, \text{ if } j \neq d. \tag{34}$$

$$\frac{\partial z_j}{\partial w_{cd}} = \sum_{i=1}^{I} \frac{\partial f_j(\mathbf{x_i}, \mathbf{w_j})}{\partial w_{cd}} y_{ij}, \text{ if } a = d.$$

Recall that $E(\mathbf{Z})$ is defined in (38). The partial derivative of any element in $E(\mathbf{Z})$ with respect to a single element w_{cd} is calculated as follows:

$$\frac{\partial E(z_j)}{\partial w_{cd}} = 0, \text{ if } j \neq d. \tag{35}$$

$$\frac{\partial E(z_j)}{\partial w_{cd}} = \frac{\sum_{i=1}^{I} \frac{\partial f_j(\mathbf{x_i},\mathbf{w_{:j}})}{\partial w_{cd}} \sum_{i=1}^{I} y_{ij}}{I}, \text{ if } a = d.$$

Recall that $cov(z_j, z_{j'})$ is defined in (39). The partial derivative of any element of $\mathbf{V}$ with respect to a single element w_{cd} is calculated as follows:

$$\frac{\partial cov(z_j,z_{j'})}{\partial w_{cd}} = 0, \text{ if } j, j' \neq d. \tag{36}$$

$$\frac{\partial cov(z_j,z_{j'})}{\partial w_{cd}} = \frac{\left(I\sum_{i=1}^{I} \frac{\partial f_j(\mathbf{x_i},\mathbf{w_{:j}})}{\partial w_{cd}} x_{ij'}^{ext} - \sum_{i=1}^{I} \frac{\partial f_j(\mathbf{x_i},\mathbf{w_{:j}})}{\partial w_{cd}} \sum_{i=1}^{I} x_{ij'}^{ext}\right)\left(I\sum_{i=1}^{I} y_{ij}y_{ij'} - \sum_{i=1}^{I} y_{ij}\sum_{i=1}^{I} y_{ij'}\right)}{I^2(I-1)}, \text{ if } j = d, j \neq j'.$$

$$\frac{\partial cov(z_j,z_{j'})}{\partial w_{cd}} = 2\frac{\left(I\sum_{i=1}^{I} \frac{\partial f_j(\mathbf{x_i},\mathbf{w_{:j}})}{\partial w_{cd}} x_{ij}^{ext} - \sum_{i=1}^{I} \frac{\partial f_j(\mathbf{x_i},\mathbf{w_{:j}})}{\partial w_{cd}} \sum_{i=1}^{I} x_{ij}^{ext}\right)\left(I\sum_{i=1}^{I} y_{ij}y_{ij'} - \sum_{i=1}^{I} y_{ij}\sum_{i=1}^{I} y_{ij'}\right)}{I^2(I-1)}, \text{ if } j = j' = d.$$

3.4.4 Transformation functions

An overview of the transformation functions $f_j(...)$ used for the purposes of this study is provided in Table 9.

Transformation	**Definition**
Linear	$f_j(\mathbf{x_i}, \mathbf{w_{:j}}) = \sum_j^J w_{ij} * x_{ij}$
Logistic	$f_j(\mathbf{x_i}, \mathbf{w_{:j}}) = \frac{1}{1+\exp(-\sum_j^J w_{ij} * x_{ij})}$
Rectified linear	$f_j(\mathbf{x_i}, \mathbf{w_{:j}}) = \max(\sum_j^J w_{ij} * x_{ij}, 0)$
Softplus	$f_j(\mathbf{x_i}, \mathbf{w_{:j}}) = \ln(1 + \exp(\sum_j^J w_{ij} * x_{ij}))$

TABLE 9: Overview of transformation functions used

3.4.5 Implementation, Parallelization and Complexity

We separate the necessary tasks into two groups: The first group consists of those tasks that are independent from $\mathbf{W}$ and have to be calculated only once. The second group consists of those tasks that are dependent on $\mathbf{W}$ and have to be recalculated in every iteration. Since we use a minibatch updating approach, we separate both $\mathbf{X}$ and $\mathbf{Y}$ into equally sized batches. $(\mathbf{Z} - E(\mathbf{Z}))'\mathbf{V^{-1}}(\mathbf{Z} - E(\mathbf{Z}))$ and its derivatives are then not calculated on the entire dataset, but on the individual batches.

It turns out that the sufficient statistics can be comfortably calculated using eight reduce operations, all of which are iterated over individual samples. This is a particularly useful finding, because it implies that the algorithm can be easily parallelized in a distributed-memory system and is therefore applicable to very large problems. The reduce operations are displayed in Figure 3.

Note that we iterate over the columns when calculating sums over $\mathbf{X}$ and $\mathbf{X^{ext}}$. For reasons of cache-efficiency, this is generally not advisable, but even more problematic in our setting, since $\mathbf{X}$ is a sparse matrix. For that reason, we divide $\mathbf{X}$ along its vertical dimension into smaller matrices corresponding to the batches. We then transpose these smaller matrices. In addition, we calculate the transpose of $\mathbf{X^{ext}}$ rather than $\mathbf{X^{ext}}$ itself.

The algorithm is particularly scalable to large problems, because it takes $\mathcal{O}(I)$ and $\mathcal{O}(J)$

For all batches:
reduce1: Calculate $\sum_{i \in batch} y_{ij}$ for all j
reduce2: Calculate $\sum_{i \in batch} y_{ij} y_{ij'}$ for all j, j'
In every iteration, for all batches:
Calculate $x_{ij}^{ext} = f_j(\mathbf{x_i}, \mathbf{w_{:j}})$ for all $i \in batch, j$
reduce3: Calculate $\sum_{i \in batch} f_j(\mathbf{x_i}, \mathbf{w_{:j}})$ for all j
reduce4: Calculate $\sum_{i \in batch} f_j(\mathbf{x_i}, \mathbf{w_{:j}}) y_{ij}$ for all j
reduce5: Calculate $\sum_{i \in batch} f_j(\mathbf{x_i}, \mathbf{w_{:j}}) f_{j'}(\mathbf{x_i}, \mathbf{w_{:j'}})$ for all j, j'
Calculate $\frac{\partial x_{id}^{ext}}{\partial w_{cd}} = \frac{\partial f_d(\mathbf{x_i}, \mathbf{w_{:d}})}{\partial w_{cd}}$ for all i, c, d
reduce6: Calculate $\sum_{i \in batch} \frac{\partial f_d(\mathbf{x_i}, \mathbf{w_{:d}})}{\partial w_{cd}}$ for all c, d
reduce7: Calculate $\sum_{i \in batch} \frac{\partial f_d(\mathbf{x_i}, \mathbf{w_{:d}})}{\partial w_{cd}} y_{id}$ for all c, d
reduce8: Calculate $\sum_{i \in batch} \frac{\partial f_d(\mathbf{x_i}, \mathbf{w_{:d}})}{\partial w_{cd}} f_{d'}(\mathbf{x_i}, \mathbf{w_{:d'}})$ for all c, d, d'
Calculate $(\mathbf{Z} - E(\mathbf{Z}))' \mathbf{V^{-1}} (\mathbf{Z} - E(\mathbf{Z}))$ as defined in (37), (38) and (39)
Update $\mathbf{W}$ using appropriate numerical technique
Repeat until convergence or maximum number of iterations is reached

FIGURE 3: Expression of Mahalanobis feature extraction in pseudocode

time. This means that if we double the number of samples I or double the number of features J , we would expect training time to double as well. In practice, it may even take less than $\mathcal{O}(I)$ time, because we use a minibatch updating approach and would therefore expect the algorithm to take fewer iterations of the entire dataset until convergence. Because of the matrix inversion, the algorithm is expected to take $\mathcal{O}\left((J^{ext})^3\right)$ time, but since J^{ext} is small by the definition of dimensionality reduction, this should not pose much of a problem.

We implement the algorithm in C++ and write an interface to Python using SWIG. We parallelize the code using OpenMPI.

3.5 Methodology and Research Hypotheses

We compare our framework with a selection of state-of-the-art techniques commonly used for dimensionality reduction, both supervised and unsupervised. These techniques are chosen on the basis of their applicability to very large-scale sparse matrices and the availability of appropriate implementations. We also test a number of classification algorithms that have been demonstrated to be successful across a large variety of problem domains (Fernández-Delgado et al., 2014).

Our analysis is conducted using the programming language Python and the machine learning library scikit-learn, developed by Pedregosa et al. (2011), as it offers excellent support for sparse matrices.

To structure our analysis, we develop a set of research hypotheses. Our goal is to demonstrate that our proposed solution for predicting product returns, a combination of adaptive boosting and Mahalanobis feature extraction does not only yield sufficiently reliable prediction, but also outperforms standard algorithms in this particular problem setting. This leads us to our first research hypothesis:

> *Research Hypothesis 1a. A combination of adaptive boosting and Mahalanobis feature extraction will yield more accurate predictions for product returns than state-of-the-art predictive algorithms and techniques for feature extraction or feature selection.*

In addition, there are some algorithms that are suitable for large-scale problems without necessitating dimensionality reduction. In particular, we consider a logistic regression and a linear kernel support vector machine. Linear kernel support vector machines scale very well when trained in their primal form (as opposed to the more common dual form) using a stochastic gradient descent algorithm. Hence, our second research hypothesis:

> *Research Hypothesis 1b. A combination of adaptive boosting and Mahalanobis feature extraction will yield more accurate predictions for product returns than a logistic regression and a linear kernel support vector machine that was trained on the entire dataset.*

Finally, we demonstrate that our framework fulfills the purpose it was designed for and provides predictions that are reasonably accurate for companies to work with. In particular, we address the p-value problem as discussed by Lin et al. (2013), who argue that hypothesis tests become less meaningful for large sample sizes and that researchers should be more concerned with what is *interesting* rather than what is *statistically significant*.

We approach this problem in two ways: First, based on our considerations that product returns are part of the business model for online retailers, we reason that an effective strategy for prediction and prevention should focus on extreme cases. Business solutions for this problem should be measured on the basis of their effectiveness at identifying product purchase intentions with a very high likelihood of a product return. Second, we follow the advise by Cohen (1992) and consider our predictive method's effect size.

> *Research Hypothesis 2. A combination of adaptive boosting and Mahalanobis feature extraction will identify consumption patterns associated with a very high rate of product returns at a level of accuracy that is sufficient for a strategy of prediction and intervention to be feasible.*

To conduct an out-of-sample-analysis, we separate our dataset into five equally sized batches, using four of them as our training set and the last one as our testing set. Samples are randomly assigned to the batches. We then corroborate our results by repeating our experiment using a different batch as the testing set every time. We also separate the 10 numeric features, which do not require dimensionality reduction, from the 5868 dummy variables, which do require dimensionality reduction. Using different techniques for feature extraction and feature selection (described below), we reduce the 5868 dummy variables to 10 numeric features, which we then combine with the original 10 numeric features to form a dataset consisting of 20 numeric features. The reduction to 10 features in particular is a compromise between reducing the computing resources required and keeping information loss to a minimum. All feature extractors and algorithms are trained on the training set. The evaluation is conducted using the testing set. The testing set is not involved in the training either the algorithms or the feature extractors in any way.

3.6 Results

To evaluate the first hypothesis, we compare Mahalanobis feature extraction, using the transformation functions discussed above, to a total of six techniques for feature extraction or feature selection, namely principal component analysis, linear discriminant analysis, a randomized truncated singular value decomposition, a feature selector that selects features based on their univariate chi-squared statistic, a random projection and non-negative matrix factorization. As an seventh "feature extraction technique" we simply ignore the nominal indicators.

We use each of these techniques on the 5868 dummy variables and reduce them to 10 numeric features. We then combine these 10 extracted features with the 10 original numeric features to form a dataset containing 20 features. The only exception is linear discriminant analysis: Since the number of extracted features cannot be larger than one in our case, we only extract a single feature. Even though linear discriminant analysis is a very popular algorithm for supervised dimensionality reduction, it actually does not scale very well: In our case, we would have to calculate and invert two (5868×5868)-variance-covariance matrices as opposed to the (10×10)-variance-covariance matrices necessary for Mahalanobis feature extraction. To keep the problem manageable, we first extract 100 features using principal component analysis and then train the linear discriminant analysis on top of these features. Such an approach is very common and implemented as the standard procedure in some machine learning libraries such as dlib C++ (King, 2009).

We also compare adaptive boosting to a total of eight other classification algorithms, namely classification and regression tree, extremely randomized trees, which are highly randomized bootstrapped decision trees, gradient boosting, linear discriminant analysis, since it can be used both as classifier and a feature extractor, logistic regression, a multilayer perceptron, random forest and a linear kernel support vector machine, since non-linear kernels do not scale to this problem size.

We combine each of the above-mentioned feature extraction techniques to each of the algorithms. Taking into account that we also try to ignore all nominal features, this amounts to a total of 99 (= 9 classifiers * 11 feature extractors) different combinations.

Since the logistic regression and linear support vector machines scale to large problem sizes (if trained using appropriate algorithms), we also train these algorithms on the entire training set, meaning both all numeric features and dummy variables.

We then calculate the correlation coefficient (Pearson's r) between the out-of-sample probabilistic predictions and the actual class variables in the testing set. Pearson's r assumes a value of 1 (best value possible) for perfect correlation between the probabilistic predictions and the actual class variables and a value of 0 for no correlation at all. It is an important measure for two reasons: First, a high correlation between the probabilistic predictions and the actual class variables implies a good ability to identify extreme cases with a high probability of a product return, which is the ultimate goal of this study. Second, Pearson's r is explicitly proposed to measure the effect size of an indicator (Cohen, 1992). Results are reported in Table 16. The best classifier given a particular feature extractor is underlined. The best feature extractor given a particular classifier is in *italics*.

We find that a combination of adaptive boosting and Mahalanobis feature extraction achieves the highest predictive accuracy. We also find that Mahalanobis feature extraction offers the highest predictive accuracy given any classifier when compared to any other method of fea-

	AdaBoost	**CART**	**ERT**	**GB**	**LDA**	**LR**	**MLP**	**RF**	**SVM**
Logistic	*0.41*	0.374	0.399	0.394	*0.358*	*0.365*	*0.411*	0.403	*0.312*
Linear	0.408	*0.382*	*0.402*	*0.4*	0.356	0.363	0.408	*0.404*	0.301
RectLin	*0.41*	0.374	0.4	0.394	*0.358*	*0.365*	*0.411*	0.403	*0.312*
Softplus	*0.41*	0.368	0.397	0.391	0.355	0.361	0.407	0.403	0.299
PCA	0.389	0.354	0.373	0.365	0.298	0.308	0.382	0.379	0.262
RanTSVD	0.382	0.347	0.367	0.358	0.298	0.308	0.376	0.373	0.263
LDA	0.384	0.375	0.384	0.381	0.329	0.337	0.389	0.386	0.29
ChiSelect	0.379	0.366	0.372	0.365	0.31	0.319	0.375	0.373	0.272
RP	0.357	0.326	0.347	0.341	0.276	0.287	0.348	0.351	0.256
NMF	0.387	0.354	0.369	0.36	0.291	0.301	0.373	0.376	0.251
NoNom	0.349	0.342	0.345	0.34	0.271	0.282	0.346	0.348	0.25
OrigData	-	-	-	-	-	0.371	-	-	0.301

Note: Number of extracted features is 1 for **LDA**, 0 for **NoNominal**, 5868 for **OrigData** and 10 for all others.
Logistic transformation = **Logistic** , linear transformation = **Linear** , rectified linear transformation = **RectLin** , softplus transformation = **Softplus** , principal component analysis = **PCA** , linear discriminant analysis = **LDA**, randomized truncated singular value decomposition = **RanTSVD**, feature selector based on univariate chi-squared statistic = **ChiSelect**, random projection = **RanProj**, non-negative matrix factorization = **NMF**, no nominal indicators = **NoNom**, classifiers trained on original dataset (where possible) = **OrigData**
Adaptive boosting = **AdaBoost**, classification and regression trees = **CART**, extremely randomized trees = **ERT**, gradient boosting = **GB**, linear discriminant analysis = **LDA**, logistic regression = **LR**, multilayer perceptron = **MLP**, random forest = **RF**, linear kernel support vector machine = **SVM**
Best classifier given feature extractor is underlined. Best feature extractor given classifier is in *italics*.

TABLE 10: Pearson's r between probabilistic out-of-sample predictions and actual class labels

ture extraction. In fact, a comparison between the logistic regression and the linear support vector machine trained on the original dataset and the same classifiers trained on the features extracted using Mahalanobis feature extraction shows that there is not much difference in predictive accuracy. This implies that even though we have reduced 5868 dummy variables to 10 numeric features (a reduction in the number of features by over 99.8%), the information loss associated with that reduction is not large and considerably smaller than for any other feature extraction algorithm.

We test whether this outperformance is statistically significant. Results are shown in Table 11. We find that the results are statistically significant at the 1%-level for all predictive methods.

In addition to analyzing Pearson's r, we calculate precision and recall: In our literature review, we have argued that product returns are inherently part of online retailers' business model and that a system for prediction and prevention should focus on extreme cases, namely product purchase intentions with a very high probability of a product return. Systems for the prediction and prevention of product returns should be measured by their ability to identify such extreme cases. For our evaluation this implies that *precision* is considerably more important than *recall*. In fact, considering that product returns are part of online retailers'

	AdaBoost	**CART**	**ERT**	**GB**	**LDA**	**LR**	**MLP**	**RF**	**SVM**
Logistic	-	0.004	0.358	0.040	-	-	-	0.741	-
Linear	0.459	-	-	-	0.433	0.336	0.299	-	0.000
RectLin	-	0.003	0.404	0.040	-	-	-	0.889	-
Softplus	-	0.000	0.078	0.002	0.194	0.170	0.184	0.743	0.000
PCA	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RanTSVD	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
LDA	0.000	0.009	0.000	0.000	0.000	0.000	0.000	0.009	0.000
ChiSelect	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RP	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
NMF	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
NoNom	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Note: Number of extracted features is 1 for **LDA**, 0 for **NoNominal**, 5868 for **OrigData** and 10 for all others.
Logistic transformation = **Logistic** , linear transformation = **Linear** , rectified linear transformation = **RectLin** , softplus transformation = **Softplus** , principal component analysis = **PCA** , linear discriminant analysis = **LDA**, randomized truncated singular value decomposition = **RanTSVD**, feature selector based on univariate chi-squared statistic = **ChiSelect**, random projection = **RanProj**, non-negative matrix factorization = **NMF**, no nominal indicators = **NoNom**, classifiers trained on original dataset (where possible) = **OrigData**
Adaptive boosting = **AdaBoost**, classification and regression trees = **CART**, extremely randomized trees = **ERT**, gradient boosting = **GB**, linear discriminant analysis = **LDA**, logistic regression = **LR**, multilayer perceptron = **MLP**, random forest = **RF**, linear kernel support vector machine = **SVM**
Best classifier given feature extractor is underlined. Best feature extractor given classifier is in *italics*.

TABLE 11: Pearson's r between probabilistic out-of-sample predictions and actual class labels

business model, a very high recall is actually undesirable. As long as recall is sufficiently high for the system to have a measurable impact, we should focus on maximizing the precision with which the system identifies product returns.

We define the 10%-threshold. This threshold is chosen such that the system would be expected to intervene for the 10% of all purchase intentions that are expected to be most likely to result in a product return. We then calculate precision and recall for all approaches based on the threshold. We have excluded the support vector machine from this table, as a probabilistic interpretation of the results in not possible. Results are reported in Table 12.

We find that the combination of the logistic transformation function and adaptive boosting achieves the highest precision at both the 10%-threshold. Regardless of the classifier, Mahalanobis feature extraction consistently achieves the highest precision and recall, with the exception of CART, where LDA performs slightly better. However, this is only restricted to this specific threshold. As shown in Tables 16 and 11, Mahalanobis feature extraction still achieves a statistically significant outperformance over other feature extraction methods.

In addition, we plot the receiver operating characteristic (ROC) curve. Since the extreme cases we are interested in are located in the lower left-hand part of the ROC curve, we concentrate on that section (see Figure 5). For comparison, we show all feature extractors used in this study in conjunction with the classifier that performed best given the particular feature extractor (in other words, the underlined combinations in Table 16). Our results

demonstrate that the classifier based on Mahalanobis feature extraction offers a better trade-off between the true positive rate and the false positive rate at all relevant levels.

		AdaBoost	**CART**	**ERT**	**GB**	**LDA**	**LR**	**MLP**	**RF**
Logistic	**P**	*0.851*	0.822	*0.845*	0.838	*0.821*	*0.821*	0.845	0.845
	R	*0.149*	0.144	*0.148*	0.146	*0.143*	*0.143*	*0.148*	*0.148*
Linear	**P**	0.849	0.82	*0.845*	*0.842*	0.819	0.82	0.842	0.846
	R	0.148	0.143	0.147	*0.147*	*0.143*	*0.143*	0.147	*0.148*
RectLin	**P**	0.847	0.822	*0.845*	0.838	*0.821*	*0.821*	*0.846*	0.846
	R	0.148	0.144	*0.148*	0.146	*0.143*	*0.143*	*0.148*	*0.148*
Softplus	**P**	*0.851*	0.818	0.843	0.839	0.819	0.818	0.844	*0.847*
	R	*0.149*	0.143	0.147	0.146	*0.143*	*0.143*	0.147	*0.148*
PCA	**P**	0.839	0.806	0.831	0.823	0.789	0.791	0.823	0.832
	R	0.147	0.141	0.145	0.144	0.138	0.138	0.144	0.145
RanTSVD	**P**	0.838	0.803	0.826	0.822	0.79	0.79	0.822	0.831
	R	0.146	0.141	0.144	0.144	0.138	0.138	0.143	0.145
LDA	**P**	0.835	*0.824*	0.837	0.833	0.805	0.804	0.832	0.837
	R	0.146	*0.145*	0.146	0.145	0.141	0.14	0.145	0.146
ChiSelect	**P**	0.834	0.817	0.826	0.819	0.788	0.788	0.821	0.826
	R	0.146	0.143	0.144	0.143	0.138	0.138	0.143	0.144
RanProj	**P**	0.82	0.791	0.813	0.809	0.777	0.776	0.812	0.816
	R	0.143	0.138	0.142	0.141	0.136	0.135	0.142	0.142
NMF	**P**	0.839	0.809	0.83	0.823	0.784	0.787	0.823	0.834
	R	0.146	0.141	0.145	0.144	0.137	0.137	0.144	0.146
NoNom	**P**	0.819	0.808	0.813	0.809	0.773	0.775	0.81	0.814
	R	0.143	0.141	0.142	0.141	0.135	0.135	0.141	0.142

Logistic transformation = **Logistic** , linear transformation = **Linear** , rectified linear transformation = **RectLin** , softplus transformation = **Softplus** , principal component analysis = **PCA** , linear discriminant analysis = **LDA**, randomized truncated singular value decomposition = **RanTSVD**, feature selector based on univariate chi-squared statistic = **ChiSelect**, random projection = **RanProj**, non-negative matrix factorization = **NMF**, no nominal indicators = **NoNom**, classifiers trained on original dataset (where possible) = **OrigData**

Adaptive boosting = **AdaBoost**, classification and regression trees = **CART**, extremely randomized trees = **ERT**, gradient boosting = **GB**, linear discriminant analysis = **LDA**, logistic regression = **LR**, multilayer perceptron = **MLP**, random forest = **RF**, linear kernel support vector machine = **SVM**

Precision = **P**, Recall = **R**

Best classifier given feature extractor is underlined. Best feature extractor given classifier is in *italics*.

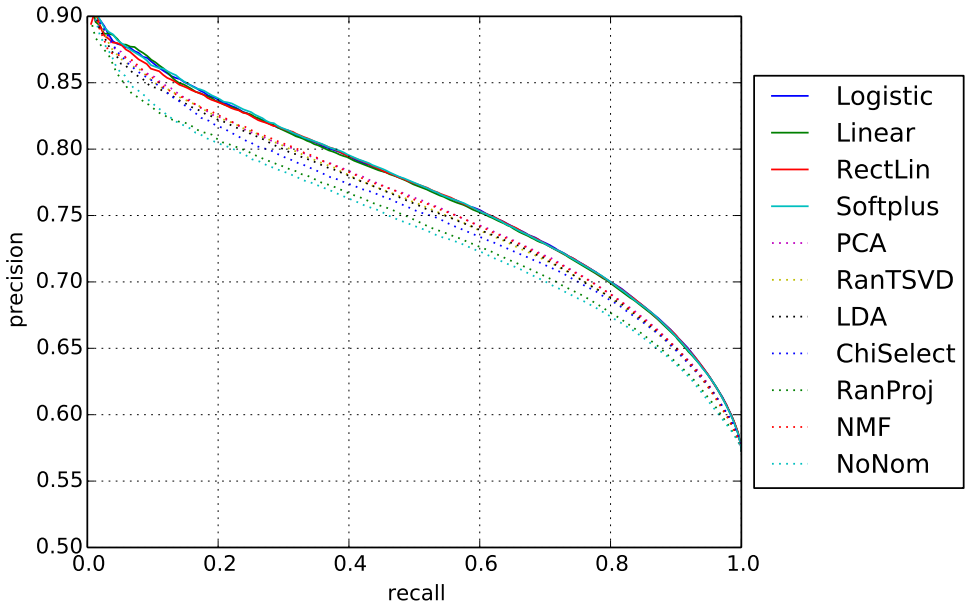
TABLE 12: Precision and recall for different methods and thresholds

In addition to a statistical analysis of the proposed method, we also analyze its usefulness as defined in Hypothesis 2. Lin et al. (2013) argue that hypothesis tests become less meaningful for large sample sizes and that researchers should be more concerned with what is *interesting* rather than what is *statistically significant*. Especially since the focus of this study is to provide a solution for a specific and important business problem (product returns in e-commerce), we provide evidence that the methodology proposed in this study can be used as the basis for a system for prediction and prevention of product returns.

We follow the advise by Cohen (1992) who propose to calculate Pearson's r to measure the effect size. A value of 0.3 implies a medium effect size whereas a value of 0.5 implies a

large effect size. Given that the predictive model proposed in this paper has a value of 0.41 (see Table 16), we conclude that our model achieves a medium to large effect size.

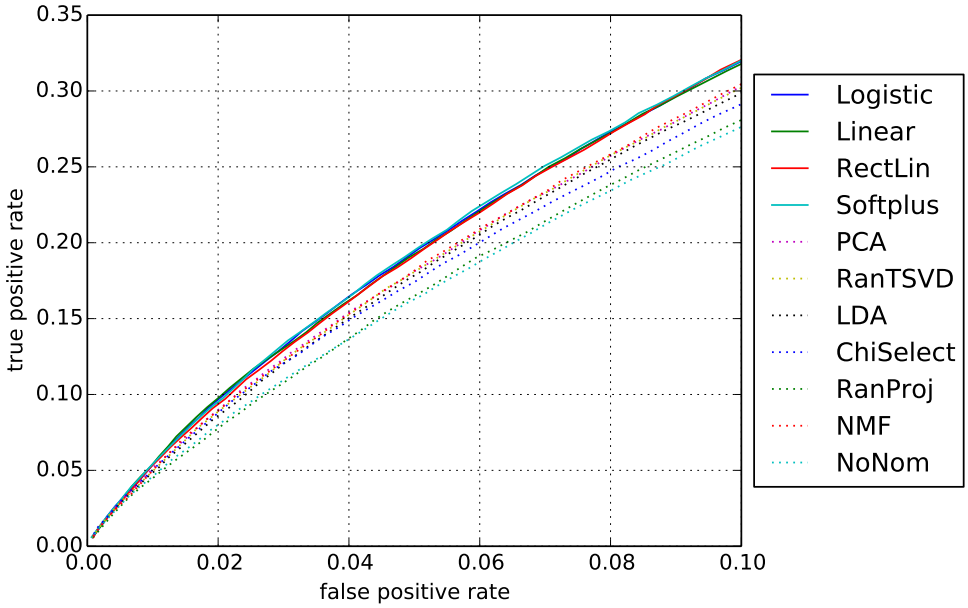
We also investigate the business value of the proposed system using precision and recall as calculated in in Table 12. These figures can be interpreted as follows: Suppose we were to construct a model of prediction and intervention based on a combination of adaptive boosting and the logistic transformation function. If we choose the 10%-threshold, the system would intervene for about 10% of all purchase intentions. 85.1% of all interventions would be justified. Assuming a reasonably effective intervention strategy, we would then be able to reduce the number of product returns by up to 14.9%. Considering that the company which provided the datasets sells several millions of products every year and every product return incurs costs of several euros, implementing such a system could easily reduce annual costs by a six-digit figure. We have no reason to believe that these findings could not be equally applied to other online retailers in fashion. Overall, this suggests that our findings have the potential for significant business impact.



Note: Benchmark feature extractors are shown as dotted line.
Logistic transformation = **Logistic** , linear transformation = **Linear** , rectified linear transformation = **RectLin** , softplus transformation = **Softplus** , principal component analysis = **PCA** , linear discriminant analysis = **LDA**, randomized truncated singular value decomposition = **RanTSVD**, feature selector based on univariate chi-squared statistic = **ChiSelect**, random projection = **RanProj**, non-negative matrix factorization = **NMF**, no nominal indicators = **NoNom**

FIGURE 4: Out-of-sample precision and recall for different feature extractors, in combination with adaptive boosting

We use the framework suggested by Urbanke et al. (2014) to test whether our predictive model generates additional information or could have been constructed by a combination of extant methods. In other words, we test whether its out-of-sample predictions remain statistically significant *when corrected for* the out-of-sample predictions generated by all standard feature extraction methods when combined with the same classifier. We find that they do so at very high significance levels. For comparison, we do the same for all other



Note: Benchmark feature extractors are shown as dotted line.
Logistic transformation = **Logistic** , linear transformation = **Linear** , rectified linear transformation = **RectLin** , softplus transformation = **Softplus** , principal component analysis = **PCA** , linear discriminant analysis = **LDA**, randomized truncated singular value decomposition = **RanTSVD**, feature selector based on univariate chi-squared statistic = **ChiSelect**, random projection = **RanProj**, non-negative matrix factorization = **NMF**, no nominal indicators = **NoNom**

FIGURE 5: Out-of-sample ROC-curves for different feature extractors, in combination with adaptive boosting

predictive methods we considered in this study. Results are reported in Table 13. We find that all the additional information generated by each of the feature extractors is statistically highly significant.

3.7 Discussion, Limitations and Implications for Further Research

The purpose of this study was to develop an innovative decision support system for the prediction of product returns in e-commerce. We have shown that our decision support system outperforms state-of-the-art methods in this particular problem domain and can generate predictions that are accurate enough for a system of prediction and targeted intervention to be feasible.

Using a large scale dataset related to product returns in e-commerce, we compared Mahalanobis feature extraction to a selection of the state-of-the art algorithms and feature selectors and demonstrated that it is able to outperform all of these methods given any of the classification algorithms we considered. This outperformance was statistically highly significant. We were also able to show that a combination of adaptive boosting and Mahalanobis feature extraction was able to generate new predictive information that other predictive methods could not achieve.

We also evaluated the business value of our decision support system. We demonstrated that the method is able to accurately identify cases in which the likelihood of a product being

	AdaBoost	**CART**	**ERT**	**GB**	**LDA**	**LR**	**MLP**	**RF**	**SVM**
Logistic	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Linear	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RectLin	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Softplus	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Logistic transformation = **Logistic** , linear transformation = **Linear** , rectified linear transformation = **RectLin** , softplus transformation = **Softplus**

Adaptive boosting = **AdaBoost**, classification and regression trees = **CART**, extremely randomized trees = **ERT**, gradient boosting = **GB**, linear discriminant analysis = **LDA**, logistic regression = **LR**, random forest = **RF**, linear kernel support vector machine = **SVM**

Best classifier given feature extractor is underlined. Best feature extractor given classifier is in *italics*.

TABLE 13: Statistical significance of predictive performance when corrected for all other feature extraction combined with the same classifier

returned is extremely high, suggesting that it enables a strategy of targeted intervention. In that we were able to fill an important gap in the existing academic literature on product returns: Whereas extant literature focused on optimizing the return policy for all customers, we proposed a strategy that focuses on individual customers and is therefore more compatible with online retailers' business model. Surveys among online retailers have demonstrated that such methods are not yet widespread: Less than 50% of online retailers analyze the likelihood of products being returned at all, 60% of those that do, do so only occasionally and they generally use only very simple measures such as a customers' past return rate to so (Pur et al., 2013).

We recognize that whether and to what extent a strategy of prediction and targeted intervention should be implemented is essentially a business decision. We have therefore provided a choice of different thresholds and calculated the accuracy with which our model can identify the likelihood of product returns given each of these thresholds. We have also discussed a range of possible intervention strategies ranging from soft interventions such as moral suasion, hard interventions such as not allowing a transaction to take place or a compromise such as limiting payment options, which makes product returns more tedious and forces customers to think twice without actually preventing the transaction.

Considering that many online retailers report that profitability would increase by more than 20%, if they could achieve a 10% reduction in the rate of product returns (Pur et al., 2013), even a conservative intervention strategy based on a predictive model such as the one proposed in this study could constitute a meaningful contribution to many companies' overall profit margins while at the same time reducing their carbon footprint. In addition, the large majority of responsible customers, who would not be targeted by the approach we propose, would benefit in that they would not have cross-subsidize irresponsible consumption patterns.

However, the success of "prediction and targeted intervention" as proposed in this study depends on the effectiveness of intervention mechanisms and customer acceptance. We believe that the use of field experiments to test different strategies in conjunction with a predictive model, such as the one proposed in this study, could provide further insight into the feasibility of different intervention strategies as well as customer acceptance. This would allow us to embed the predictive method presented in this paper into a larger customer lifetime value concept.

We have also introduced a new algorithm for linear dimensionality reduction, *Mahalanobis feature extraction,* and demonstrated that it outperforms state-of-the-art dimensionality reduction algorithms at statistically highly significant levels in the context of this particular problem domain. In principle, the algorithm is applicable to any problem domain which requires the reduction of large-scale sparse matrices. Because most customer databases contain categorical data, the algorithm is applicable to many business intelligence problems. We therefore believe that the usefulness of *Mahalanobis feature extraction* may transcend this particular problem domain. Future research will have to further examine the extent of its usefulness for other business intelligence problems.

3.8 Conclusion

In this paper, we developed a framework for predicting product returns in e-commerce by introducing *Mahalanobis feature extraction* as a new method of dimensionality reduction. We used an extensive dataset obtained from a major German online retailer specializing in fashion to evaluate our model. We demonstrated that *Mahalanobis feature extraction* in combination with an adaptive boosting algorithm outperforms a wide selection of benchmarks and shown that our model can effectively identify consumption patterns associated with a high rate of product returns.

4 A Customized and Interpretable Neural Network for High-Dimensional Business Data - Evidence from an E-Commerce Application

Abstract

Extracting actionable information from complex data is a key challenge for business analytics researchers (Hedgebeth, 2007). This is particularly difficult for high-dimensional datasets, to which an increasing number of businesses have access (Martens et al., forthcoming). In this study, we develop a customized neural network for extracting interpretable features from very high-dimensional datasets. These features can be interpreted both at an aggregated as well as a very fine-grained level. Interpreting non-linear interactions is no more difficult than interpreting a linear regression. We apply the algorithm to a dataset related to product returns in online retail which contains a total of 3,637,654 transactions and 26,875 dimensions. Comparing 74 different models, we demonstrate that, in addition to being interpretable, our algorithm yields higher predictive accuracy than extant methods. The approach is sufficiently holistic to applicable to a wide variety of business analytics datasets.

Keywords: Machine learning, Business intelligence, E-commerce, Online retail, Neural networks, Decision support

4.1 Introduction

Even though there is a wide variety of definitions for business analytics, it is commonly agreed upon that one of its most important goals is to provide *actionable information* or *knowledge* to business pracitioners (Hedgebeth, 2007). This is particularly true in the context of the "data explosion", which constitutes the biggest disruption in business practise and business research since the emergence of the internet (Agarwal and Dhar, 2014). An increasing number of businesses have access to massively fine-grained and thus very high-dimensional data (Martens et al., forthcoming). Extracting actionable information from these very high-dimensional datasets is an important and difficult challenge for business analytics researchers.

In this study, we develop a customized neural network to extract interpretable features from very high-dimensional business analytics datasets. We demonstrate the usefulness of the algorithm by using a real-world dataset related to product returns in e-commerce, which contains a total of 3,637,654 transactions and 26,875 dimensions. The output of our neural network can be interpreted at an aggregated as well as a very fine-grained level. Interpreting non-linear interactions is no more difficult than interpreting a linear regression. Comparing a total of 74 different models, we show that in addition to being interpretable, our approach generates more accurate predictions than state-of-the-art non-interpretable methods for dimensionality reduction and classification.

Even though this study uses a specific business case as an example dataset, the approach is applicable to any dataset that consists, at least in part, of nominal variables for which it is likely that there are significant non-linear interactions. Since many business analytics datasets contain a combination of nominal and numeric indicators, the proposed approach is potentially useful for a wide variety of business cases.

This study was inspired by the successful application of neural networks to artificial intelligence problems (Bengio et al., 2007) such as speech recognition (Sainath et al., 2015) or mastering the game of Go, which is exceptionally challenging (Silver et al., 2016). We reason that if such approaches can be designed to be interpretable, particularly for very high-dimensional data, they have the potential to be a powerful tool to gaining a deep understanding of causal relationships underlying business phenomena.

The idea that customization of neural networks should improve predictive accuracy is controversial - the most common form of neural networks, multilayer perceptrons, have been theoretically demonstrated to be universal approximators (Hornik et al., 1989). On the other hand, customized neural networks have superior predictive accuracy on tasks such as image classification (Ciresan et al., 2011, 2012) or speech recognition (Hochreiter and Schmidhuber, 1997). This suggests that, despite theoretical arguments to the contrary, including prior knowledge on the business problem in the architecture of a neural network can improve predictive accuracy.

The remainder of this study is organized as follows: In the following section, we review the related literature. We then develop the statistical basis for the dimensionality reduction approach introduced in this study. In the subsequent section, we present the data model. The following section contains our methodology. We then present our results. The final section concludes.

4.2 Literature Review

Research on neural networks dates back to Rosenblatt (1958), but did not attain wide-spread popularity until backpropagation was introduced to neural networks in the 1980s (Hecht-Nielsen, 1989; Rumelhart et al., 2004). Another important milestone was reached was when Hinton et al. (2006) demonstrated that the restricted Boltzmann machine can be used to pre-train very deep neural networks. Graphical processing units and the development of new optimization techniques have even enabled researchers to train deep and recursive neural networks without pre-training (Martens, 2010; Martens and Sutskever, 2011). Since then, deep neural networks have enjoyed considerable attention among artificial intelligence researchers with applications such as image recognition (Krizhevsky et al., 2012) or speech recognition (Dahl et al., 2012).

Even though numerous kinds of neural networks exist (Bishop, 1995), the long short term memory neural network (Hochreiter and Schmidhuber, 1997) is particularly noteworthy for the purposes of this study: The long short term memory neural networks has been developed to solve some of the problems associated with training deep recurrent neural networks to time series datasets. It contains a so-called “input gate”, an “output gate” and a “forget gate” which decide over the use of information in the network. It is among the most successful architectures to date (Schmidhuber, 2015) and has been successfully applied to sequential datasets (Schmidhuber et al., 2005) such as speech recognition (Sainath et al., 2015; Graves et al., 2013), keyword spotting (Fernández et al., 2007) or keyboard gesture decoding (Alsharif et al., 2015). The neural network architecture introduced in this study, particularly the idea of introducing AND-gates, was inspired by the gates in long short term memory neural networks.

A widely held belief on neural networks is that they are black box algorithms that are very useful for pattern recognition, but less useful for inferring causal relationships (Dreiseitl and

Ohno-Machado, 2002; Elmolla et al., 2010; Minsky, 1991; Sjöberg et al., 1995; Suykens and Vandewalle, 2012). However, a number of approaches exist to "illuminate the black box": Radial-basis function neural networks and multilayer perceptrons can be theoretically demonstrated to be functionally equivalent to fuzzy rules (Jang and Sun, 1993; Benítez et al., 1997). Such approaches have been demonstrated to be useful for extracting simple rules from simple, low-dimensional datasets (Benítez et al., 1997; Nauck and Kruse, 1997; Huang and Xing, 2002). A similar approach for extracting rules from neural networks for function approximation has been proposed by Setiono et al. (2000). Another possible method is to identify slow and fixed points which has also been shown to yield feasible results for low-dimensional datasets (Sussillo and Barak, 2013). Garson (1991) propose an approach to weigh the relative importance of input factors. Another possibility is to visualize the architecture of the neural network using a neural interpretation diagram that represents each of the weights by the thickness of the connecting line (Özesmi and Özesmi, 1999) or to use sampling techniques to study the contribution of each of the input factors (Olden and Jackson, 2002). Yet another possibility is to quantify pairwise non-linear interactions using the interactive effect and visualizing them using three-dimensional plots (Buckler and Hennig-Thurau, 2008).

Even though existing approaches for interpreting neural networks have been demonstrated to be workable on low-dimensional and toy datasets, they are more difficult to apply to high-dimensional, real-world business analytics datasets. In the case of our dataset, each rule extracted from one of the hidden nodes would have 26,857 elements, which makes it difficult to extract any meaningful information. Similar arguments apply for visualization techniques or sampling: It is difficult to see how a neural network with 26,857 input nodes could be visualized or how sampling techniques are feasible for architectures of this size. The problem becomes even more pronounced when we want to study the pairwise non-linear interactions using the interactive effects or pairwise visualization as proposed by Buckler and Hennig-Thurau (2008). This would result in 26,857*26,857 = 721,298,449 interaction plots. The approach could only be feasible with significant dimensionality reducion at the expense of predictive accuracy. Moreover, the plots they propose are in part based on arbitrarily set values, which can have significant impact on the shape of the generated plots, leading to inaccurate conclusions.

In this study, we propose a customized neural network for high-dimensional business analytics datasets. Interpreting a non-linear interaction is no more complicated than interpreting a linear model of comparable size, making the approach very scalable to high-dimensional datasets.

4.3 Calculation

In this section, we develop the algorithm by which we train the neural network used for dimensionality reduction (see below, particularly Figure 7).

4.3.1 Motivation

Suppose we have an $(I \times J)$-matrix $\mathbf{X}$, I being the number of samples, J being the number of dimensions. Suppose further that we have a $(I \times K)$-matrix $\mathbf{Y}$ with $K << J$, our goal being to train algorithms to predict $\mathbf{Y}$ using the data contained in $\mathbf{X}$. Suppose that $\mathbf{X}$ is very high-dimensional (J is very large). Our goal is then to transform $\mathbf{X}$ using a neural

network to produce the $(I \times K)$-matrix $\hat{\mathbf{Y}}$, in other words to optimize the parameters of the neural network to maximize the predictive power $\hat{\mathbf{Y}}$ has for $\mathbf{Y}$ as measured using a statistical approach originally proposed by Urbanke et al. (2014).

The basic intuition behind the approach is to maximize the impact (as measured in terms of R^2) that each of the extracted features has on the corresponding column in $\mathbf{Y}$ when corrected for the impact that other extracted features have on their corresponding columns in $\mathbf{Y}$.

4.3.2 Basic idea

Let x_{ij}, y_{ij} and $\hat{y}_{ij}$ be the element in the i^{th} row and j^{th} column of $\mathbf{X}$, $\mathbf{Y}$ and $\hat{\mathbf{Y}}$. respectively. Let z_j be defined as follows:

$$z_j = \sum_{i=1}^{I} \hat{y}_{ij} y_{ij}. \tag{37}$$

Let $\mathbf{Z}$ be a vector of length K containing all z_j. Under the null hypothesis that $\hat{\mathbf{Y}}$ has no predictive power for $\mathbf{Y}$, E($\mathbf{Z}$) is calculated as follows (Urbanke et al., 2014):

$$E(z_j) = \frac{\sum_{i=1}^{I} \hat{y}_{ij} \sum_{i=1}^{I} y_{ij}}{I}. \tag{38}$$

$\mathbf{V}$ is calculated as follows (Urbanke et al., 2014):

$$cov(z_j, z_{j'}) = \frac{\left(I\sum_{i=1}^{I} \hat{y}_{ij}\hat{y}_{ij'} - \sum_{i=1}^{I} \hat{y}_{ij} \sum_{i=1}^{I} \hat{y}_{ij'}\right)\left(I\sum_{i=1}^{I} y_{ij}y_{ij'} - \sum_{i=1}^{I} y_{ij} \sum_{i=1}^{I} y_{ij'}\right)}{I^2(I-1)}. \tag{39}$$

Our goal is to maximize $(\mathbf{Z} - E(\mathbf{Z}))'\mathbf{V}^{-1}(\mathbf{Z} - E(\mathbf{Z}))$, which can either be interpreted as the squared Mahalanobis distance of the sample outcome to the expected value under the null hypothesis or as a a chi-squared statistic.

4.3.3 Numerical approximation

We use standard backpropagation to optimize the neural network with $(\mathbf{Z} - E(\mathbf{Z}))'\mathbf{V}^{-1}(\mathbf{Z} - E(\mathbf{Z}))$ as our objective function. This necessitates the calculation of the partial derivative of $(\mathbf{Z} - E(\mathbf{Z}))'\mathbf{V}^{-1}(\mathbf{Z} - E(\mathbf{Z}))$ with respect to a single element $\hat{y}_{ij}$:

$$\begin{aligned}\frac{\partial(\mathbf{Z}-E(\mathbf{Z}))'\mathbf{V}^{-1}(\mathbf{Z}-E(\mathbf{Z}))}{\partial \hat{y}_{ij}} = 2\Big(\frac{\partial \mathbf{Z}}{\partial \hat{y}_{ij}} - \frac{\partial E(\mathbf{Z})}{\partial \hat{y}_{ij}}\Big)'\mathbf{V}^{-1}(\mathbf{Z} - E(\mathbf{Z})) - (\mathbf{Z} - \\ E(\mathbf{Z}))'\mathbf{V}^{-1}\frac{\partial \mathbf{V}}{\partial \hat{y}_{ij}}\mathbf{V}^{-1}(\mathbf{Z} - E(\mathbf{Z})).\end{aligned} \tag{40}$$

It follows from (40), that we have to develop formulas for calculating $\frac{\partial \mathbf{Z}}{\partial \hat{y}_{ij}}$, $\frac{\partial E(\mathbf{Z})}{\partial \hat{y}_{ij}}$ and $\frac{\partial \mathbf{V}}{\partial \hat{y}_{ij}}$.

An element $\frac{\partial z_a}{\partial \hat{y}_{ij}}$ in $\frac{\partial \mathbf{Z}}{\partial \hat{y}_{ij}}$ is calculated by deriving (37):

$$\begin{aligned}\frac{\partial z_a}{\partial \hat{y}_{ij}} &= 0, \text{ if } a \neq j \\ \frac{\partial z_a}{\partial \hat{y}_{ij}} &= y_{ij}, \text{ if } a = j.\end{aligned} \tag{41}$$

An element $\frac{\partial E(z_a)}{\partial \hat{y}_{ij}}$ in $\frac{\partial E(\mathbf{Z})}{\partial \hat{y}_{ij}}$ is calculated by deriving (38):

$$\begin{aligned}\frac{\partial E(z_a)}{\partial \hat{y}_{ij}} &= 0, \text{ if } a \neq j \\ \frac{\partial E(z_a)}{\partial \hat{y}_{ij}} &= \frac{\sum_{i=1}^{I} y_{ij}}{I}, \text{ if } a = j.\end{aligned} \tag{42}$$

An element $\frac{\partial cov(z_a, z_{a'})}{\partial \hat{y}_{ij}}$ in $\frac{\partial \mathbf{V}}{\partial \hat{y}_{ij}}$ is calculated by deriving (39):

$$\frac{\partial cov(z_a,z_{a'})}{\partial \hat{y}_{ij}} = 0, \text{ if } a,a' \neq d. \tag{43}$$

$$\frac{\partial cov(z_a,z_{a'})}{\partial \hat{y}_{ij}} = \frac{\left(I\hat{y}_{ia'} - \sum_{i=1}^{I}\hat{y}_{ia'}\right)\left(I\sum_{i=1}^{I} y_{ia}y_{ia'} - \sum_{i=1}^{I} y_{ia}\sum_{i=1}^{I} y_{ia'}\right)}{I^2(I-1)}, \text{ if } a = j, a \neq a'.$$

$$\frac{\partial cov(z_a,z_{a'})}{\partial \hat{y}_{ij}} = 2\frac{\left(I\hat{y}_{ia'} - \sum_{i=1}^{I}\hat{y}_{ia'}\right)\left(I\sum_{i=1}^{I} y_{ia}y_{ia'} - \sum_{i=1}^{I} y_{ia}\sum_{i=1}^{I} y_{ia'}\right)}{I^2(I-1)}, \text{ if } a = a' = j.$$

Taking this into account, we can rewrite the derivative. Let v_j be the j^{th} element in $\mathbf{V^{-1}}(\mathbf{Z} - E(\mathbf{Z}))$. Note that by (41), (42) and (43) the following holds true:

$$\begin{aligned}\frac{\partial(\mathbf{Z}-E(\mathbf{Z}))'\mathbf{V^{-1}}(\mathbf{Z}-E(\mathbf{Z}))}{\partial \hat{y}_{ij}} &= 2\left(\frac{\partial \mathbf{Z}}{\partial \hat{y}_{ij}} - \frac{\partial E(\mathbf{Z})}{\partial \hat{y}_{ij}}\right)'\mathbf{V^{-1}}(\mathbf{Z} - E(\mathbf{Z})) - (\mathbf{Z} - \\ &\quad E(\mathbf{Z}))'\mathbf{V^{-1}}\frac{\partial \mathbf{V}}{\partial \hat{y}_{ij}}\mathbf{V^{-1}}(\mathbf{Z} - E(\mathbf{Z})) \\ &= 2v_j\left(y_{ij} - \frac{\sum_{i=1}^{I} y_{ij}}{I}\right) - 2v_j\sum_{j'}\frac{\left(I\hat{y}_{ij'} - \sum_{i=1}^{I}\hat{y}_{ij'}\right)\left(I\sum_{i=1}^{I} y_{ij}y_{ij'} - \sum_{i=1}^{I} y_{ij}\sum_{i=1}^{I} y_{ij'}\right)}{I^2(I-1)}v_{j'}.\end{aligned} \tag{44}$$

For all batches:
- reduce1: Calculate $\sum_{i\in batch} y_{ij}$ for all j
- reduce2: Calculate $\sum_{i\in batch} y_{ij}y_{ij'}$ for all j, j'

In every iteration, for all batches:
- reduce3: Calculate $\sum_{i\in batch} \hat{y}_{ij}$ for all j
- reduce4: Calculate $\sum_{i\in batch} \hat{y}_{ij}y_{ij}$ for all j
- reduce5: Calculate $\sum_{i\in batch} \hat{y}_{ij}\hat{y}_{ij'}$ for all j, j'
- For all i, j:
  - Calculate $\frac{\partial(\mathbf{Z}-E(\mathbf{Z}))'\mathbf{V^{-1}}(\mathbf{Z}-E(\mathbf{Z}))}{\partial \hat{y}_{ij}}$ as described in (41), (42), (43) and (44)
- Update weights using standard backpropagation
- Repeat until convergence or maximum number of iterations is reached

FIGURE 6: Expression of the approach in pseudocode

4.3.4 Implementation and parallelization

We implement the algorithm in C++ and write an interface to Python using SWIG (Simplified Wrapper and Interface Generator) (Beazley, 1996). We parallelize the part of the code written in C++ using OpenMPI (Gabriel et al., 2004) and the part written in Python using mpi4py (Dalcin et al., 2008). This allows for efficient distributed-memory parallelization and makes our approach scalable to much larger datasets than the one presented in the current study.

The reduce operations necessary for minibatch update backpropagation using an MPI framework work are displayed in pseudocode in Figure 6.

4.4 Data Model

The dataset used has been obtained from a German online retailer selling fashion and sports equipment. It contains a total of 3,637,654 transactions, each of which represent one particular product that a customer has chosen to place in one particular virtual shopping basket. The goal is to predict whether the product will be returned using only information that is available at the time the product is ordered.

A major cause for product returns is impulsive shopping behavior (LaRose, 2001). Impulse

shopping is strongly related to a hedonistic consumption tendency (Hausman, 2000; Joo Park et al., 2006; Moe, 2003), in-store browsing (Beatty and Ferrell, 1998; Moe, 2003), gender, in that women tend to impulsively buy different items than men (Dittmar et al., 1995), or visual stimuli (Adelaar et al., 2003; Joo Park et al., 2006).

Description	
Actual price at purchase	
Standard product *price*	
Total number of products a customer has placed in the virtual shopping basket	
The number of products in the virtual shopping basket that have the *same category and target group* as this product	
The number of products in the virtual shopping basket that have the *same subcategory and target group* as this product	
The number of products in the virtual shopping basket that have a *product ID similar to this product*	
The number of products in the virtual shopping basket that have the same *brand and target group* as this product	
The number of products in the virtual shopping basket that have the *exact same features* as this product, with the *exception of color*	
The number of products in the virtual shopping basket that have the *exact same features* as this product, with the *exception of size*	
The number of products in the virtual shopping basket that have the *same category* as this product	
The number of products in the virtual shopping basket that have the *same target group* as this product	
The number of products in the virtual shopping basket that have the *same subcategory* as this product	
The number of products in the virtual shopping basket that were designed for the *same activity* as this product	
The number of products in the virtual shopping basket that have the *same category* and were designed for the *same activity* as this product	
The number of products in the virtual shopping basket that were purchased through the *same path*	
The *number of products* a customer has *previously purchased*	
The *share of products* a customer has *previously returned*. Assumes a value of 0, if customer has not purchased anything before.	
The *number of pictures* the product is advertised with	
Total number of dense variables	**18**

TABLE 14: Overview of dense variables

Since impulse shopping is tied to certain consumption patterns that we are able to observe in an e-commerce context (Moe, 2003), we can design our neural networks in such a way that they are particularly suitable for identifying consumption patterns associated with impulse shopping.

We differentiate between *indicators* and *variables*: Indicators are factors influencing the likelihood of a product being returned, such as the price of the product or its brand. Variables are quantitative measures representing these indicators. An indicator can be represented by one or several variables. For instance, the price of the product is represented by only one variable, whereas the brand of a product is represented by a wider selection of dummy variables (e.g., one dummy variable for Tommy Hilfiger, one for adidas etc.). If an indicator is represented by only one variable, we refer to this variable as a *dense variable* (e.g., price), otherwise we refer to the variables as *sparse variables* (e.g., brand).

The data model contains a total of 18 dense variables. An overview of the dense variables is provided in Table 14.

Based on the insight that product returns are often caused by impulsive or compulsive shopping patterns (LaRose, 2001), many of the variables constructed in Table 14 are related to interactions on the basket level: For instance, we hypothesize that if there are a large number of products of the same category in a virtual shopping basket, this will increase the likelihood of each of these products being returned, since it is an indicator for hedonistic consumption tendency and can be seen as the online equivalent of in-store browsing, both of which lead to impulse shopping (Beatty and Ferrell, 1998; Hausman, 2000; Joo Park et al., 2006; Moe, 2003).

However, these indicators neglect products of a *similar* (rather than the same) category in the same shopping basket. To capture these effects, we create an additional set of sparse variables that count the number of times a product of a particular category has been placed in the virtual shopping basket. We also measure how often the customer has previously purchased and previously returned products of a particular category, hypothesizing that if a customer has had purchased and returned a high number of products of a similar category, this will also impact the likelihood of a product being returned. We do the same for other indicators related to product characteristics as well.

This means that for each of our indicators related to product characteristics, we have four sets of variables: First, a set of dummy variables signifying the category/brand/color/... of the product itself, second, a set of variables counting the number of times a product of a particular category/brand/color/... has been placed into the virtual shopping basket, third, the number of times the customer has previously purchased a product of a particular category/brand/color/... and fourth, the number of times the customer has previously returned a product of a particular category/brand/color/....

An overview of the resulting sparse variables is provided in Table 15.

4.5 Methodology

4.5.1 Neural network for dimensionality reduction

Above, we have highlighted the importance of interpretability in a business analytics context. Given the high number of dimensions of the dataset described in the previous section, our goal is to develop a model that can effectively reduce the number of dimensions while maintaining interpretability. We achieve that goal by customizing a model that reflects the domain knowledge that there are interactions between a product and similar products the

Indicator	**Number of variables**			
	Product itself	**Interactions**		
		In Basket	**Previously purchased**	**Previously returned**
Hour of the day at which product was ordered	24	0	0	0
Platform from which product was ordered	7	0	0	0
Device from which product was ordered	37	0	0	0
Operating system from which product was ordered	49	0	0	0
Web browser from which product was ordered	87	0	0	0
Payment method used	5	0	0	0
Product *brand*	967	967	967	967
Product *category*	74	74	74	74
Product *subcategory*	957	957	957	957
Target group product is designed for	6	6	6	6
Activity product is designed for	69	69	69	69
Product *size*	395	395	395	395
Sales channel from which product was reached	771	771	771	771
Product *color*	3,423	3,423	3,423	3,423
Total number of sparse variables	**26,857**			

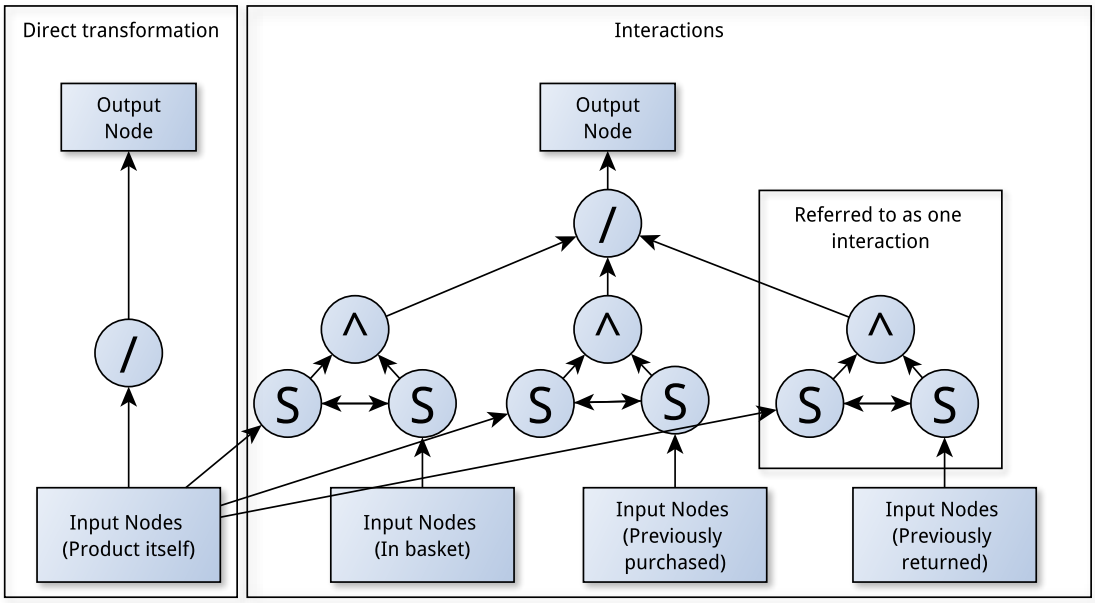
TABLE 15: Overview of sparse variables

customer has placed in the same basket, previously purchased or previously returned.

These interactions are modeled by combining the logistic function traditionally used in neural networks with the concept of an AND-gate. The logistic function always assumes a value between 0 and 1. We interpret this in a fuzzy logic sense that the node is activated with the probability of the value the logistic function assumes. If two nodes are combined in an AND-gate, the probability of the AND-gate being activated can be calculated by multiplying the probability of the input nodes being activated.

For each of the eight indicators related to product characteristics (see Table 15), we model an interaction by transforming both the variables representing the product itself as well as the interaction variables (in basket, previously purchased, previously returned) using a logistic function. These two logistic functions are then inserted into an AND-gate. Both of the logistic functions inserted into the AND-gate share their weights with each other, which means that both of the logistic functions have the exact same weights at all times. Any combination of two logistic functions and one AND-gate is referred to as an interaction. For each of the eight indicators, we model thirty such interactions which are then combined using a linear function. In addition, we also model the direct effect (product itself) using a linear function. This basic architecture is visualized in Figure 7.

The data model also includes six indicators that are related to the virtual basket rather than



Legend: / = Linear function, S = Logistic function, ^ = AND-gate, <-> = weight sharing
Note: This figure only depicts three interactions for one feature. In reality, the model includes thirty interactions per feature.

FIGURE 7: Basic architecture of the neural network for dimensionality reduction

the product. For these indicators, modeling such an interaction does not make sense, because the variables representing them are dummy variables for which non-linear interactions are impossible (see Table 15). This means that the resulting neural network has 22 output nodes (14 direct transformations + 8 output nodes needed for the interactions), 720 hidden nodes (3 nodes needed for each interaction * 30 interactions per indicator * 8 indicators for which we model interactions) and 26,857 input nodes (see Table 15).

This neural network is trained using the algorithm described in Figure 6. Note that even though the example business case presented in this study is a single-label classification problem, the algorithm and the neural network structure have been deliberately designed to be applicable to other problem classes as well, including regression, multiclass and multilabel classification.

This customized architecture can be interpreted in two ways, depending on the desired level of detail: First, it is interpretable at an aggregated level by studying the impact of the extracted features. Second, due to the shared-weight architecture, the extracted features can be interpreted as a set of fuzzy groups. The degree to which each category belongs to the group is determined by its weight. Specific examples for both approaches are provided below (see Tables 19 and 20).

Because of the linear nature by which the interactions are combined (see Figure 7), any non-linear effects are contained in a single interaction. Because of the weight-sharing, we only have to read one list of features and their corresponding weights when interpreting an interaction. This means that interpreting an interaction requires no more effort than reading and interpreting the parameters of a linear model, which most researchers are very accustomed to.

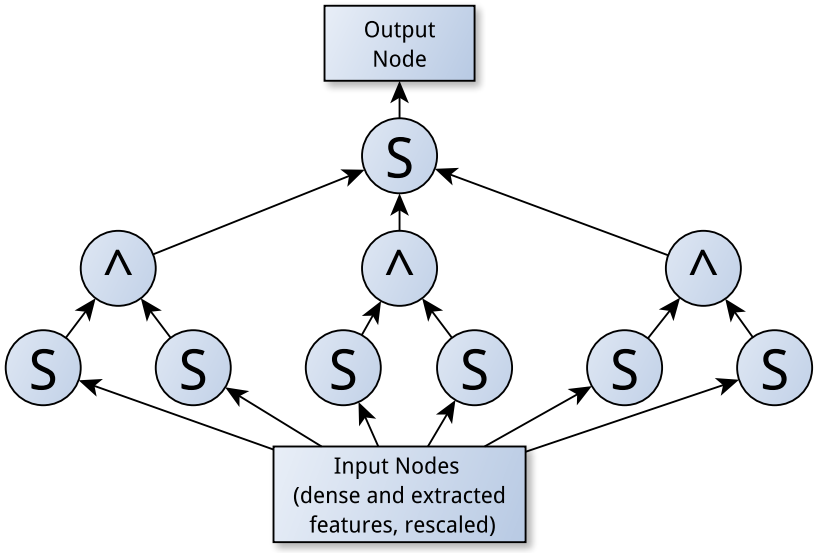
This approach significantly reduces the effort needed to get useful information over extant

methods: For instance, if we were to extract fuzzy rules from a multilayer perceptron as proposed by Benítez et al. (1997), each of the extracted rules contains only linearities and the non-linear effects we are interested in can only be interpreted by comparing different rules, which becomes extremely difficult as the number of dimensions increases.

4.5.2 Neural network for classification

Once we have extracted the features from the sparse variables, we combine these extracted features with the dense variables (see Table 14). We begin by rescaling each of these factors, so that every feature in the training set has a mean of zero and a standard deviation of 1. Failing to rescale will lead the neural network to a local optimum. The rescaled factors are then used to produce a probabilistic prediction whether a particular product is going to be returned. We train 50 logistic functions onto the input nodes, which are then combined in pairwise fashion into an AND-gate. These 25 AND-gates are then combined in another logistic function. The architecture of the neural network used for classification is depicted in Figure 8.

This particular architecture is mainly motivated by interpretability: Since the number of input nodes is only 40 (18 dense features + 22 extracted features), the problem of interpretability is not nearly as challenging as for the neural network used for dimensionality reduction (see Figure 7) and the standard rule extraction framework (Benítez et al., 1997) is applicable. However, this rule extraction technique extracts a set of linear rules from a multilayer perceptron and interpreting non-linearities captured by that multilayer perceptron might still be a challenge. The AND-gate structure as depicted in Figure 8 makes it easier to focus on non-linear interactions when interpreting the network.



Legend: S = Logistic function, ^ = AND-gate
Note: This figure only depicts only three interactions. In reality, the model includes 25 interactions.

FIGURE 8: Basic architecture of the neural network used for classification

4.5.3 Combined neural network

Finally, we can combine the two neural networks described above into one neural network which can also be trained as a whole using the parameters obtained from training the two previous neural networks for initialization. Since we have rescaled the dense and extracted features, these rescaling parameters must be included in the combined neural network.

The architecture of the combined neural network is depicted in Figure 9.

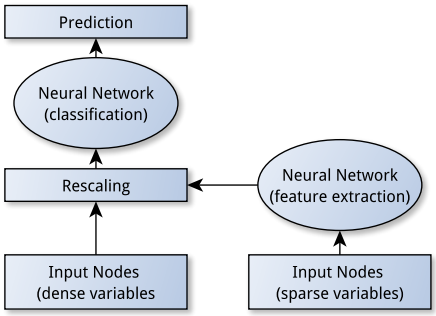


FIGURE 9: Architecture of the combined neural network

4.5.4 Evaluation

The goal of this study is to present a customized neural network that is interpretable. This implies that the evaluation of our approach should focus on two aspects: First, the effectiveness of the customized neural networks as a dimensionality reduction and classification technique in comparison to extant approaches (*predictive accuracy*). Second, our ability to extract knowledge from these approaches (*interpretability*). These two basic requirements are reflected in the evaluation of our model.

We begin by evaluating the predictive accuracy of the customized neural networks in comparison to extant methods. In order to do so, we separate our dataset into three parts: The training set (2,182,036 samples), the validation set (726,405 samples) and the testing set (729,213 samples). The dataset is separated along the basket IDs: All transactions belonging to one basket must also be part of the same set. We train the classifiers on the training set and use the validation set to optimize the hyperparameters for our classifiers. Once we have optimized the classifiers, we evaluate our dataset on the testing set. As soon as we have produced the results for the testing set, they are *immediately reported* in the results section of this study and *no further changes of any kind* to either the dimensionality reduction or the classification algorithms are allowed. This procedure ensures that our results are truly out-of-sample and not fine-tuned to the testing set in any way.

We then evaluate the interpretability of our methods by discussing some of the findings we were able to extract from the model.

4.6 Results

4.6.1 Predictive accuracy

The two neural networks described above can be either seen as two separate neural networks, one of which is used for dimensionality reduction (see Figure 7) the other one for classifi-

cation (see Figure 8), or as one combined neural network (see Figure 9). All of these three neural networks are evaluated separately.

We compare the neural network for dimensionality reduction to both a collection of dimensionality reduction techniques (principal component analysis, linear discriminant analysis, feature selector based on univariate chi-squared statistic, random projection). To ensure comparability with the neural network used for dimensionality reduction (see Figure 7), we also extract 22 features using the benchmark approaches, with the exception of linear discriminant analysis, for which this is not possible and only one feature is extracted. We also compare the proposed dimensionality reduction algorithm to an uncustomized version of the neural network used for dimensionality reduction based on 22 logistic functions trained on all 26,857 sparse variables using the mathematical approach described above. We then evaluate how predictive performance behaves without any of the sparse variables as defined above and listed in Table 15.

The neural network used for classification is also compared to a collection of benchmark algorithms, namely adaptive boosting, classification and regression tree, extremely randomized trees, linear discriminant analysis (which is both a classifier and a dimensionality reduction), logistic regression, a multilayer perceptron, random forest and a linear kernel support vector machine.

The neural networks, the multilayer perceptron and the Mahalanobis-distance based dimensionality reduction algorithms are self-implemented in C++ and parallelized using OpenMPI. All other algorithms are taken from the machine learning library scikit-learn (Pedregosa et al., 2011). We also consider benchmarking against structural equation models, but find that standard implementations such as SmartPLS are not scalable enough for a dataset of this size and complexity.

In addition, we train the combined neural network as described in Figure 9 on the full dataset using the weights obtained from training the individual two networks for initialization. For comparison, we replicate the structure in Figure 9 using the logistic functions as the dimensionality reducing and the multilayer perceptron as the classifying neural network. The result is a deep-layered multilayer perceptron (with one additional layer for rescaling), which we also train on the dataset using the weights from the aforementioned algorithms for initialization. For both algorithms, we optimize the hyperparameters involved in the training process by varying the learning rate as well as momentum and evaluating on the validation set (which is different from the testing set, on the basis of which our results are reported). Both neural networks receive the same amount of computing resources for training. Comparison to the multilayer perceptron is particularly important, because in theory the multilayer perceptron is a universal approximator (Hornik et al., 1989) and customization should not be necessary.

Also, some standard algorithms implemented in scikit-learn are scalable enough that dimensionality reduction is not necessary. As additional benchmarks, we train these algorithms, namely the logistic regression and the linear-kernel support vector machine, on the original dataset without dimensionality reduction.

This results in a total of 74 different models (10 classifiers * 7 dimensionality reduction algorithm including “NoSparse” + 4 algorithms trained on the original dataset without dimensionality reduction). After tuning the hyperparameters on the validation set, we evaluate the models on the testing set. As soon as the results for the testing set are obtained, they are immediately reported with no further changes to the models of any kind to ensure a true out-of-sample evaluation.

We calculate Pearson's r between the probabilistic predictions and the actual class labels. The choice for Pearson's r is motivated by the fact that we are primarily interested in the model's ability to identify products with a high probability of being returned rather than overall accuracy (Urbanke et al., 2015a) and Pearson's r is a good measure for a model's discriminatory ability. Also, Pearson's r is readily interpretable and is an important measure for a model's effect size (Cohen, 1992). We also calculate the statistical significance at which the neural network used for dimensionality reduction (see Figure 7) outperforms other dimensionality reduction techniques. Results are reported in Tables 16 and 17.

	NN	**Ada-Boost**	**CART**	**ERT**	**GB**	**LDA**	**LR**	**MLP**	**RF**	**SVM**
NNDim-Red	*0.422*	*0.421*	*0.386*	*0.420*	*0.409*	*0.403*	*0.404*	*0.420*	*0.418*	*0.340*
LogDim-Red	0.411	0.409	0.380	0.405	0.406	0.391	0.393	0.409	0.412	0.326
PCA	0.387	0.387	0.347	0.381	0.374	0.308	0.321	0.386	0.385	0.277
LDA	0.386	0.379	0.366	0.382	0.376	0.323	0.332	0.384	0.386	0.282
ChiSelect	0.384	0.385	0.364	0.380	0.373	0.304	0.317	0.380	0.383	0.275
RanProj	0.381	0.382	0.342	0.373	0.367	0.302	0.314	0.377	0.378	0.269
NoSparse	0.370	0.369	0.357	0.367	0.361	0.287	0.300	0.370	0.372	0.265
OrigData	0.422		-	-	-	-	0.382	0.414	-	0.324

Note: Number of extracted features is 1 for **LDA**, 0 for **NoNominal** and 22 for all other approaches
Neural network for dimensionality reduction as pictured in Figure 7 = **NNDimRed**, dimensionality reduction using logistic functions, but same mathematical approach as for NNDimRed = **LogDimRed**, principal component analysis = **PCA**, linear discriminant analysis = **LDA**, feature selector based on univariate chi-squared statistic = **ChiSelect**, random projection = **RanProj**, no sparse variables = **NoSparse**, classifiers trained on original dataset (where possible) = **OrigData**
Adaptive boosting = **AdaBoost**, classification and regression trees = **CART**, extremely randomized trees = **ERT**, gradient boosting = **GB**, linear discriminant analysis = **LDA**, logistic regression = **LR**, multilayer perceptron = **MLP**, neural network for classification as pictured in Figure 8 = **NN**, random forest = **RF**, linear kernel support vector machine = **SVM**
Best classifier given dimensionality reduction algorithm is underlined. Best dimensionality reduction algorithm given classifier is in *italics*. The approaches proposed in this study are blue.

TABLE 16: Pearson's r between probabilistic out-of-sample predictions and actual class labels

We find that the neural network used for dimensionality reduction (see Figure 7) consistently outperforms all other dimensionality reduction algorithm considered, almost regardless of the classifier being used. The only exception is the gradient boosting classifier: The outperformance of gradient boosting + neural network used for dimensionality reduction over gradient boosting + the logistic benchmark (highlighted in Table 17) is only significant at the 5%-level. The neural network used for classification (see Figure 8) performs slightly better than other classification algorithms and consistently better than the standard multilayer perceptron, even though the improvement is small by comparison. Of any combination of a dimensionality reduction and classification algorithm, the neural networks proposed in this study achieve the highest predictive accuracy with a strong advantage over the best-performing methods taken from scikit-learn (combination of adaptive boosting and principal component analysis).

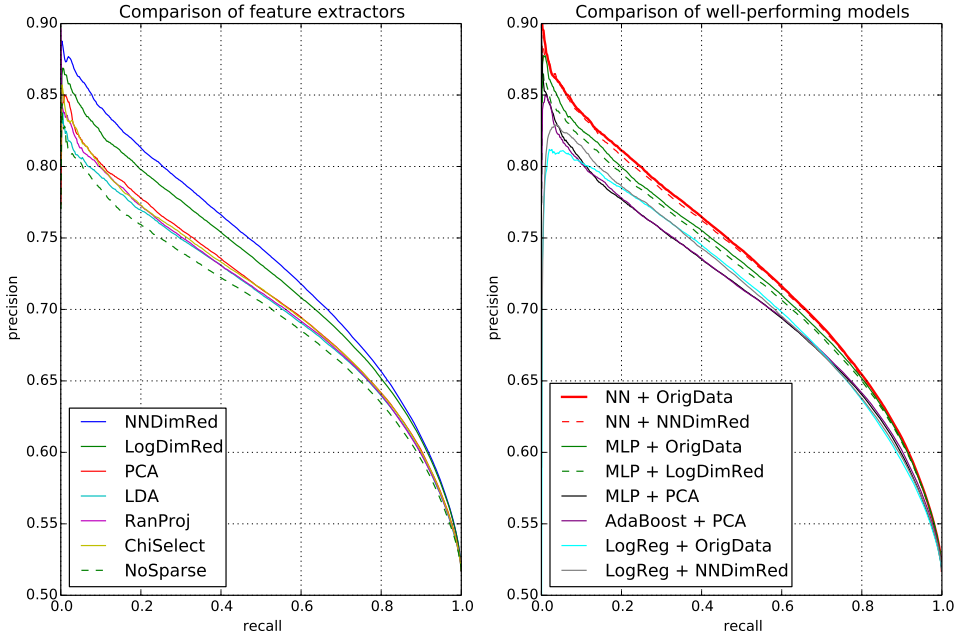
The combined neural network (see Figure 9) is also trained on the original dataset using the

weights obtained from the separate training sessions for initialization. For comparison, we undertake the same fine-tuning step for the combination of the multilayer perceptron and the logistic dimensionality reduction we used as a benchmark , which results in a multilayer perceptron. We find that fine-tuning yields slight improvement to both models.

	NN	**Ada-Boost**	**CART**	**ERT**	**GB**	**LDA**	**LR**	**MLP**	**RF**	**SVM**
LogDim-Red	0.000	0.000	0.000	0.000	***0.045***	0.000	0.000	0.000	0.000	0.000
PCA	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
LDA	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ChiSelect	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RanProj	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
NoSparse	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
OrigData	-	-	-	-	-	-	0.000	0.000	-	0.000

Note: Outperformance of **NNDimRed** was compared to other dimensionality reduction techniques given classifier. For **NN + OrigData** it was calculated in comparison to **LR**, **MLP** and **SVM** other given **OrigData**.

TABLE 17: Statistical significance (p-value) of outperformance given classifier



Note: This figure was created using matplotlib (Hunter, 2007).

FIGURE 10: Trade-off between precision and recall for selected models

We also measure the statistical significance of the neural network used for dimensionality reduction see Figure 7) when corrected for all other dimensionality reduction algorithms applied to the same classifier. The purpose of this procedure is to test whether the approach

provides genuinely new information or the same prediction could have been generated by stacking extant methods. Results are shown in Table 18. The results indicate high statistical significance, demonstrating that the approach generates genuinely new information.

	NN	**Ada-Boost**	**CART**	**ERT**	**GB**	**LDA**	**LR**	**MLP**	**RF**	**SVM**
NNDim-Red	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

TABLE 18: Statistical significance (p-value) of predictive performance when corrected for all other feature extraction combined with the same classifier

We corroborate these results using precision and recall as additional measures. We first compare the neural network used for dimensionality reduction (see Figure 7) to all other dimensionality reduction algorithm using adaptive boosting as an example classifier. Afterwards, we compare the models trained on the original datasets to a selection of well-performing standard models. Results are depicted in Figure 10.

These results demonstrate the superiority of the dimensionality reduction method developed in this study. Given any level of recall, the approach offers the best precision and is therefore the most attractive approach to any decision maker, regardless of his or her preferences regarding precision and recall. They also show that failing to include the sparse features significantly reduces predictive accuracy (performance of the neural network used for dimensionality reduction is shown as a thick blue line, the model solely based on dense features is shown as a dashed green line, see Figure 10, left hand side).

Both the combined neural network as well as the multilayer perceptron display a better trade-off between recall and precision after fine-tuning than before (performance before fine-tuning is displayed as a dashed line of the same color as the fine-tuned model, see Figure 10, right hand side). Nonetheless, even before fine-tuning, the combined neural network is strictly preferable to the fine-tuned multilayer perceptron.

4.6.2 Interpretability

The neural network for classification (see Figure 8) has only 40 input nodes. Therefore, techniques such as the rule extraction method proposed by Benítez et al. (1997) are readily applicable to this neural network as well as its fine-tuned version. This results in 25 rules, each of which contains two linear equations connected by an AND-gate (not shown due to space restrictions).

The neural network used for dimensionality reduction (see Figure 7) can be interpreted in two different ways, depending on the desired level of detail: First, by studying the effect sizes of each of the extracted features using Pearson's r (following the advice of Cohen (1992)) and second, by studying the interactions more closely.

The effect sizes of each of the extracted features are presented in Table 19. Studying the impact factors demonstrates that interactions on the level of product category, sales channel and product subcategory have the strongest impact on the probability of a product being returned. These interactions have much greater impact than the direct effects of the product itself. This implies that rather than there being high-return and low-return product categories, there are high-return and low-return consumption patterns.

Indicator	**Effect size**	
	Product itself	**Interactions**
Hour of the day at which product was ordered	-0.007	-
Platform from which product was ordered	-0.0003	-
Device from which product was ordered	-0.0001	-
Operating system from which product was ordered	-0.001	-
Web browser from which product was ordered	0.139	-
Payment method used	-0.099	-
Product *brand*	0.063	0.032
Product *category*	0.144	-0.298
Product *subcategory*	-0.051	0.179
Target group product is designed for	0.099	-0.080
Activity product is designed for	-0.074	-0.097
Product *size*	0.002	0.013
Sales channel from which product was reached	-0.027	-0.233
Product *color*	0.057	0.078

TABLE 19: Effect size (Pearson's r) of extracted features

We demonstrate how the interactions can be interpreted using one example interaction. Since the 30 interactions on the level of product category have the highest total impact (see Table 19), we choose the most impactful interaction among these 30 interactions, which turns out to be an interaction between the product categories in the same virtual shopping basket. This interaction is displayed in Table 20.

Ten highest weights		**Ten lowest weights**	
shoes	4.10	skirts	-0.81
vests	3.85	camping	-0.94
hats	3.75	underpants	-0.94
bags	3.57	climbing gear	-1.00
jumpers	3.17	brassiers	-1.01
dresses	3.08	protective gear	-1.12
inlines	2.15	flashlights	-1.19
inline skates	2.09	snowboards	-1.21
skates	2.08	suits	-1.34
mattresses	1.02	gloves	-2.82
Note: The table displays interactions between product categories that have been placed in the same basket. A high weight implies a higher degree of belonging to the fuzzy group and products that belong to this fuzzy group have a positive impact on each others' return probability. Due to space restrictions only the ten highest and ten lowest out of 74 weights are shown.			

TABLE 20: Example for an interaction table

Because of the AND-Gate and the weight-sharing architecture of the neural network used for dimensionality reduction (see Figure 7), the categories can be interpreted as fuzzy groups. The weights indicate the degree of belonging of that category to a fuzzy group. Since "shoes" and "vests" both receive high weights, having a vest in the same virtual shopping basket will increase the return probability of a pair of shoes by almost as much as having another pair of shoes in the same shopping basket. Because the category "gloves" receives a negative weight in that same interaction, a glove in the basket will reduce the likelihood of shoes and vests being returned.

4.7 Discussion and Conclusion

The purpose of this study was to develop an approach for extracting interpretable features from very high-dimensional business datasets.

A number of stakeholders stand to benefit from the approach. Management researchers can interpret the highly aggregated features, as presented in Table 19, to extract theoretically useful insight from large-scale, high dimensional datasets. Practitioners looking to develop targeted strategies for influencing customer behavior will find the highly fine-grained approach, as presented in Table 20, more compelling. Both groups will stand benefit from the superior predictive accuracy provided by the model.

In fact, the increase in predictive accuracy achieved by the algorithms proposed in this study is of considerable practical significance: If we were to design a system that intervenes whenever the likelihood of a product being returned is in excess of 80% (in other words, design a predictive system that has a precision of about 80%), the combined neural network would achieve a recall of close to 25%. The multilayer perceptron trained on the original data would achieve a recall of less than 20% and the combination of adaptive boosting and principal component analysis (the best performing model solely based on algorithm implemented in scikit-learn, see Table 16) would achieve a recall of just over 13% (see Figure 10, right hand side).

Another interesting finding is the performance of linear models: When compared to the combination of principal component analysis and adaptive boosting, the logistic regression achieves a higher precision given most levels of recall. However, it does poorly for the segment of products with a very high likelihood of being returned, which is the most important segment for this particular business case (Urbanke et al., 2015a). The poor performance in this segment can be reduced by using the neural network for dimensionality reduction (see Figure 7) instead of training the logistic regression on the entire dataset. These results imply that in order to understand behavioral patterns related to a very high return probability, we need to understand non-linear interactions. Linear models, as are generally used for quantitative analyses, will fail to capture relevant information for the segment that is most important to business practitioners.

The most important limitation of this study is that due to space restrictions and the complexity of the data model required for this approach, our findings are based on a single business problem. However, it is very likely that the lessons learned from this study can be applied to other business problems as well: Neither the example dataset used in this study nor the business case are unusual. In fact, most business analytics datasets are heterogenous combining numeric and nominal indicators. An increasing number of businesses have access to massively fine-grained data consisting of millions of transactions (Martens et al., forthcom-

ing). The general approach in this study, including the differention between dense and sparse features, is therefore applicable to most business analytics settings:

1. Identify potential interactions for the nominal indicators contained in the dataset (for instance customers' previous purchases).
2. For each of the possible manifestations of the nominal variables, define a variable counting the number of times this manifestation has occurred (for instance the number of times a customer has previously purchased a product of category X). Call these sparse variables.
3. For each of the nominal indicators, combine the interactions in several AND-gates as depicted in Figure 7. Model direct effects, if you believe them to be relevant.
4. Train the resulting neural network using the statistical approach presented in this study.
5. Rescale the extracted and dense features and train another neural network on these features.
6. Combine the neural networks as depicted in Figure 9 and fine-tune.

Such an approach requires business knowledge about potential interactions and a deeper understanding of the underlying business problem. However, our findings show that this approach can significantly improve predictive performance. Information systems researchers' comparative advantage of combining an understanding of technical issues, data management and business processes (Agarwal and Dhar, 2014) is important. Given appropriate tools, such as the one developed in this study, information systems researchers can leverage this advantage to achieve superior predictive accuracy and a deeper understanding of important business problems.

5 An Interpretable Machine Learning Algorithm Based On Randomized Neural Networks

Abstract

Interpretable machine learning algorithms are important for many advanced business intelligence problems. The aim of interpretable machine learning algorithms is to support decision makers by providing an understanding of complex relationships within a dataset. In this study, we develop an interpretable machine learning algorithm based on randomized neural networks. We illustrate its usefulness as a decision support system by applying it to a dataset related to product returns in e-commerce and demonstrate how it can be leveraged to generate actionable insight for industry practitioners. We also compare its predictive accuracy with widely used interpretable machine learning algorithms showing that its predictive performance on our example dataset is comparable to non-interpretable approaches and outperforms other interpretable approaches as well as standard randomized neural networks.

Keywords: Machine learning, Business intelligence, E-commerce, Online retail, Neural networks, Decision support

5.1 Introduction

This study was motivated by our cooperation with an online retailer suffering from a product return rate of over 50%. The purpose of these projects was to develop a system that can predict product returns before the customer has even hit the "order"-button, so that the online retailer can intervene, if necessary. However, in the course of these projects we found that standard machine learning algorithms often fail to meet the requirements of real-world decision makers and business intelligence experts. Standard machine learning algorithms are generally "black box" approaches, which are very useful for prediction, but cannot be interpreted, even by machine learning experts. However, our industry partners required "glass box" algorithms that would also inform them as to *why* customers return products purchased online, in addition to providing an accurate prediction. They wanted to gain a deeper understanding of how different factors influence product returns and interact with each other, so they could develop new strategies to address this important problem.

We argue that interpretability can greatly increase the usefulness of an algorithm for business practitioners in general. The purpose of a decision support system is to inform decision makers by presenting complex information in a way that they can understand (Bonczek et al., 2014; Power et al., 2002; Sprague Jr, 1980). However, few machine learning algorithms meet this requirement.

The purpose of this study is to make a contribution to closing this gap. In order to do so, we build on and extend the existing literature on randomized neural networks (RNN).[4] RNNs have been successfully applied to a large variety of domains, including sales forecasting

[4]Note that in the more recent literature, randomized neural networks are often referred to as "Extreme Learning Machines" or "ELM". It has been demonstrated that the authors introducing this term have plagiarized extant studies and the term is now considered to be unethical (Wang and Wan, 2008). We therefore do not use this term and studies introducing the unethical term are not cited.

(Choi et al., 2014; Sun et al., 2008), forecasting wind power output (Wan et al., 2014) or predicting the monsoon rainfall (Acharya et al., 2014).

In this study, we introduce an algorithm which summarizes larger datasets using several RNNs. These RNNs can be visualized using techniques such as contour plots or heat maps. Each RNN is trained to maximize non-redundant information content. Decision makers or business intelligence experts using the algorithm do not have to understand the mathematics involved in the creation of the RNNs. It is fully sufficient to be able to interpret standard visualization techniques. We therefore argue that the algorithm introduced in this study has the potential for significant practical relevance for business intelligence across many different domains and industries.

We illustrate how the algorithm can be useful to business practitioners by using our motivating example, namely product returns in online retail. We apply the algorithm to a dataset obtained from a major German online retailer containing a total of 3,637,654 transactions and demonstrate that the algorithm can be used to identify the main factors influencing product returns and how they interact with each other. We then compare its predictive performance to extant approaches and demonstrate that the predictive accuracy of our own approach is comparable to non-interpretable machine learning algorithms, outperforms existing interpretable approaches and is even slightly superior to standard RNNs.

The remainder of this study is organized as follows: In Section 2, we provide background on the algorithm developed in this study. In Section 3, we introduce the approach itself. In Section 4, we explain the methods employed for conducting this study as well as the dataset used for evaluation. In Section 5, we present the results of our evaluation. In Section 6, we discuss our results and conclude.

5.2 Background

In this section, we provide some background on RNNs as well as interpretable machine learning. Then we explain how our own study contributes to closing existing research gaps in these areas.

5.2.1 Randomized neural networks

Research on RNNs dates back to Rosenblatt (1958), who develops a perceptron to model the human brain. Important advances were made by Lowe (1989), who introduces radial basis functions (RBF) to the RNN literature as well as Broomhead and Lowe (1988), who propose to solve the resulting linear system using the Moore-Penrose-pseudoinverse rather than numeric techniques, which is the standard approach in neural network research. Performance can be improved by using supervised techniques when consciously selecting the centers of the RBFs rather than sampling randomly (Chen et al., 1991; Wettschereck and Dietterich, 1991). It can be theoretically shown that single-layer RBF neural networks have the capability for universal approximation, if the same variance factor is applied to each of the RBF (Park and Sandberg, 1991). It has been experimentally demonstrated that RNNs can achieve greater predictive accuracy in less training time than single-layer neural networks trained using backpropagation (Pao and Takefji, 1992). In a large-scale comparative study, RNNs have been demonstrated to be among the best performers and are the machine learning algorithm with the highest probabilty of achieving the maximum accuracies (Fernández-Delgado et al., 2014).

5.2.2 Interpretable machine learning

Most machine learning algorithms, including RNNs, are considered to be "black box" approaches, which are not interpretable. However, there are some "glass box" algorithms which produce results that can be interpreted in a meaningful way. These include linear models, genetic rule-based algorithms, decision trees and machine learning algorithms embedded in structural equation models.

Linear models are based on a simple, linear transformation of the input factors and can be used for classification or regression (Bryk and Raudenbush, 1992; Neter et al., 1996; Searle, 2012). In addition to being easily interpretable, they are also computationally cheap and very scalable making them easily applicable to large-scale datasets (Agarwal et al., 2014).

A less popular class of interpretable machine learning algorithms are evolutionary rule-based systems, which summarize problems into a set of interpretable decision rules. These rules are trained using genetic approaches. Implementations include SIA (Venturini, 1993), XCS (Wilson, 1995, 2000), NOW G-net (Anglano and Botta, 2002), UCS (Bernadó-Mansilla and Garrell-Guiu, 2003), OCEC (Jiao et al., 2006), GASSIST (Bacardit and Garrell, 2007) and HIDER (Aguilar-Ruiz et al., 2007).

Most genetic algorithms fall into the two categories of the Pitt approach and the Michigan approach: In the Pitt approach, a single genetic entity is a predefined number of rules and the algorithm aims to train these rules as whole. In the Michigan approach, a genetic entity consists of a single rule and the goal of genetic optimization is to look for a number of efficient rule. Whereas the former approach tends to produce rules that are more adapted to each other, the latter has a smaller search space and therefore tends to converge faster. (Venturini, 1993; Corcoran and Sen, 1994)

The main idea of *SIA (supervised inductive algorithm)* (Venturini, 1993) is to by only allow existing rules to be generalized. This drastically reduces the search space in comparison to either the Pitts or the Michigan approach. Corcoran and Sen (1994) propose a Pitt-style approach that allows for three types of variations on the rules: Crossover, creep mutation and simple random mutation. Under the crossover approach, the genes successful rules are interchanged. Creep mutation varies the minimum and maximum threshold of each rule. Simple random mutation can replace the minimum and maximum threshold with random values. *XCS (extended classifier system)* (Wilson, 1995, 2000) updates rules based on a fitness measure, which is related to the relative accuracy of each rule. Rules are then crossed over with a probability predefined by the user. Newly generated classifiers are also compared to existing classifiers to make sure that there are no redundant rules. *NOW G-net (network of workstations genetic network)* (Anglano and Botta, 2002) adapts genetic rule-based learning to distributed computing: Each rule is selected for crossover with a probability proportional to its fitness. The rules are then separated across the nodes depending on the computational power each of the nodes possesses. *UCS (supervised classifier system)* (Bernadó-Mansilla and Garrell-Guiu, 2003) modifies XCS by replacing its reinforcement learning with a supervised scheme. *OCEC (organizational coevolutionary algorithm for classification)* (Jiao et al., 2006) differs from other genetic learning algorithms in that it uses a bottom-up scheme to build the rules. The standard approach to genetic rule-based learning is to generate random rules which are then amended through mutation or crossover. By contrast, OCEC first groups the training samples into clusters of similar attributes and guides the learning process on the basis of these clusters. This results in a significant reduction of runtime *GASSIST (genetic algorithms based classifier system)* (Bacardit and Garrell, 2007) has been developed to

solve the so-called bloat effect: The rules generated by the Pitt approach tend to be very long and overfitted to the training set. As a remedy, Bacardit and Garrell (2007) propose a regularization procedure that involves the *minimum description length* of a generated rule, which weighs the complexity of a rule against its accuracy. *HIDER (hierarchical decision rules)* (Aguilar-Ruiz et al., 2007) is similar to OCEC in that it guides the learning process by using examples from the training set. This results in simpler rules that are easier to understand.

Decision trees are a set of binary decision rules that are organized in a tree-like structure, where the outcome of a previous rule determines the next rule to be applied. Popular implementations include *CART (classification and regression trees)* (Breiman et al., 1984) and *C4.5* (Quinlan, 2014) or *CHAID (chi-squared automatic interaction detection)* (Kass, 1980).

All of these decision tree algorithms are greedy algorithms, which means that any change to the tree is generated considering only the improvement that this change generates, without regard to how subsequent changes might affect the tree. The algorithms differ in the impurity measure used to train the tree: *CART* (Breiman et al., 1984) uses the gini coeefficient. *C4.5* (Quinlan, 2014) uses information gain. *CHAID* (Kass, 1980) uses the chi-squared test statistic as the discriminating function.

Neural networks are often considered to be the quintessential example of a black box algorithm (Dreiseitl and Ohno-Machado, 2002; Elmolla et al., 2010; Minsky, 1991; Sjöberg et al., 1995; Suykens and Vandewalle, 2012). However, some approaches exist to understand the structure of neural networks. Some approaches (Jang and Sun, 1993; Benítez et al., 1997; Setiono et al., 2000) represent neural networks as fuzzy rule systems , which can be demonstrated to extract useful information from low-dimensional toy datasets (Benítez et al., 1997; Nauck and Kruse, 1997; Huang and Xing, 2002). Other approaches attempt to visualize the architecture of the neural network by representing the thickness of nodes in proportion to their weights (Özesmi and Özesmi, 1999). A relatively new approach is to identify slow and fixed points within the network, which also yields interpretable results on low-dimensional datasets (Sussillo and Barak, 2013).

Finally, it is possible to embed machine learning approaches into partial least squares path modeling frameworks. Partial least squares path modeling is a type of structural equation model (Wold, 1975; Lohmöller, 1989). Structural equation models represents are number of observed variables (called *manifest variables*) in the form of unobserved variables (called *latent variables*). Partial least squares path modeling differentiates between the *measurement model* and the *structural model*, which are typically trained iteratively (that is, one is held fixed while the other one is being optimized). The measurement model represents the relationship between the latent variables and their corresponding manifest variables whereas the structural model represents the relationship between the latent variables (Vinzi et al., 2010).

Partial least squares path modeling is not to be conflated with partial least squares regression (Wold, 1966), which is an approach to model the relationships between only two blocks of manifest variables. Partial least squares regression may, with some liberty, be regarded as the special case of partial least squares path modeling for which two blocks of manifest variables exist.

Whereas standard partial least square regression and partial least squares path modeling are linear algorithms measuring linear relationships between the variables, some approaches exist to embed machine learning algorithms into these models. The approaches typically replace the relationship between the latent variables (the *structural model* in the terminology of partial least squares path modeling) with non-linear machine learning algorithms, leaving

the *measurement model* unchanged. Since the relationship between the latent variables are low-dimensional by construction, the non-linear relationships in the structural model can be easily visualized.

To the best of our knowledge, the earliest of these approaches was proposed by Wold et al. (1989) and Frank (1990), both of whom model the relationships between latent variables in a partial least squares regression structure using a polynomial regression. Instead of a polynomial regression, it is also possible to use a spline function (Wold, 1992) or a neural network (Wilson et al., 1997). Another possibility is to use a fuzzy inference system for the structural model (Bang et al., 2002), which has the added benefit that the resulting rules are easily interpretable. All of the approaches mentioned so far only use non-linearities for modeling the relationship between the latent variables. However, the radial basis functions can also be used to enhance the relationship between the manifest and the latent variables (Li et al., 2005).

The approaches above were all developed to enhance partial least squares *regression*. However, it is also possible to use machine learning to enhance partial least squares *path modeling*. This can be achieved by integrating neural networks into the structural model (Buckler and Hennig-Thurau, 2008; Garbe and Richter, 2009). However, this approach is costly and it is therefore benefitial from a computational standpoint to use decision trees instead, which reduces predictive accuracy only slightly (Oztekin et al., 2011).

5.2.3 Research contributions

RNNs, as introduced in section 5.2.1, are "black box" approaches that cannot be used for interpretative purposes. The first contribution of this study is to modify the algorithm such that it can be visualized, thus making the resulting RNNs interpretable. Second, it is difficult to apply standard RNN approaches to larger datasets due to their memory requirement. In order to adress this problem, we propose an approach that truncates the randomized RBF in the hidden layer, which produces a sparse transformation and allows us to significantly reduce memory usage and training time.

Extant approaches to interpretable machine learning, as introduced in section 5.2.2, can be separated into four categories: Linear approaches, such as a linear or a logistic regression, and verbal approaches, such as decision trees, rule-based learning, which summarize relationships within data into verbal IF - THEN rules, visualization and embedding into structural equation models.

Unlike linear models, visualization can model non-linear relationships and is more effective than verbalization at communicating complex concepts (Keim et al., 2002). However, existing approaches to visualizing neural networks are either only applicable to toy datasets with a handful of dimensions or can even be misleading: The plots proposed by Buckler and Hennig-Thurau (2008) are in part based on arbitrarily set values. The arbitrarily set values can have significant impact on the shape of the plot. Moreover, the impact of unobserved third variables can suggest that there is a positive interaction between to variables when in fact there is not.

Thus, the third contribution of this study is to introduce an algorithm which achieves interpretability through visualization and would therefore be expected to model relationships between variables more effectively than linear or verbal approaches. The approach is to break down the classification problem into randomized network trained to low-dimensional subsets

of the data. Visualization of interaction of these subsets is feasible without the possibility of unobserved third variables or arbitrarily set values distorting the results.

5.3 Calculation

The algorithm proposed in this study is an ensemble of RNNs. The procedure of combining the RNNs is based on a novel statistical approach. Training a new RNN consists of three steps:

Step 1: Feature selection - select a small number (3-5) of features that form the basis of the new RNN. Choose them such that the information content *when corrected for existing* RNNs is maximized.

Step 2: Non-linear transformation - use a large number of random, non-linear transformations to lift the problem on a higher dimensionality.

Step 3: Linear reduction - reduce these dimensions to a single prediction using a linear reduction technique. Maximize the information content of the single prediction *when corrected for existing* RNN*s*.

For the remainder of this section, we introduce these three steps at greater mathematical detail.

5.3.1 Step 1: Feature selection

Let $\mathbf{X}$ be the data matrix containing the independent variables and $\mathbf{y}$ be the vector representing the dependent variable. Let $\mathbf{x_{:j}}$ be the feature represented by the j^{th} column in $\mathbf{X}$. The standard χ^2-feature selection technique uses Pearson's χ^2-statistic of each of the indicators $\mathbf{x_{:j}}$. The χ^2-statistic for each of the $\mathbf{x_{:j}}$ with regard to $\mathbf{y}$ is calculated and the k features with the highest χ^2-values are chosen. This approach is a very popular and commonly used technique for feature selection. However, the approach fails to take redundancies into account. If two features $\mathbf{x_{:j}}$ and $\mathbf{x_{:j'}}$ are highly correlated with each other in addition to strong predictive power for $\mathbf{y}$, including both features in the model might be inferior to including a different feature, which has less overall predictive power, but contains more non-redundant information. In our case, this is a particularly important shortcoming, as we would like to select a set of features that contains minimally redundant information when corrected for all previous RNNs. We therefore employ a statistical approach originally introduced by Urbanke et al. (2014) for the purposes of evaluating predictive methods to develop a modification to the standard χ^2-feature selection technique.

We define a scalar Z_j, which is the dot product between a feature $\mathbf{x_{:j}}$ and the vector of our dependant variable $\mathbf{y}$:

$$Z_j = \mathbf{x_{:j}^T y}. \tag{45}$$

Let x_{ij} be the i^{th} element in $\mathbf{x_{:j}}$ and y_i be the i^{th} element in $\mathbf{y}$. Under the null hypothesis that $\mathbf{x_{:j}}$ is uncorrelated to $\mathbf{y}$, the expected value of Z_j can be calculated as follows (Urbanke et al., 2014):

$$E(Z_j) = \frac{\sum_{i=1}^{N} x_{ij} \sum_{i=1}^{N} y_i}{N}. \tag{46}$$

Furthermore, the covariance of any two Z_j and $Z_{j'}$, which are related to $\mathbf{x_{:j}}$ and $\mathbf{x_{:j'}}$ respectively, can be calculated as follows (Urbanke et al., 2014):

$$cov(Z_j, Z_{j'}) = \frac{(N\sum_{i=1}^{N} x_{ij}x_{ij'} - \sum_{i=1}^{N} x_{ij}\sum_{k=1}^{N} x_{ij'})(N\sum_{j=1}^{N} y_i^2 - (\sum_{j=1}^{N} y_i)^2)}{N^2(N-1)}. \tag{47}$$

Let V_j be $cov(Z_j, Z_j)$, as defined in (47). Let $\mathbf{Z^{ctrl}}$ be a vector containing the scalars Z_j for a set of variables for which we would like to control $\mathbf{x_{\cdot j}}$. Let $\mathbf{V^{ctrl}}$ be their variance-covariance matrix calculated using (47). Let σ_j be the covariance vector of $\mathbf{Z^{ctrl}}$ and Z_j. Let V_j be the variance of Z_j. It can be demonstrated that the following test statistic can be modeled using a chi-squared distribution (Urbanke et al., 2014):

$$\begin{bmatrix} (\mathbf{Z^{ctrl}} - \mathbf{E}(\mathbf{Z^{ctrl}}))^{\mathbf{T}} & Z_j - E(Z_j) \end{bmatrix} \begin{bmatrix} \mathbf{V^{ctrl}} & \sigma_j \\ \sigma_j^T & V_j \end{bmatrix}^{-1} \tag{48}$$
$$\begin{bmatrix} (\mathbf{Z^{ctrl}} - \mathbf{E}(\mathbf{Z^{ctrl}}))Z_j - E(Z_j) \end{bmatrix}$$

$$-(\mathbf{Z^{ctrl}} - \mathbf{E}(\mathbf{Z^{ctrl}}))^{\mathbf{T}}(\mathbf{V^{ctrl}})^{-1}(\mathbf{Z^{ctrl}} - \mathbf{E}(\mathbf{Z^{ctrl}})) \sim \chi^2(1).$$

Based on this statistical approach, we propose a modification to the standard χ^2-feature selection technique, which takes redundant information into account. Let $\mathbf{Z^{RNNs}}$ be a vector containing the scalars Z for a set of RNNs which we have trained in previous iterations of the algorithm and for which we would like to control $\mathbf{x_{\cdot j}}$. Let $\mathbf{V^{RNNs}}$ be their variance-covariance matrix calculated using (47). The modified approach is described in pseudocode in Figure 11.

Set $\mathbf{J^{index}} := [1, 2, 3, ..., J]$
Set $\mathbf{J^{select}} := []$
Set $\mathbf{Z^{ctrl}}$=$\mathbf{Z^{RNNs}}$
Set $\mathbf{V^{ctrl}}$=$\mathbf{V^{RNNs}}$
Repeat J^{select} times:
  For all j in $\mathbf{J^{index}}$:
    Calculate χ_j^2 as defined in (45), (46), (47) and (48)
  Pick the j that maximises χ_j^2
  Set $\mathbf{Z^{ctrl}}$=$\begin{bmatrix} \mathbf{Z^{ctrl}} \\ Z_j \end{bmatrix}$
  Calculate $\mathbf{V^{ctrl}}$as defined in (47)
  Eliminate j from $\mathbf{J^{index}}$
  Add j to $\mathbf{J^{select}}$

FIGURE 11: Expression of the approach for selecting J^{select} features in pseudocode

This approach will yield a vector $\mathbf{J^{select}}$ of length J^{select}, which contains the indices j of the features $\mathbf{x_{\cdot j}}$ which form the basis of the next RNN. In order to keep the RNN interpretable, we do not let J^{select} exceed a value of 3 to 5.

5.3.2 Step 2: Non-Linear transformation using randomized, truncated RBFs

The resulting dataset $\mathbf{X^{select}}$ is, by construction, low-dimensional with $N >> J^{select}$. In order to capture non-linear relationships within the data, we use a large number of non-linear transformations to lift the dataset on a higher dimension. As the algorithm is developed for larger datasets with $N > 1,000,000$, a large number of transformations could easily cause technical problems as the size of the resulting dataset might exceed the limits of our computer's RAM.

In order to address this problem, we introduce the idea of *truncated radial basis functions*, which result in a sparse dataset. If stored using an appropriate data structure, the memory requirement of the transformations can be reduced significantly.

We therefore define a density δ (a reasonable value for which might be 5% or 1%) and a cut-off value $0 < \gamma < 1$ (a reasonable value for which might be 0.01). Let $\mathbf{x_{i:}}$ be the instance represented by the i^{th} line in $\mathbf{X}$. We define a radial basis function $rbf_k(\mathbf{x_{i:}})$ around a center $\mathbf{c_k}$ which is randomly sampled from our original dataset $\mathbf{X}$. Let c_{kj} be the j^{th} element in $\mathbf{c_k}$.

We calculate the Euclidian distance d_{ik} of each point $\mathbf{x_{i:}}$ from $\mathbf{c_k}$ with regard to every feature chosen during the feature selection process in Step 1, which we weight by the inverse variance of the corresponding feature, denoted as v_j:

$$d_{ik} = \sqrt{\sum_{j \in \mathbf{J^{select}}} \frac{\left(x_{ij} - c_{kj}\right)^2}{v_j}}. \tag{49}$$

We then order d_{ik} alongside i and find the δ^{tb} percentile of all d_{ik}, which we denote as $d_k^{critical}$. We define $rbf_k(\mathbf{x_{i:}})$ such that it assumes a value of 0, if d_{ik} exceeds $d_k^{critical}$. Moreover, in order to ensure a smooth rendering, we would like to $rbf_k(\mathbf{x_{i:}})$ to assume the (small) value of γ, if $d_{ik} = d_k^{critical}$. We therefore calculate a value ω_k such that it fulfills the following condition:

$$\gamma = \exp(\omega_k d_k^{critical}) \Leftrightarrow \omega_k = \frac{\ln \gamma}{d_k^{critical}}. \tag{50}$$

In summary, the non-linear transformation $rbf_k(\mathbf{x_{i:}})$ is defined as follows:

$$rbf_k(\mathbf{x_{i:}}) = \exp(\omega_k * d_{ik}), \text{ if } d_{ik} \leq d_k^{critical} \; rbf_k(\mathbf{x_{i:}}) = 0, \text{ if } d_{ik} > d_k^{critical}. \tag{51}$$

We calculate K such transformations. A reasonable value for K might be 2000. Because each $rbf_k(...)$ is constructed such that it will assume a non-zero value for a share of δ of all transformations, we can store the resulting matrix in CSR matrix format, which only stores the non-zeros values. This makes our approach applicable to larger datasets without requiring unreasonable amounts of RAM. The training algorithm for Step 2 is summarized in Figure 12.

For all j in $\mathbf{J^{select}}$:
  Calculate $\bar{x_j} := \frac{\sum_i x_{ij}}{N}$
  Calculate $v_j := \frac{\sum_i \left(x_{ij} - \bar{x_j}\right)^2}{N}$
For all k in $[1, 2, ..., K]$:
  Choose a random sample $\mathbf{c_k}$ from $\mathbf{X}$
  For all i in $[1, 2, ..., N]$:
    Calculate d_{ik} as defined in (49)
  Find $d_k^{critical} = \delta^{th}$ percentile of all d_{ik} alongside i in $[1, 2, ..., N]$
  Calculate ω_k using equation (50)

FIGURE 12: Expression of the non-linear transformations using randomized and truncated RBFs in pseudocode

5.3.3 Step 3: Linear reduction

The goal of the linear reduction step is to reduce the large number of non-linear transformations to a single prediction. This linear transformation should be conducted such that the

non-redundant information in the resulting prediction in maximized. In order to achieve this goal, we develop a modification of the approach developed by Urbanke et al. (2015b). In this approach, a number of linear transformation is introduced such that the transformations contain maximal predictive power for an dependant variable, but are minimally correlated with each other. However, Urbanke et al. (2015b) do not consider the possibility of control variables. This is very important, as we would like to maximize predictive power corrected for all previous RNNs, that is treating previous RNNs as control variables. In this section, we introduce a modification to the approach that fills this gap.

The main idea of the approach proposed by Urbanke et al. (2015b) is to define a set of extracted variables $\mathbf{X^{ext}} := \mathbf{XW}$ and then to optimize $\mathbf{W}$ such that the combined predictive power of $\mathbf{X^{ext}}$ for $\mathbf{Y}$ is maximized. This predictive power is measured using the same statistical approach we already used for Step 1:

$$Z_j^{ext} := (\mathbf{x_{:j}^{ext}})^{\mathbf{T}}\mathbf{y} \tag{52}$$

$$E(Z_j^{ext}) = \frac{\sum_{i=1}^N x_{ij}^{ext} \sum_{i=1}^N y_i}{N}$$

$$cov(Z_j^{ext}, Z_{j'}^{ext}) = \frac{(N\sum_{i=1}^N x_{ij}^{ext} x_{ij'}^{ext} - \sum_{i=1}^N x_{ij}^{ext} \sum_{k=1}^N x_{ij'}^{wxt})(N\sum_{j=1}^N y_i^2 - (\sum_{j=1}^N y_i)^2)}{N^2(N-1)}.$$

We adapt this approach to visual RNN-based based learning as follows: Let $\mathbf{X^{RNNs}}$ be the matrix representing the predictions of all previously trained RNNs. Our goal is to optimise a vector $\mathbf{w}$ such that the resulting vector $\mathbf{x^{ext}} := \sum_k w_k rbf_k(\mathbf{X})$ has maximal predictive power for $\mathbf{y}$ when controlled for $\mathbf{X^{RNNs}}$.

Equations (52) are applied to $\mathbf{X^{RNNs}}$ as well as σ, which is the covariance vector between Z^{ext} and $\mathbf{Z^{RNNs}}$.

It is then our goal to solve the following optimization problem:

$$\max_{\mathbf{w}} \left[\; (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}}))^{\mathbf{T}} \quad Z^{ext} - E(Z^{ext}) \; \right] \begin{bmatrix} \mathbf{V^{RNNs}} & \sigma \\ \sigma^T & V^{ext} \end{bmatrix}^{-1} \tag{53}$$

$$\left[\; (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}}))^{\mathbf{T}} \quad Z^{ext} - E(Z^{ext}) \; \right]$$

$$-(\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}}))^{\mathbf{T}} (\mathbf{V^{RNNs}})^{-\mathbf{1}} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}})).$$

Since the problem stated in (53) has to be optimized numerically, we calculate its gradient for the weight vector $\mathbf{w}$. This gradient can be calculated using one of the two following approaches.

The first approach is to directly take the derivative for each of the elements in $\mathbf{w}$:

$$\frac{\partial}{\partial w_k} \left[\; (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}}))^{\mathbf{T}} \quad Z^{ext} - E(Z^{ext}) \; \right] \begin{bmatrix} \mathbf{V^{RNNs}} & \sigma \\ \sigma^T & V^{ext} \end{bmatrix}^{-1} \tag{54}$$

$$\left[\; (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}})) \quad Z^{ext} - E(Z^{ext}) \; \right]$$

$$-(\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}}))^{\mathbf{T}} (\mathbf{V^{RNNs}})^{-1} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}}))$$

$$= 2 \left[\begin{array}{cc} \mathbf{0^T} & \frac{\partial Z^{ext}}{\partial w_k} - \frac{\partial E(Z^{ext})}{\partial w_k} \end{array} \right] \left[\begin{array}{cc} \mathbf{V^{RNNs}} & \sigma \\ \sigma^T & V^{ext} \end{array} \right]^{-1}$$
$$\left[\begin{array}{cc} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}})) & Z^{ext} - E(Z^{ext}) \end{array} \right]$$

$$- \left[\begin{array}{cc} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}}))^{\mathbf{T}} & (Z^{ext} - E(Z^{ext}))^T \end{array} \right] \left[\begin{array}{cc} \mathbf{V^{RNNs}} & \sigma \\ \sigma^T & V^{ext} \end{array} \right]^{-1}$$
$$\left[\begin{array}{cc} \mathbf{0} & \frac{\partial \sigma}{\partial w_k} \\ \frac{\partial \sigma^{\mathbf{T}}}{\partial w_k} & \frac{\partial V^{ext}}{\partial w_k} \end{array} \right] \left[\begin{array}{cc} \mathbf{V^{RNNs}} & \sigma \\ \sigma^T & V^{ext} \end{array} \right]^{-1}$$
$$\left[\begin{array}{cc} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}})) & Z^{ext} - E(Z^{ext}) \end{array} \right].$$

The second approach is to rewrite the optimization problem before taking the derivative. Using block-wise inversion of a matrix, we can easily verify that the following holds true (Urbanke et al., 2014):

$$\frac{\partial}{\partial w_k} \left[\begin{array}{cc} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}}))^{\mathbf{T}} & Z^{ext} - E(Z^{ext}) \end{array} \right] \tag{55}$$

$$\left[\begin{array}{cc} \mathbf{V^{RNNs}} & \sigma \\ \sigma^T & V^{ext} \end{array} \right]^{-1}$$
$$\left[\begin{array}{cc} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}})) & Z^{ext} - E(Z^{ext}) \end{array} \right]$$

$$-(\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}}))^{\mathbf{T}} (\mathbf{V^{RNNs}})^{-\mathbf{1}} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}}))$$

$$= \frac{\partial}{\partial w_k} \left(Z^{ext} - E(Z^{ext}) - \sigma^T (\mathbf{V^{RNNs}})^{-1} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}})) \right)^{\mathbf{T}}$$

$$\left(V^{ext} - \sigma^T (\mathbf{V^{RNNs}})^{-1} \sigma \right)^{-1}$$

$$\left(Z^{ext} - E(Z^{ext}) - \sigma^T (\mathbf{V^{RNNs}})^{-\mathbf{1}} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}})) \right)$$

$$= 2 \left(\frac{\partial Z^{ext}}{\partial w_k} - \frac{\partial E(Z^{ext})}{\partial w_k} - \frac{\partial \sigma^{\mathbf{T}}}{\partial w_k} (\mathbf{V^{RNNs}})^{-1} \right.$$
$$\left. (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}})) \right)^{\mathbf{T}} \left(V^{ext} - \sigma^T (\mathbf{V^{RNNs}})^{-1} \sigma \right)^{-1}$$

$$\left(Z^{ext} - E(Z^{ext}) - \sigma^T (\mathbf{V^{RNNs}})^{-\mathbf{1}} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}})) \right)$$
$$- \left(Z^{ext} - E(Z^{ext}) - \sigma^T (\mathbf{V^{RNNs}})^{-1} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}})) \right)^{\mathbf{T}}$$

$$\left(V^{ext} - \sigma^T (\mathbf{V^{RNNs}})^{-1} \sigma \right)^{-1} \left(\frac{\partial V^{ext}}{\partial w_k} - 2 \sigma^{\mathbf{T}} (\mathbf{V^{RNNs}})^{-1} \frac{\partial \sigma}{\partial w_k} \right)$$

$$\left(V^{ext} - \sigma^T (\mathbf{V^{RNNs}})^{-1} \sigma \right)^{-1}$$

$$\left(Z^{ext} - E(Z^{ext}) - \sigma^T (\mathbf{V^{RNNs}})^{-1} (\mathbf{Z^{RNNs}} - \mathbf{E}(\mathbf{Z^{RNNs}}))\right).$$

Even though (55) seems mathematically more complicated than (54), it is computationally more efficient. This becomes apparent, when considering the steps necessary to calculate the subsequent equations. For the first approach, we need to calculate the following: Z^{ext}, $E(Z^{ext})$, $\mathbf{Z^{RNNs}}$, $\mathbf{E}(\mathbf{Z^{RNNs}})$, $\begin{bmatrix} \mathbf{V^{RNNs}} & \sigma \\ \sigma^T & V^{ext} \end{bmatrix}^{-1}$, $\begin{bmatrix} \mathbf{0} & \frac{\partial \sigma}{\partial w_k} \\ \frac{\partial \sigma^{\mathbf{T}}}{\partial w_k} & \frac{\partial V^{ext}}{\partial w_k} \end{bmatrix}$, $\frac{\partial Z^{ext}}{\partial w_k}$ and $\frac{\partial E(Z^{ext})}{\partial w_k}$. Of these matrices, the following matrices do not depend on $\mathbf{w}$ and therefore do not have to be recalculated every time we update $\mathbf{w}$ in our numerical optimization: $\mathbf{Z^{RNNs}}$, $E(\mathbf{Z^{RNNs}})$, $\frac{\partial Z^{ext}}{\partial w_k}$ and $\frac{\partial E(Z^{ext})}{\partial w_k}$. The following matrices do depend on $\mathbf{w}$ and therefore have to be recalculated in every iteration: Z^{ext}, $E(Z^{ext})$, $\begin{bmatrix} \mathbf{V^{RNNs}} & \sigma \\ \sigma^T & V^{ext} \end{bmatrix}^{-1}$, $\begin{bmatrix} \mathbf{0} & \frac{\partial \sigma}{\partial w_k} \\ \frac{\partial \sigma^{\mathbf{T}}}{\partial w_k} & \frac{\partial V^{ext}}{\partial w_k} \end{bmatrix}$, and V^{ext}. To see that this is the case, consider that $\mathbf{Z^{RNNs}}$ and $\mathbf{E}(\mathbf{Z^{RNNs}})$ are control variables and therefore by definition unaffected by $\mathbf{w}$. Due to the linear nature of the transformation, the matrices $\frac{\partial Z^{ext}}{\partial w_k}$ and $\frac{\partial E(Z^{ext})}{\partial w_k}$ do also not depend on $\mathbf{w}$. See (55) for more details.

Consider, by contrast, the second approach, for which we need to calculate Z^{ext}, $E(Z^{ext})$, $\mathbf{Z^{RNNs}}$, $\mathbf{E}(\mathbf{Z^{RNNs}})$, $\left(V^{ext} - \sigma^T (\mathbf{V^{RNNs}})^{-1} \sigma\right)^{-1}$, $(\mathbf{V^{RNNs}})^{-1}$,σ,$\frac{\partial Z^{ext}}{\partial w_k}$, $\frac{\partial E(Z^{ext})}{\partial w_k}$, $\frac{\partial V^{ext}}{\partial w_k}$ and $\frac{\partial \sigma}{\partial w_k}$. Of these matrices, the following matrices do not depend on $\mathbf{w}$ and therefore do not have to be recalculated every time we update $\mathbf{w}$ in our numerical optimization: $\mathbf{Z^{RNNs}}$, $\mathbf{E}(\mathbf{Z^{RNNs}})$, $(\mathbf{V^{RNNs}})^{-1}$, $\frac{\partial Z^{ext}}{\partial w_k}$ and $\frac{\partial E(Z^{ext})}{\partial w_k}$. The following matrices do depend on $\mathbf{w}$ and therefore have to be recalculated in every iteration: Z^{ext}, $E(Z^{ext})$,$\left(V^{ext} - \sigma^T (\mathbf{V^{RNNs}})^{-1} \sigma\right)^{-1}$,$\sigma$, $\frac{\partial \sigma}{\partial w_k}$ and $\frac{\partial V^{ext}}{\partial w_k}$. The crucial difference is that $\left(V^{ext} - \sigma^T (\mathbf{V^{RNNs}})^{-1} \sigma\right)^{-1}$ is a (1×1)-matrix, whereas $\begin{bmatrix} \mathbf{V^{RNNs}} & \sigma \\ \sigma^T & V^{ext} \end{bmatrix}^{-1}$ is $\left(J^{RNNs} + 1 \times J^{RNNs} + 1\right)$-matrix. The computational complexity of matrix inversions is $\mathcal{O}\left(n^3\right)$ in the number dimensions. In effect, the second approach allows us to replace a computationally expensive operation, that of inverting a $\left(J^{RNNs} + 1 \times J^{RNNs} + 1\right)$-matrix after every update of $\mathbf{w}$, with a computationally considerably cheaper approach, that of inverting a (1×1)-matrix after every update of $\mathbf{w}$.

Recall that the calculation of Z^{ext}, $E(Z^{ext})$ and V^{ext} is provided in (52). The calculation of $\mathbf{Z^{RNNs}}$, $\mathbf{E}(\mathbf{Z^{RNNs}})$, $(\mathbf{V^{RNNs}})^{-1}$ is provided in (45), (46) and (47) respectively. This is implies that we still have to explicitly derive the following values: σ,$\frac{\partial \sigma}{\partial w_k}$,$\frac{\partial Z^{ext}}{\partial w_k}$, $\frac{\partial E(Z^{ext})}{\partial w_k}$ and $\frac{\partial V^{ext}}{\partial w_k}$.

Applying (47) to σ yields the following:

$$cov(z_j^{RNNs}, z^{ext}) = \left(N \sum_{i=1}^N y_i^2 - (\sum_{i=1}^N y_i)^2\right) * \frac{\left(N \sum_{i=1}^N x_{ij}^{RNNs} x_i^{ext} - \sum_{i=1}^N x_{ij}^{RNNs} \sum_{i=1}^N x_i^{ext}\right)}{N^2(N-1)}. \tag{56}$$

Taking into account the definition of x_i^{ext}, the partial derivative of any element of σ with respect to a single element w_k, hence $\frac{\partial \sigma}{\partial w_k}$, is calculated as follows:

$$\frac{\partial cov(z_j^{RNNs}, z^{ext})}{\partial w_k} = \left(N \sum_{i=1}^N y_i^2 - (\sum_{i=1}^N y_i)^2\right) * \frac{\left(N \sum_{i=1}^N x_{ij}^{RNNs} rbf_k(\mathbf{x_i}) - \sum_{i=1}^N x_{ij}^{RNNs} \sum_{i=1}^N rbf_k(\mathbf{x_i})\right)}{N^2(N-1)}. \tag{57}$$

Following Urbanke et al. (2015b), we calculate $\frac{\partial \mathbf{Z^{ext}}}{\partial w_k}$, $\frac{\partial E(\mathbf{Z^{ext}})}{\partial w_k}$ and $\frac{\partial \mathbf{V^{ext}}}{\partial w_k}$ as follows:

$$\frac{\partial z^{ext}}{\partial w_k} = \sum_{i=1}^N rbf_k(\mathbf{x_i}) y_i \tag{58}$$

$$\frac{\partial E(z^{ext})}{\partial w_k} = \frac{\sum_{i=1}^{N} rbf_k(\mathbf{x_i}) \sum_{i=1}^{N} y_{ik}}{N}$$

$$\frac{\partial var(Z^{ext})}{\partial w_k} = 2 * \left(N \sum_{i=1}^{N} y_i^2 - (\sum_{i=1}^{N} y_i)^2\right) * \frac{\left(N \sum_{i=1}^{N} rbf_k(\mathbf{x_i}) x_i^{ext} - \sum_{i=1}^{N} rbf_k(\mathbf{x_i}) \sum_{i=1}^{N} x_i^{ext}\right)}{N^2(N-1)}.$$

In summary, the sufficient statistics needed are as follows: $\sum_{i=1}^{N} y_i$, $\sum_{i=1}^{N} y_i^2$, $\sum_{i=1}^{N} rbf_k(\mathbf{x_i})$, $\sum_{i=1}^{N} rbf_k(\mathbf{x_i}) y_i$, $\sum_{i=1}^{N} x_{ij}^{RNNs}$, $\sum_{i=1}^{N} x_{ij}^{RNNs} y_i$, $\sum_{i=1}^{N} x_{ij}^{RNNs} x_{ij'}^{RNNs}$, $\sum_{i=1}^{N} x_{ij}^{RNNs} rbf_k(\mathbf{x_i})$, $\sum_i x_i^{ext}$, $\sum_i x_i^{ext} y_i$, $\sum_i \left(x_{id}^{ext}\right)^2$, $\sum_i rbf_k(\mathbf{x_i}) x_i^{ext}$ and $\sum_i x_{ij}^{RNNs} x_i^{ext}$, the latter five of which include x_i^{ext}, are therefore dependent on $\mathbf{w}$ and need to be recalculated after every update. This results in the training algorithm for Step 3 presented in Figure 13.

Calculate $\sum_{i=1}^{N} y_i$
Calculate $\sum_{i=1}^{N} y_i^2$
Calculate $\sum_{i=1}^{N} rbf_k(\mathbf{x_{i:}})$ for all k
Calculate $\sum_{i=1}^{N} rbf_k(\mathbf{x_{i:}}) y_i$ for all k
Calculate $\sum_{i=1}^{N} x_{ij}^{RNNs}$ for all j
Calculate $\sum_{i=1}^{N} x_{ij}^{RNNs} y_i$ for all j
Calculate $\sum_{i=1}^{N} x_{ij}^{RNNs} x_{ij'}^{RNNs}$ for all j, j'
Calculate $\sum_{i=1}^{N} x_{ij}^{RNNs} rbf_k(\mathbf{x_{i:}})$ for all j, k
Calculate $\mathbf{Z^{RNNs}}$ as defined in (45)
Calculate $E(\mathbf{Z^{RNNs}})$ as defined in (46)
Calculate$(\mathbf{V^{RNNs}})^{-1}$ as defined in (47)
Calculate $\frac{\partial \mathbf{Z^{ext}}}{\partial w_k}$ and $\frac{\partial E(\mathbf{Z^{ext}})}{\partial w_k}$ as defined in (58)
Calculate $\frac{\partial \sigma}{\partial w_k}$ as defined in (52)
Repeat until convergence:
    Calculate $x_i^{ext} = \sum_{k=1}^{K} rbf_k(\mathbf{x_{i:}}) w_k$ for all i
    Calculate $\sum_i x_i^{ext}$
    Calculate $\sum_i x_i^{ext} y_i$
    Calculate $\sum_i \left(x_{id}^{ext}\right)^2$
    Calculate $\sum_i rbf_k(\mathbf{x_{i:}}) x_i^{ext}$ for all k
    Calculate $\sum_i x_{ij}^{RNNs} x_i^{ext}$ for all j
    Calculate σ as defined in (56)
    Calculate Z^{ext}, $E(Z^{ext})$,$\left(V^{ext} - \sigma^T (\mathbf{V^{RNNs}})^{-1} \sigma\right)^{-1}$ as defined in (52)
    Calculate $\frac{\partial V^{ext}}{\partial w_k}$ as defined in (58)
    Calculate gradient as defined in (55)
    Based on gradient, update $\mathbf{w}$

FIGURE 13: Expression of the approach for linear dimensionality reduction in pseudocode

5.3.4 Visualization

Because each of the resulting RNNs is by construction based on a small number of dimensions, it can be effectively visualized using an interactive contour plot or an interactive heat map. Additional dimensions can be modeled using a slider.

5.4 Material and methods

In this section, we briefly describe how we implement the algorithm introduced in this study, provide some background on our motivating case and describe our example dataset at greater detail.

5.4.1 Implementation

We implement the algorithms described in section 5.3 in C++ and write an interface to Python using SWIG (Beazley, 1996). We visualize the resulting RNNs using Mayavi (Ramachandran and Varoquaux, 2011). We compare our approach to a choice of algorithms implemented in the widely used Python library scikit-learn (Pedregosa et al., 2011). All of the results in this study are based either on self-written or open-source software.

5.4.2 Product returns in online retail

Product returns have been recognized as an important factor impacting the profitability of many online retailers (Grewal et al., 2004). They can have numerous causes, such as impulse shopping (LaRose, 2001; Rabinovich et al., 2011) or fraudulent behaviour on behalf of the customers (Autry et al., 2007; Harris, 2010; Wachter et al., 2012). However, it is also an important part of e-tailers' business model as it has been shown to reduce the risks customers take (Anderson et al., 2009; Li et al., 2013; Wood, 2001) and can positively influence customers' willingness to purchase products (Autry, 2005; Bower and Maxham III, 2012; Petersen and Kumar, 2009). It is well-established industry knowledge that customers like to buy several similar products of different sizes to see which one fits best and return those they do not like, [5] behavior we will refer to as *strategic shopping*.

5.4.3 Dataset

We use a dataset obtained from a major German online retailer specializing in fashion. It contains a total of 3,637,654 transactions from July 2014 to May 2015. The task is to predict whether a particular product ordered by a particular customer in a particular virtual shopping will be returned.

In the case background presented in section 5.4.2, we have identified impulse and strategic shopping as key factors influencing product returns. Both of these factors suggest that the composition of the virtual shopping basket will have an impact on product returns. Based on these considerations, we have created numerous indicators related to other products in the shoppung basket.

The indicators contained in the dataset can be separated into four categories:

1. Indicators describing the product itself - These include *ProductPrice1*, *ProductPrice2* and *NumberOfPictures*. The difference between *ProductPrice1* and *ProductPrice2* is that the former denotes the actual price at purchase whereas the latter denotes the standard price of the product. *NumberOfPictures* denotes the number of picture the

[5] http://www.informationweek.com/e-commerce/uk-online-retailers-struggle-with-product-returns/d/d-id/1108423?, accessed 2015-10-31
http://www.wsj.com/articles/SB10001424052702304773104579270260683155216, accessed 2015-10-31

product is advertised with.

Feature	**Description**
ProductPrice1	Actual price at purchase
ProductPrice2	Standard product price
NumberOfExactSame ProductsInBasket	The number of times a customer has placed the exact same product in the virtual shopping basket. Must at least be one.
NumberOfProducts InBasket	The total number of products a customer has placed in the virtual shopping basket
SameCategory TargetGroup	The number of products in the virtual shopping basket that have the same category and target group as this product
SameSubcategory TargetGroup	The number of products in the virtual shopping basket that have the same subcategory and target group as this product
SimilarProduct	The number of products in the virtual shopping basket that have a product ID similar to this product
SameBrandTarget Group	The number of products in the virtual shopping basket that have the same brand and target group as this product
OnlyColorDiff	The number of products in the virtual shopping basket that have the exact same features as this product, with the exception of color
OnlySizeDiff	The number of products in the virtual shopping basket that have the exact same features as this product, with the exception of size
SameCategory	The number of products in the virtual shopping basket that have the same category as this product
SameTargetGroup	The number of products in the virtual shopping basket that have the same target group as this product
SameSubcategory	The number of products in the virtual shopping basket that have the same subcategory as this product
SameActivity	The number of products in the virtual shopping basket that were designed for the same activity as this product
SameCategoryActivity	The number of products in the virtual shopping basket that have the same category and were designed for the same activity as this product
SamePath	The number of products in the virtual shopping basket that were purchased through the same path
NumProducts PreviouslyPurchased	The number of products a customer has previously purchased
CustomerPrevious ReturnRate	The share of products a customer has previously returned. Assumes a value of 0, if customer has not purchased anything before.
NumberOfPictures	The number of pictures the product is advertised with
Platform	The platform (desktop/notebook/mobile) from which the product was ordered
Hour	The hour of the day on which the product was ordered
PaymentMethod	The payment method with which the product was paid for

TABLE 21: Overview of the indicators used

2. Indicators describing the transaction - These include *Platform* (whether the product was ordered from a desktop, a mobile or a tablet), *Hour* (on which hour of the day the transaction took place) and *PaymentMethod*.

3. Indicators describing previous customer behavior - These include *NumProductsPreviouslyPurchased* and *CustomerPreviousReturnRate*.

4. Indicators describing other products in the same basket - The online retailer we are cooperating with uses an extensive system to categorize its products. Based on these categories, we define indicators that count the number of products in the basket that share one or several traits with the product. These indicators include *NumberOfExactSameProductsInBasket*, *NumberOfProductsInBasket*, *SameCategoryTargetGroup*, *SameSubcategoryTargetGroup*, *SameSubcategoryTargetGroup* , *SimilarProduct* (based on product ID), *SameBrandTargetGroup*, *OnlyColorDiff*, *OnlySizeDiff*, *SameCategory*, *SameTargetGroup*, *SameSubcategory*, *SameActivity*, *SameCategoryActivity* and *SamePath*.

   An overview is provided in Table 21.

RNN	**Features chosen**	χ^2	**p-value**
1	*CustomerPreviousReturnRate*, *SameCategoryTargetGroup*, *NumberOfProductsInBasket*	1678.61	0.000
2	*OnlyColorDiff*, *PaymentMethod*, *NumberOfPictures*	4.88	0.027
3	*ProductPrice1*, *SameCategoryActivity*, *SameCategory*	401.20	0.000
4	*SameTargetGroup*, *NumberOfPictures*, *NumberOfExactSameProductsInBasket*	26966.99	0.000
5	*CustomerPreviousReturnRate*, *OnlySizeDiff*, *SameActivity*	7800.60	0.000
6	*SameCategory*, *SameBrandTargetGroup*, *PaymentMethod*	487.77	0.000
7	*NumberOfPictures*, *SameCategory*, *SameBrandTargetGroup*	647.66	0.000
8	*SameCategory*, *ProductPrice1*, *NumberOfPictures*	7.14	0.008
9	*NumberOfExactSameProductsInBasket*, *SameSubcategoryTargetGroup*, *SameBrandTargetGroup*	1363.71	0.000
10	*PaymentMethod*, *OnlyColorDiff*, *NumberOfPictures*	91.34	0.000

TABLE 22: Overview of the indicators chosen and out-of-sample predictive accuracy for each RNN

5.5 Evaluation

It has long been recognized that it is difficult to objectively measure the insight generated from visualization (North, 2006; Plaisant, 2004). Since this study is primarily on a machine learning algorithm that can be visualized rather than visualization itself, our evaluation is based on two criteria: First, the algorithm's ability to generate interpretable RNNs that can

help decision makers gain actionable insight. Second, the algorithm's ability to generate accurate out-of-sample predictions in comparison to commonly used "black box" machine learning algorithms. Both of these aspects are measured using our motivating case as an example dataset.

Using the testing set, we calculate the statistical significance of each of these RNNs when corrected for all previous RNNs. Results are displayed in Table 22.

5.5.1 Strategic shopping

Applying the algorithm in this study to the motivating case yields insight into the interactions between different variables. In particular, we are able to analyze the phenomenon of strategic shopping.

We find that simply focusing on product size is a too parochial view. Our algorithm consistently chooses more broadly defined measures of product similarity (such as *SameCategoryTargetGroup*) over narrower indicators (such as *SimilarProduct*).

In following we will present some of insight generated from applying our algorithm to the motivating case. We apply our algorithm to the training set to extract ten different RNNs. Due to space restrictions, we do not present all of the RNNs and focus on the three RNNs with the highest additional out-of-sample predictive accuracy, based on the χ^2-squared statistic which was calculated as defined in (53).

5.5.2 RNN 1: CustomerPreviousReturnRate, SameCategoryTargetGroup and NumberOfProductsInBasket

The interaction between *CustomerPreviousReturnRate*, *SameCategoryTargetGroup* and *NumberOfProductsInBasket* is visualized in Figure 14. Perhaps unsurprisingly, the strongest single predictor for product returns is *CustomerPreviousReturnRate*. Customers who have returned a high percentage of the products they purchased in the past are likely to continue doing so in the future. The relationship is close to being linear with the exception of cases where the *CustomerPreviousReturnRate* is 0. This is most likely due to the fact that customers who have not purchased anything before are assigned a *CustomerPreviousReturnRate* of 0.

The interaction between *SameCategoryTargetGroup* and *NumberOfProductsInBasket* is more complicated. When *SameCategoryTargetGroup* is 0, a higher number of *NumberOfProductsInBasket* tends to lead to higher return rate. This increase is steepest when there are few products in the basket. However, when *SameCategoryTargetGroup* is higher, the positive impact of *NumberOfProductsInBasket* on product returns all but disappears and can even become negative: By definition of the two variables, a *SameSubcategoryTargetGroup* of 4 implies that *NumberOfProductsInBasket* must at least be 5. That means if *SameCategoryTargetGroup* is 4 and *NumberOfProductsInBasket* is 5, then the customer has ordered five products in total, all of which are of the same category and target group. This is a strong indicator that the customer has ordered these products not intending to keep all of them. If, in addition, the customer has returned little in the past or is a new customer, having additional products in the basket decreases the probability of a product being returned.

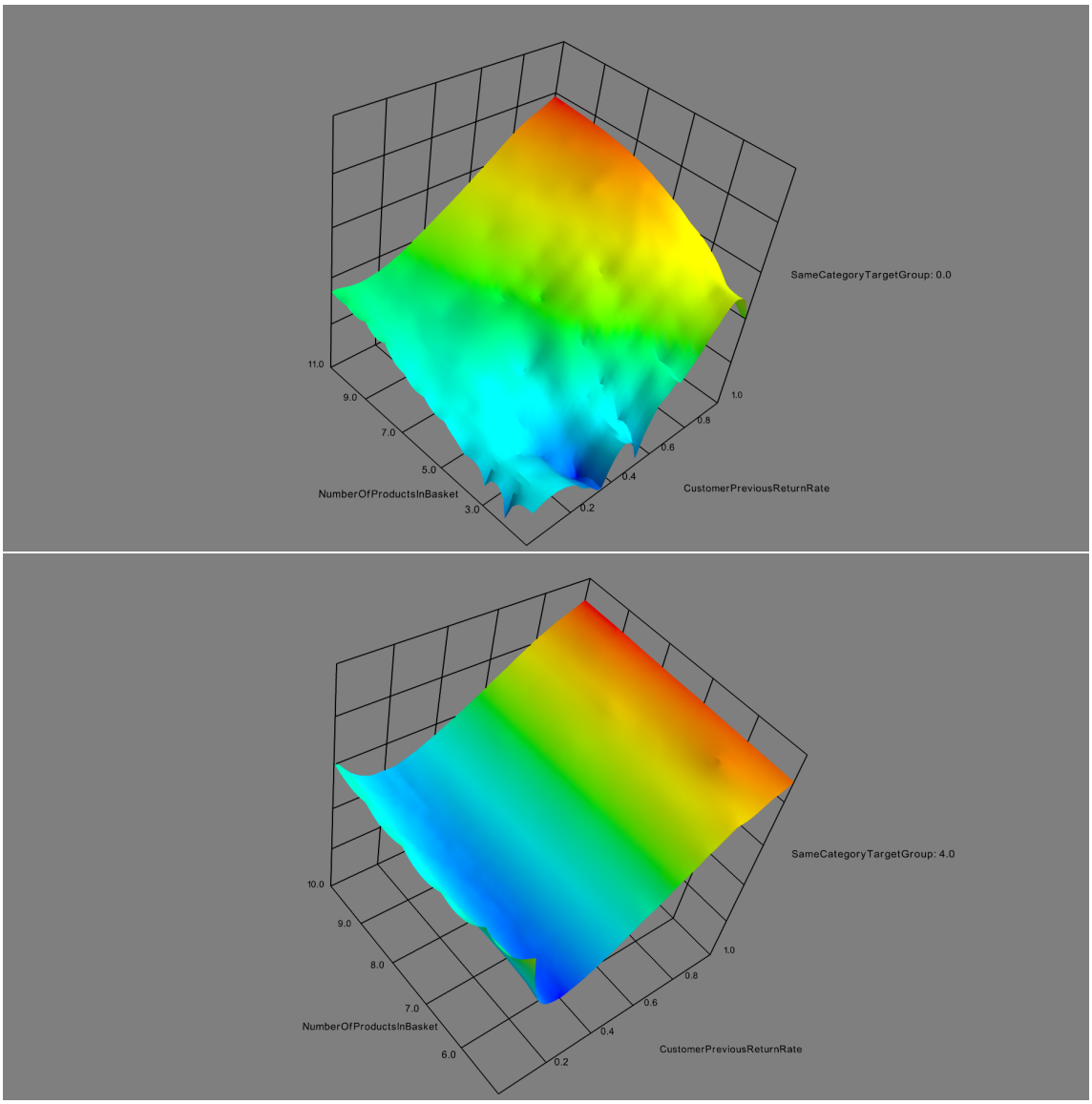


FIGURE 14: RNN 1: Probability of a product being returned depending on *CustomerPreviousReturnRate*, *SameCategoryTargetGroup* (on slider) and *NumberOfProductsInBasket*. Above: *SameCategoryTargetGroup* = 0.0. Below: *SameCategoryTargetGroup* = 4.0.

5.5.3 RNN 3: ProductPrice1, SameCategoryActivity and SameCategory

The relationship between *ProductPrice1*, *SameCategoryActivity* and *SameCategory* is visualized in Figure 15. Note that *SameCategory* is a discrete variable, which is the reason the plot seems volatile. The most obvious finding is that *ProductPrice1* has a positive impact on product returns: More expensive products are more likely to be returned than less expensive ones. This increase is steeper for cheaper products than it is for more expensive ones.

In addition, when *SameCategoryActivity* is 0, *SameCategory* acts to increase the influence of *ProductPrice1*: If there are more products of the same category, the difference between the probability of a high-priced product being returned and the the probability of a low-priced product being returned increases. It appears that when customers purchase several low-priced products that are similar to each other, it is more likely that they intend to keep all of them than when they buy several high-priced products that are similar to each other.

In the previous subsection, we have seen that a higher *SameCategoryTargetGroup* can neutralize the predictive power of *NumberOfProductsInBasket*. For *SameCategoryActivity* and *SameCategory* it is exactly the other way around: A higher *SameCategoryActivity* and *SameCategory* can increase the predictive power of *SameCategory*. When customers have several products of the same category and activity in the shopping basket, having additional products of the same category, but not the same activity, increases the likelihood of a product being returned, if the product is comparatively low-priced.

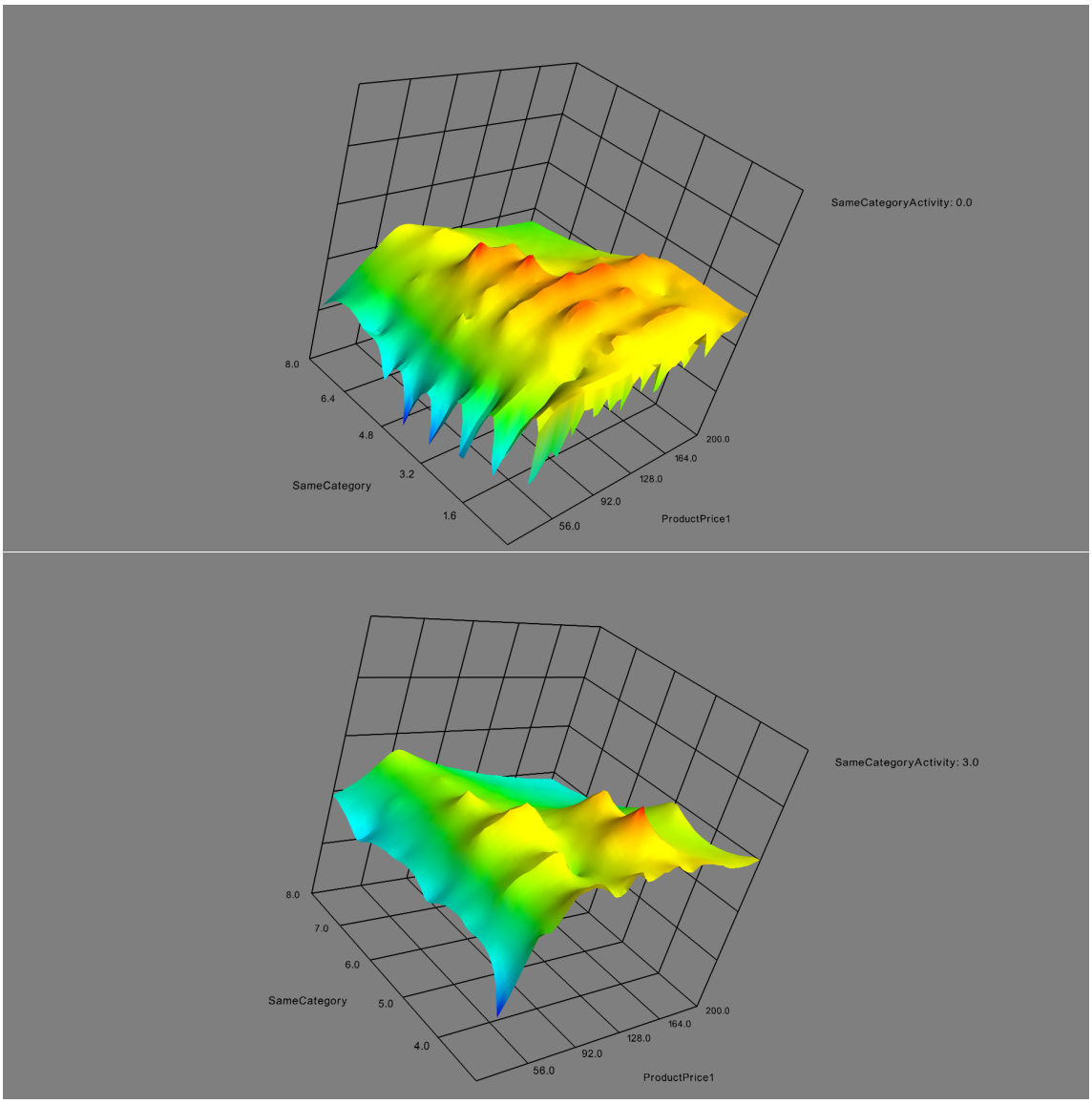


FIGURE 15: RNN 3: Probability of a product being returned depending on *ProductPrice1*, *SameCategoryActivity* (on slider) and *SameCategory*. Above: *SameCategoryActivity* = 0.0. Below: *SameCategoryActivity* = 3.0.

5.5.4 *RNN 4: SameTargetGroup, NumberOfPictures and NumberOfExactSameProductsInBasket*

The relationship between *SameTargetGroup*, *NumberOfPictures* and *NumberOfExactSameProductsInBasket* is visualized in Figure 16. Note that *NumberOfExactSameProductsInBasket* is discrete, must at least be greater or equal to 1 and hardly ever assumes a value that is

greater than 2.

Similar to the previous measures of product similarity, *SameTargetGroup* is positively correlated with product returns. This increase is also strongest when there are fewer products in the basket.

Contrary to what one might have expected, we find that a higher *NumberOfExactSameProductsInBasket* is negatively correlated with the likelihood of a product being returned. This implies that when a customer orders two products that are exactly the same, this is an indicator he or she has a clear idea of what he or she wants and is therefore less likely to return the product. Additionally, when the customer orders a product despite it being advertised with only one picture, it also implies that he or she already has a clear understanding about the product and is therefore less likely to return it.

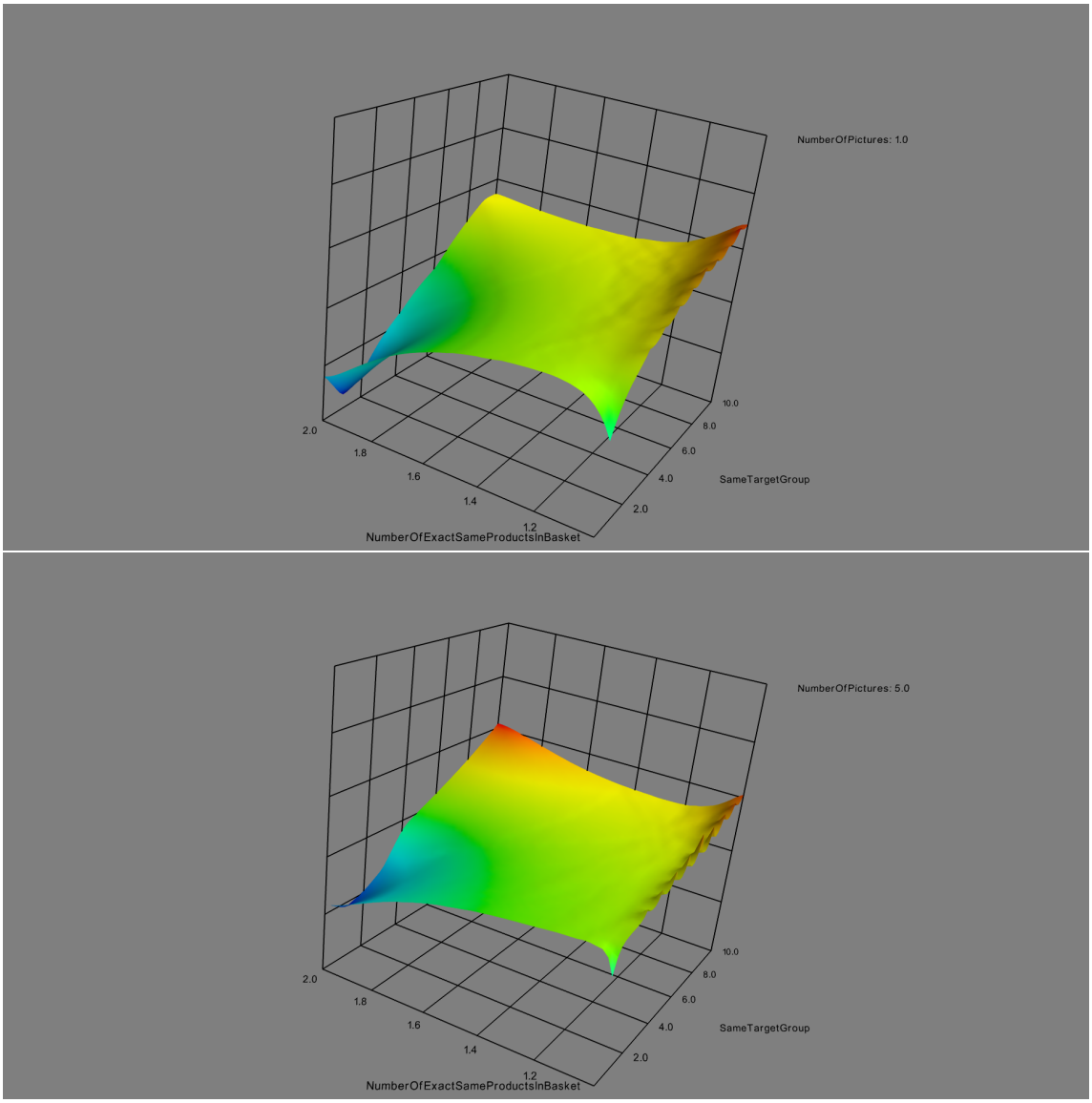


FIGURE 16: RNN 4: Probability of a product being returned depending on *SameTargetGroup*, *NumberOfPictures* (on slider) and *NumberOfExactSameProductsInBasket*. Above: *NumberOfPictures* = 1.0. Below: *NumberOfPictures* = 5.0.

5.5.5 *RNN 5: CustomerPreviousReturnRate, OnlySizeDiff and SameActivity*

The relationship between *CustomerPreviousReturnRate*, *OnlySizeDiff* and *SameActivity* is visualized in Figure 17. The correlation between *SameActivity* and the probability of a product being returned remains positive, even though this RNN is corrected for the effect of all previous RNNs.

We also find that *SameActivity* has a particularly strong impact on product returns when *CustomerPreviousReturnRate* is low. The probability of a product return is particularly low when *SameActivity* is 0 or 1. However, when *OnlySizeDiff* is 1 (or higher), this effect is neutralized. This is likely to be due to the fact that *OnlySizeDiff* is a narrower indicator of product similarity than *SameActivity*. By definition, *OnlySizeDiff* must always be smaller of equal to *SameActivity*.

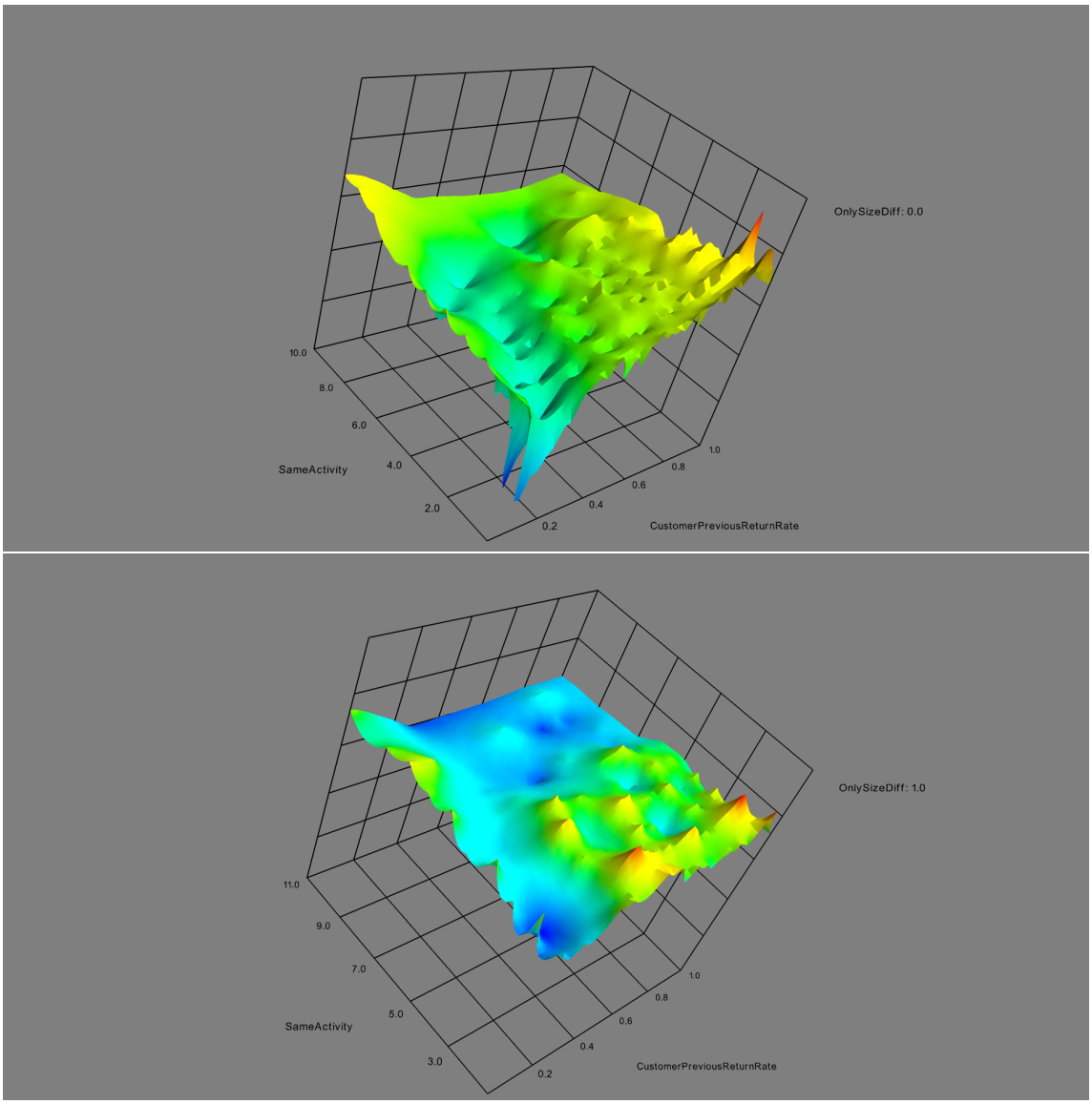


FIGURE 17: RNN 5: Probability of a product being returned depending on *CustomerPreviousReturnRate*, *OnlySizeDiff* (on slider) and *SameActivity.* Above: *OnlySizeDiff* = 0.0. Below: *OnlySizeDiff* = 1.0

5.5.6 Lessons for our motivating case

In this section, we summarize the insight for business practitioners generated by applying the algorithm to our motivating case. Our findings suggest that when trying to understand product returns, simply focusing on product size may be a too parochial view. In fact, we find that our algorithm consistently chooses more broadly defined measures of product similarity (such as *SameCategoryTargetGroup*) over narrower indicators (such as *SimilarProduct*). The phenomenon of customers ordering different similar products for the purpose of selecting one or a few of them goes beyond product size.

For instance, we have learned that broader indicators such as *NumberOfProductsInBasket, SameCategoryTargetGroup*, *SameCategory, SameActivity* or *SameCategoryActivity* are also important influencing factors for product returns. In addition, these indicators should also be seen in proportion of the total number of products in the basket. Customers who order several products of the same category and target group are looking for something very specific and are ordering strategically whereas those who order additional products of different categories or target groups are less likely to be strategic shoppers.

However, we have also learned that whenever customers do order products that differ in size only, more broadly defined indicators lose most of their impact.

In summary, when applying the decision support system developed in this study to the specific business problem of product returns in online retail, we find that strategic shopping behavior is a multi-faceted problem. Customers compare products for a whole variety of reasons. Business practitioners should take many dimensions into consideration as opposed to simply focusing on product size.

Specifically, we can derive new actionable strategies to reduce the return rate in online retail:

1. In our previous study on the topic, we have developed a "black box" prediction system that can prevent transactions before they even take place and discussed possible intervention strategies such as limiting a customer's payment options or moral suasion (Urbanke et al., 2015b). Using the insight developed in this study, we can develop more targeted intervention strategies. For instance, online retailers can specifically inform customers that their payment options where limited because the number of products of the same category and target group was too high and ask them to reduce that number to be able to regain the full choice of payment options.

2. We have seen that the industry has developed recommender systems that can help customers choose the appropriate size of an item of clothing.[6] Our results suggest that recommender systems focusing on a larger set of items (rather than just items that differ only in size), would have a greater impact on product returns.

5.5.7 Predictive accuracy

In addition to evaluating our decision support system's ability to generate actionable insight for business practitioners, we compare its predictive ability on our example dataset to standard machine learning algorithms, both interpretable and non-interpretable. We do so to measure the degree to which the information contained in the dataset is captured by the algorithm. As interpretable machine algorithms, we choose a decision tree of depth 3 and depth

[6] http://www.wsj.com/articles/SB10001424052702304773104579270260683155216, accessed 2015-10-31

5 as well as a logistic regression. The decision tree of depth 5 is chosen despite the fact that it may be difficult to interpret in practice. We do so to ensure that our results do not depend on any specific setting of hyperparameters.

Pearson's r:						
	Trans-parent?	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
AdaBoost	No	0.3906	0.3900	0.3901	0.3893	0.3898
Decision Tree, depth 3	Yes	0.3045	0.3031	0.3028	0.2997	0.3037
Decision Tree, depth 5	Difficult	0.3360	0.3347	0.3341	0.3319	0.3349
Extra Trees	No	0.3747	0.3741	0.3740	0.3720	0.3736
Gradient Boosting	No	0.3571	0.3562	0.3555	0.3535	0.3564
Logistic Regression	Yes	0.3091	0.3082	0.3086	0.3072	0.3078
Random Forest	No	0.3797	0.3792	0.3788	0.3775	0.3792
RNN	No	0.3512	0.3536	0.3502	0.3496	0.3517
Visualized RNN	**Yes**	**0.3592**	**0.3595**	**0.3585**	**0.3568**	**0.3572**
Accuracy:						
	Interpre-table?	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
AdaBoost	No	0.6670	0.6669	0.6671	0.6660	0.6670
Decision Tree, depth 3	Yes	0.6360	0.6346	0.6351	0.6334	0.6357
Decision Tree, depth 5	Difficult	0.6451	0.6443	0.6443	0.6434	0.6445
Extra Trees	No	0.6618	0.6614	0.6615	0.6610	0.6616
Gradient Boosting	No	0.6555	0.6543	0.6545	0.6536	0.6549
Logistic Regression	Yes	0.6360	0.6359	0.6357	0.6353	0.6355
Random Forest	No	0.6635	0.6631	0.6633	0.6625	0.6636
RNN	No	0.6530	0.6537	0.6525	0.6524	0.6530
Visualized RNN	**Yes**	**0.6563**	**0.6553**	**0.6554**	**0.6550**	**0.6551**

TABLE 23: Out-of-sample performance of Visual RNN-Based Learning and benchmark algorithms

We also choose a set of "black box" algorithms against which we benchmark our own approach. These algorithms have been demonstrated to be among the best performers across a wide selection of datasets (Fernández-Delgado et al., 2014). We impose no restrictions on the trees in our ensemble learners other than requiring a minimum of 200 samples on every leaf to prevent overfitting. This results in very deep trees, which can not be interpreted, but have high out-of-sample predictive accuracy. We also train an RNN on all of our inputs variables which we transform using 2000 truncated radial basis functions, as described in Section 3.2 and a standard logistic regression.

We separate our dataset in five different equally sized subsets. We use four of the subsets as our training set and the remaining one as a testing set. We repeat our experiment for five iterations, using a different subset as the testing set every time.

The ten rules trained in section 5.5.1 are combined using a logistic regression. We calculate both Pearson's r between the probabilistic predictions and the actual outcome as well as the standard accuracy measure. Pearson's r is a useful measure for the business problem and hand, because our goal is to identify cases with a very high probability of a product being returned. Results are displayed in Table 23. Our own algorithm is denoted as "Visualized

RNN" and highlighted.

We find that our own approach consistently outperforms all other interpretable machine learning algorithms and is comparable in performance with a Gradient Boosting approach or an ordinary RNN (both of which it outperforms). Compared to the best performer, the price we pay for our machine learning algorithm being interpretable is a reduction in accuracy of about one percentage point. This implies that our approach can accurately capture most of the non-linear interactions within the dataset.

5.6 Discussion and Conclusion

In this study, we have introduced a new decision support systems that allows business practitioners to analyze complex, non-linear interactions in larger datasets. We have introduced a interpretable machine learning algorithms based on an ensemble RNNs, each of which can be visualized effectively. Using the issue of product returns in online retail as a use case, we have demonstrated how the approach can be useful to practitioners.

Using the important issue of product returns in online retail as a use case, we demonstrated how the decision support system can be used to gain an understanding of the complex interactions between different factors influencing product returns. In particular, we were able to gain insight about strategic shoppers.

When comparing our approach to standard machine learning algorithms, we found that our approach outperforms other interpretable machine learning approaches. Its predictive performance is comparable to some "black box" approaches. This suggests that the visualization contains most of the information contained in the dataset giving human decision makers a relatively full picture of the non-linear interactions in the dataset. We argue that the comparatively minor drawbacks the approach has in terms of predictive accuracy are by far outweighed by the additional value it yields in terms of interpretability that help decision makers gain an *understanding* of complex, non-linear interactions in larger datasets rather than just non-interpretable predictions associated with most machine learning algorithms. The algorithm is scalable, making it applicable to very large-scale datasets.

However, there are some limitations to our approach that require further research. The most important limitation of the algorithm is that it is difficult to apply to high-dimensional datasets. Datasets related to business intelligence problems often contain nominal indicators that need to be encoded in a large number of dummy variables. It is difficult to capture this "long tail" of information using only the approach described in this study and further research is needed to develop methods for effectively gathering the information contained in a large number of sparse variables in a manner that delivers interpretable information to human decision makers.

In summary, the algorithm developed in this study can help decision makers and business practitioners gain an understanding of the complex, non-linear interactions contained in their datasets.

6 Contribution

6.1 Findings and Results

The following section will summarize the main findings with regard to the research questions raised in the introduction to this dissertation.

6.1.1 Machine Learning for Business Intelligence

> *RQ 1.1: How can predictive methods in business intelligence be statistically evaluated?*

Chapter 2 of this dissertation demonstrated that the IS literature on predictive analytics uses the concept of outperformance almost exclusively when evaluating predictive methods. It argued that additional concepts can enhance the value of such an evaluation: In some predictive analytics studies, the goal is to *assess predictability* of a problem (Shmueli and Koppius, 2011). Such studies should be supported by testing the overall predictive accuracy. When the predictive model is to be used in combination with extant models, researchers can test for *forecast encompassing*, to assess whether the model actually provides new information or said information could also be retrieved by a combination of extant methods.

This dissertation presented a unified framework that combines the concepts of a test for overall predictive accuracy, tests for outperformance and tests for forecast encompassing into a single coherent model.

Moreover, addressing the empirical findings that hypothesis tests evaluating predictive methods can be severely distorted by a violation of the assumption of a normal distribution (Harvey et al., 1998; Harvey and Newbold, 2000), this dissertation presented an approach that allows researchers to use Monte-Carlo sampling instead of assuming a normal distribution. We also demonstrated how the framework can be applied to a panel data setting as frequently found in business intelligence problems such as customer retention prediction or bankruptcy prediction.

> *RQ 1.2: How can the complex interaction between nominal and numeric variables be modelled and the resulting a high-dimensional datasets reduced with minimal information loss?*

Chapter 3 introduced a approach for dimensionality reduction that is mainly linear, but also allows for simple non-linear transformations such as a logistic function or a rectified linear function. The approach was based on the statistical framework developed in Chapter 2. The high-dimensional input data was channelled through a set of transformations. Predictive power is measured using the approach proposed in Chapter 2. The weights of the transformation are then adjusted to maximally reject said null hypothesis that these transformations do not have statistically significant predictive power for a set of output variables, in other words, to maximize the Mahalanobis distance of the transformation from their expected value under the null hypothesis.

The approach was then applied to a dataset related to product returns in online retail and benchmarked against extant methods using a selection of classification algorithms. The results demonstrated that the approach outperforms extant methods for dimensionality reduction regardless of the classifier being used.

In chapter 4, the approach was refined further. Whereas the previous approach only allows for linear or simple non-linear transformations, the approach developed in chapter 4 is integrated into a standard backpropagation framework which enables the training of complex and possibly very deep neural networks. The chapter also introduced a new data model, which allowed for modeling complex non-linear interactions between the highly sparse features of the dataset. The new approach could be demonstrated to significantly increase predictive accuracy over both the method developed in chapter 3 as well as extant dimensionality reduction techniques.

Even though it was applied to only the singular business case of product returns in online retail, the approach is sufficiently generic to be applicable to a wide variety of business problems. The concept relies on the idea that the dataset consists of nominal variables that interact with each other in a non-linear fashion. Since most real-world business intelligence datasets consist of a combination of numeric and nominal variables, this idea is abstract enough to be applicable to a wide variety of business problems.

> *RQ 1.3: How can machine learning algorithms be customized to specific business intelligence problems?*

This dissertation identified the need for customization as an important difference between advanced business intelligence and artificial intelligence. Business analytics problems tend to be more heterogeneous than standard artificial intelligence problems: Lessons drawn from one image classification problem are likely to be applicable to another image classification problem whereas lessons drawn from one BI are less likely to applicable to another BI problems. This calls for an approach to customize machine learning algorithms to specific business problems.

Chapter 4 of this dissertation presented an approach for customizing deep neural networks to a specific business problem. The architecture was based on the interpretation of the logistic function as node that is activated with the probability of the value the logistic function assumes and models interactions between variables by interpreting the multiplication of two logistic functions as an AND-gate, which is only activated if all of its input nodes are activated as well.

Based on this idea, chapter 4 introduced a customized neural network architecture for capturing interactions in very high-dimensional, sparse datasets. It also introduced a less customized neural network architecture for classification. These architectures were then evaluated individually as well as in the form of a combined neural network. A dataset related to product returns in online retail was used as an example business case.

Using ten different classification algorithms, the customized neural network architecture for dimensionality reduction was compared to dimensionality reduction techniques developed in previous chapters as well as standard approaches. The results demonstrated that both customized architectures for dimensionality reduction strictly outperform all other dimensionality reduction techniques considered, almost regardless of the classifier being used. These results were statistically highly significant.

Out of a total of 74 different models compared, the customized approach achieved the best performance. These results strongly suggest that customizing deep neural networks to a specific business problem can significantly increase predictive accuracy, despite theoretical arguments to the contrary (Hornik et al., 1989).

Chapter 4 also presented a recipe for customization that can help experts use the frame-

work to develop more effective solutions for specific business problems. The purpose of this framework was to help practitioners who wish to apply the algorithm to a different business problem, thus increasing generalizability.

> *RQ 1.4: How can machine learning algorithms be designed to be interpretable and provide researchers and practitioners with an understanding of complex, non-linear relationships in large-scale datasets?*

This dissertation has introduced two different approaches for making machine learning algorithms interpretable.

Due to the interpretation of the logistic function in a fuzzy logic framework, the approach developed in chapter 4 is interpretable. It allows for researchers to understand the impact of non-linear interactions on a more abstract as well as a very fine-grained level. This is particularly true when a weight-sharing approach is used. This allows researchers to group possible manifestation of nominal variables into sets of similar items for which interactions exist. For instance, when applying the weight-sharing approach to product categories used for predicting product returns in online retail, the algorithm identified categories that are similar to each other. If customers have previously purchased or returned products of category A, this will have an impact on the likelihood of returning products of category B. Such information can be studied using the weight-sharing approach presented in chapter 4.

However, the approach requires significant prior knowledge on the problem domain. The researcher must have a good understanding of factors influencing product returns and possible interactions to model an architecture that he or she expects to be able to capture the prior knowledge. It also requires a more detailed understanding of the underlying algorithm. The approach is therefore not useful for presentation to domain experts who might have a good understanding of the problem domain, but do not have detailed knowledge on neural networks or machine learning algorithms in general.

Whenever the researcher does not have sufficient knowledge on the problem domain and would like understand interactions at a greater level of detail, the approach presented in chapter 5 can be used. It uses randomized neural networks to visualize non-linear interactions. In order to interpret these visualizations, it is not necessary to have a detailed understanding of the underlying algorithm, in fact being able to read and interpret standard visualization techniques such as heat maps or contour plots is sufficient. This is a particularly useful feature for sharing the acquired knowledge with domain experts.

In fact, the approaches presented in Chapters 4 and 5 can be combined for maximal effectiveness. Since the neural network architectures proposed in chapter 4 allow for extracting interpretable features (see Tables 19 and 20)), these extracted features can be combined with dense features in a combined model based on the randomized neural network approach presented in Chapter 5. The resulting model can be presented to domain experts and decision makers, since they do not require a very detailed understanding of all of the extracted features to make use of the information provided. For instance, it would be sufficient for them to understand that a particular extracted feature represents interactions between different categories or interactions between different brands in order to make sense of the model being presented to them.

This combined approach is fully interpretable in the sense that even though it is expected to achieve very good predictive accuracy, it also provides domain experts who do not have a detailed understanding of machine learning issues with actionable information and insight.

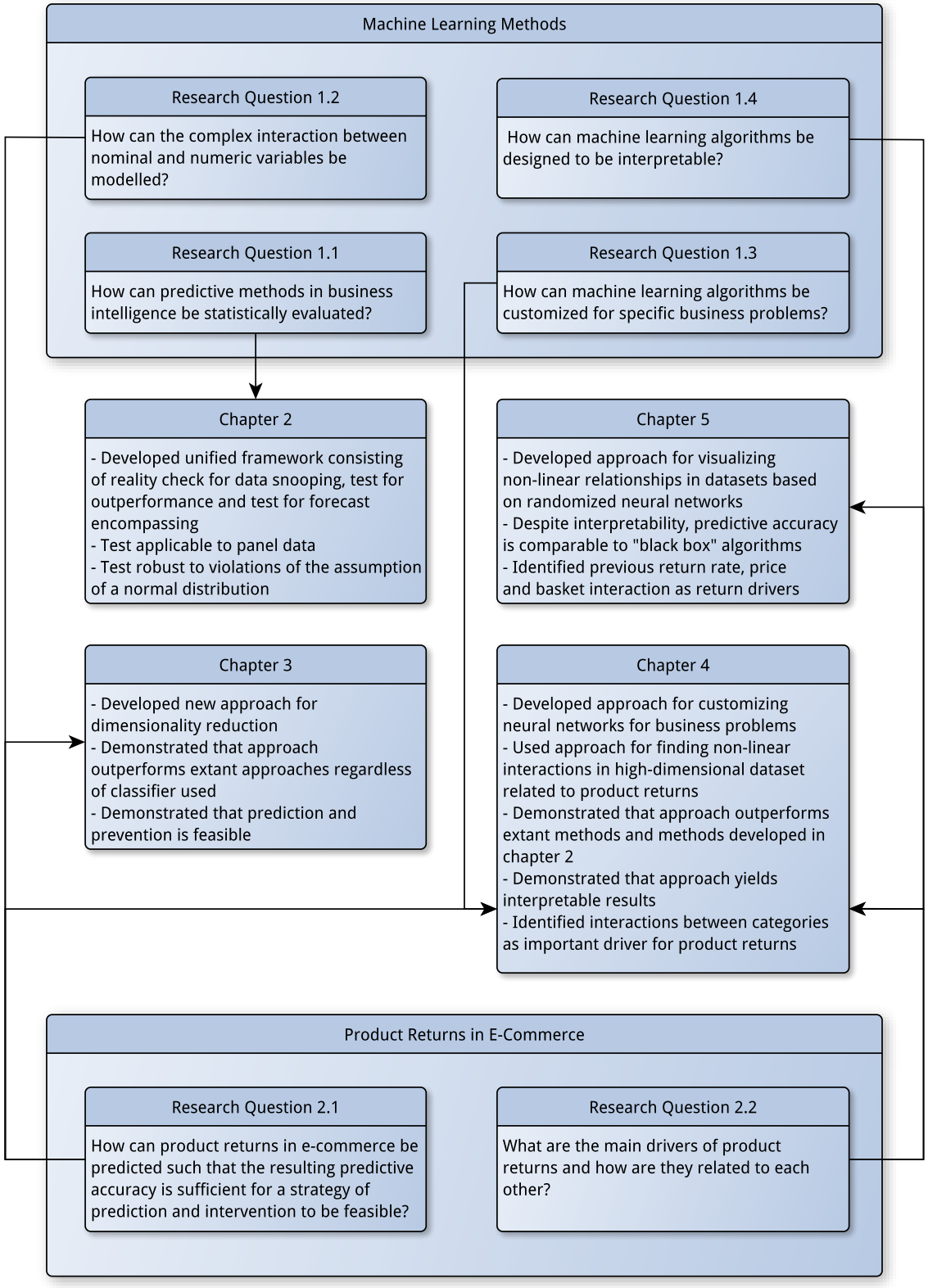


FIGURE 18: Contributions to the research questions raised in this dissertation

6.1.2 *Product Returns in E-Commerce*

RQ 2.1: How can product returns in e-commerce be predicted such that the resulting predictive accuracy is sufficient for a strategy of prediction and preven-

tion to be feasible?

Chapter 3 introduced the idea of a strategy of prediction and prevention. As a customer puts together a virtual shopping basket, the system predicts the likelihood of a product being returned before the customer has hit the order button. Whenever the likelihood of a product being returned is deemed to high, the system intervenes. One of the major research questions raised in this dissertation was whether such a strategy is feasible.

Assume that the online retailer would like to intervene whenever the probability of a product being returned is in excess of 80%, in other words we adjust the critical value at which an intervention is triggered such that the precision of the model is 80%. Using the precision-recall plots presented in both chapters 3 and 4, we can assess the recall achieved by the models in such a scenario: The model presented in chapter 3 would achieve a recall of about 40% whereas the model presented in chapter 4 achieves a recall of over 25% (note that these two numbers are based on different data models and not quite comparable).

A recall of 25% implies that the total number of returns can be reduced by up to 25% assuming a highly effective intervention strategy. Even under conservative assumptions regarding the effectiveness of an intervention strategy, the model could help achieve a substantial reduction in the number of product returns.

RQ 2.2: What are the main drivers of product returns and how are they related to each other?

Since the approaches presented in chapters 4 and 5 of this dissertation are both interpretable, we can use these findings to identify the main drivers of product returns.

In chapter 4, we found that interactions on the level of the *product category* are the best predictors for product returns. When customers place products of the same or similar categories into their basket, this increases the likelihood of each of these products being returned.

In chapter 5, we identified a number of interactions that are strong influencing factors for product returns. These interactions include the finding that the customer's return history has a positive impact on product returns, the finding that the product return probability is positively correlated to the price of a product and that the likelihood for strategic shopping increases when there are several similar products in a basket, all of which are expensive.

These results are consistent with previous findings in the marketing literature

A summary of the contributions of this dissertation structured in terms of the explicit research questions raised in the beginning is provided in Figure 18.

6.2 Implications for Theory and Practice

6.2.1 Implications for Theory

Banker and Kauffman (2004) categorize IS research into different streams. This dissertation is part of the decision support and design science stream. As noted in the introduction, predictive analytics inside this stream is mainly limited to apply existing artificial intelligence algorithms to specific research questions (Wu et al., 2008; Chen et al., 2012), with very little research being done on machine learning issues that are specific for business intelligence. One of the two major goals of this dissertation was to close this gap.

Chapter 4 identified the most important philosophical differences between traditional artifi-

cial intelligence research and advanced business analytics, namely *customization* and *interpretability*.

Whereas artificial intelligence researchers aim to develop models that are applicable to many different settings and problems, IS researchers aim to develop specific solutions for specific business problems. The differences between these approaches can be explained by different factors:

- *Heterogeneity of datasets* - BI datasets are heterogeneous than typical artificial intelligence datasets. BI datasets often consist of a combination of numeric, nominal and unstructured data. By contrast, datasets for artificial intelligence problems such as image classification, speech recognition or playing board games are more homogeneous, typically consisting of only one kind of data (such as image data or sound files).
- *Uniqueness of problems* - BI problems are also less unique. Whereas an algorithm that works well on one image classification problem could reasonably expected to work very well on another image classification problem, the same is not true for BI problems. An algorithm that performs well on a credit scoring problem may do poorly on a churn prediction problem and vice versa. BI has a greater plurality of problems that need to be addressed.
- *Comparative advantage* - IS researchers have a comparative advantage in that they integrate data management skills with business knowledge (Agarwal and Dhar, 2014). For them, making use of this comparative advantage can increase predictive accuracy.

In addition, IS researchers often require their algorithms to be interpretable, whereas this is not particularly important for artificial intelligence researchers. The reason for this difference is that artificial intelligence researchers attempt to train machines to be able to fulfill tasks that humans can already do, whereas BI researches attempt to teach machines fulfill tasks that humans cannot do. For artificial intelligence researchers, a deeper understanding of how the algorithm produces its predictions is not interesting to human decision makers. In BI, human decision makers can benefit from a deeper understanding of factors influencing certain business phenomena.

One of the major theoretical contributions of this dissertation is to provide approaches that address these philosophical differences. Researchers can make use of the algorithms developed in Chapters 4 and 5 to develop customized solutions for specific business problems and gain a deeper understanding of the underlying business phenomena.

The introduction to this dissertation also discussed important matters that have recently been debated in the IS literature regarding predictive analytics as an interpretative tool. Shmueli and Koppius (2010) and Shmueli and Koppius (2011) advocate the use of predictive analytics to evaluate theories. In the framework provided by Gregor (2006), such an approach could be useful for P-theories, that is theories that predict, but don't explain whereas more classical statistical methods are useful for E-theories, that is theories that explain, but don't predict. The approaches developed in this dissertation on the other hand are useful of EP-theories, that is theories that explain and predict. Whereas the methods and approaches used in this dissertation are clearly in the field of predictive analytics in the definition provided by Shmueli and Koppius (2011), they can be used to gain a deeper understanding of underlying phenomena and are thus useful for explanation as well.

This is particularly interesting when considering the important point raised by Lin et al. (2013) and Cohen (1992), which is that the concept of statistical significance uses becomes meaningless in the context of big datasets. They advocate that researchers should instead study the impact of certain variables. The approaches developed in chapters 4 and 5 of this dissertation are useful in this regard. They can be seen as a deviation from the inherently linear concept of statistical significance, focusing on the impact of non-linear interactions instead.

Regarding the issue of product returns in online retail, our findings can be seen as strong evidence for the impact of impulsive consumption patterns on product returns. A large number of similar products in the same virtual shopping basket is the online equivalent to in-store browsing, which previous literature had demonstrated to positively influence product returns. Our results suggest that such behaviour can measured in an e-commerce context and has significant impact on product returns.

6.2.2 Implications for Practice

Chapter 4 of this dissertation presented a general approach for customizing deep neural networks to specific business problems. The purpose of this section is to present a more general framework practitioners can use to develop customized solutions for specific business problems.

1. *Research Business Problem* - If the researchers do not already have prior knowledge on the business problem, the first step should be to review the relevant business literature on the problem. The goal should be to identify influencing factors, possible interactions between these influencing factors and business criteria for a successful predictive model.

   For instance, chapter 3 of this dissertation reviewed the relevant literature on product returns. It highlighted the difficulty of the problem, since product returns are a cost factor on the one hand, but also an essential part of an e-commerce business model. It also identified fraud and impulse shopping as major influencing factors for product returns. Chapter 4 of this dissertation had a stronger focus on the issue of impulse shopping. It argued that impulse shopping is often caused by factors such as hedonistic consumption patterns or in-store browsing that should be readily observable in an e-commerce dataset.

   These findings had two important implications: First, a predictive model for product returns should focus on observable consumption patterns related to impulse shopping. This includes the number and similarity of products in the basket as well as previous customer behavior. Constructing a model that focuses on these interactions will yield the best results. Second, since product returns are also an inherent part of online retailers' business model, the focus should be on those customers that abuse their rights. In other words, the goal is to develop a model with a reasonably low recall but as high a precision as possible.

2. *Build Data Model* - The second step is to construct the data model. Since the business intelligence dataset is likely to consist of a mixture of numeric and nominal variables, it makes sense to differentiate between the two. In order to capture useful interactions, the nominal variables should be transformed into a sparse, high-dimensional dataset which can then be reduced using the approaches proposed in Chapters 3 or 4.

For instance, chapter 4 of this dissertation built on the previously generated insight that the focus should be on identifying impulsive consumption patterns. It represented each of the manifestations of a nominal variable by counting the number of times this particular manifestation had occurred. For instance, it counted the number of times a customer had placed a product of category X into the virtual shopping basket or previously returned a product of category X.

3. *Customize Neural Network* - The third step is to customize the neural network such to the interactions of the sparse variables.

   For instance, one of the factors used in chapter 4 of this dissertation to predict the probability of a product being returned by modeling the interaction between the category of the product itself and the customer's behavior with regard to products of the same and other category. The model was constructed to identify categories that are close substitutes to each other, assuming that return behavior of similar categories are related to each other.

4. *Train Classifiers or Regressors* - The fourth step is train a selection of classifiers on the extracted and numeric features. For some classifiers (such as neural networks) it is beneficial to rescale the extracted features first.

   For instance, Chapters 3, 4 and 5 of this dissertation benchmarked a selection of standard approaches and self-developed algorithms. The approach developed in Chapter 5 can also be used to gain further insight on non-linear interactions.

5. *Evaluate* - The predictive accuracy of the different models can be statistically evaluated using the framework developed in chapter 2. In addition to outperformance, researchers can also evaluate forecast encompassing and, when necessary, check for data snooping. The predictive model should also be evaluated in terms of the business criteria formulated in the first step.

   For instance, in chapter 3 of this dissertation concluded that a good predictive model addressing the issue of product returns should have a high precision at a reasonably low level of recall. The predictive accuracy of different models was therefore evaluated in these terms. Furthermore, simple calculations were conducted to estimate the feasibility of a strategy of prediction and prevention.

6. *Interpret Results* - One of the major advantages of the algorithms developed in chapter 4 and 5 of this dissertation is that unlike ordinary machine learning algorithms, they allow for interpreting the results. The approach in chapter 4 allows the researchers to extract interpretable features from high-dimensional datasets. The approach in chapter 5 allows the researchers to visualize these features effectively. This is particularly advantageous, as it is not necessary to have deep understanding of the underlying algorithms to interpret the visualizations. They can therefore be shared with domain experts to ask for comments.

If necessary, this insight can be used to further refine the data model in step 2 or re-customize the neural networks in step 3.

A summary of this approach is depicted in Figure 19.

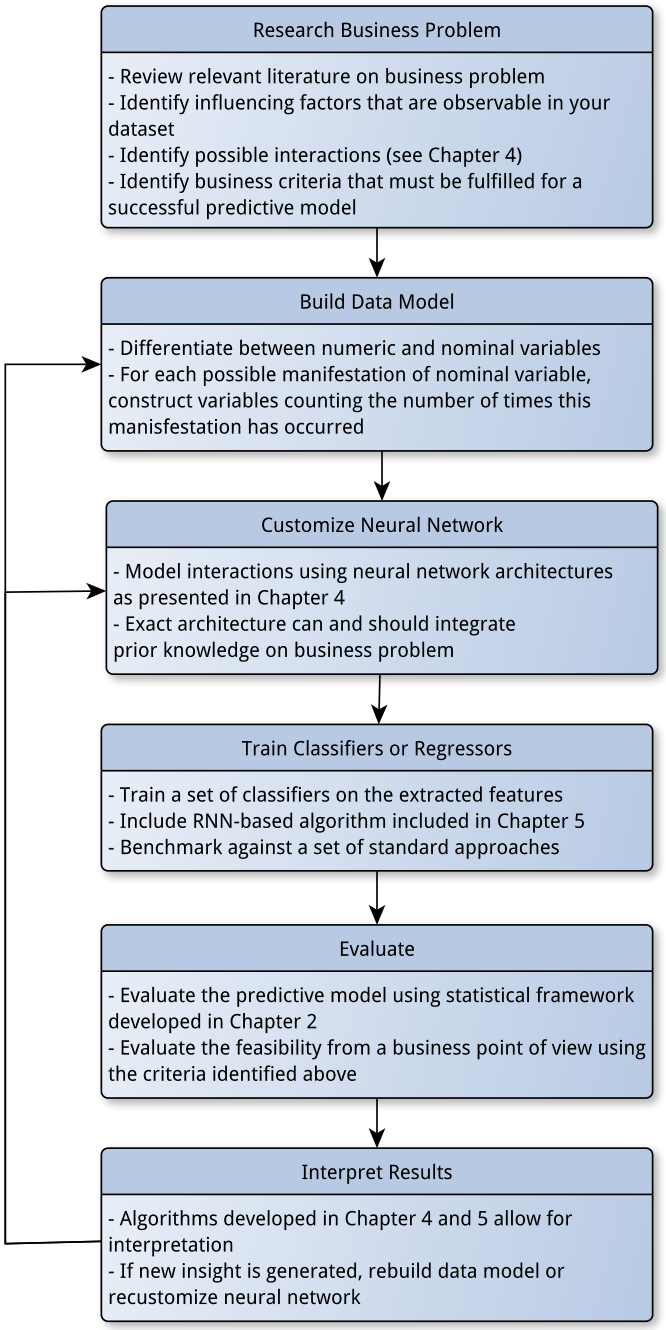


FIGURE 19: Lifecycle of a machine learning experiment using approaches developed in this study

For online retailers, this dissertation provides important guidelines to address the issue of product returns in online retail. Whereas existing strategies on product returns were one-size-fits-all and failed to differentiate between wanted and unwanted returns, the strategy of prediction and prevention proposed in this dissertation is more targeted and closely aligned with online retailers' business model. Chapter 3 of this dissertation has had a particular focus on the economic feasibility of such an approach. The findings support the view that such a strategy has the potential for significantly reducing the product return rate.

6.3 Limitations and Further Research

6.3.1 Limitations

The findings in chapters 3,4 and 5 are based on a single business dataset. This is due to the difficulty of obtaining appropriate datasets to sufficiently address this business problem. In fact, this dissertation differs from previous research on product returns in that can be based on such rich data. In addition, given the interdisciplinary nature of IS, it is far from being unusual to develop algorithms for single business cases and the dataset related to this business case is typical for many business intelligence problems. Therefore, the results of this study can reasonably be expected to be applicable to other business cases as well.

The most important limitation for the practical relevance of the return strategies proposed in this dissertation is that the effectiveness of the intervention strategies have not been empirically validated. Even though previous research suggests that these strategies should be successful, it is hard to quantify their impact. This dissertation demonstrated that predictive models are sufficiently accurate for a strategy of prediction and prevention to be feasible in principle and provides rough estimates for its potential, however the specific impact needs further study.

6.3.2 Further Research

So far, the algorithm for dimensionality reduction presented in chapters 3 and 4 has only been used in a supervised way. In particular, the approach for customizing neural networks to specific business problems can be seen as a non-linear version of structural equation modeling or partial least squares regression. However, both structural equation modeling and partial least squares regression contain the concept of latent variables, which neural networks do not have (unlike Bayesian networks). Further research could integrate the mathematical framework developed in these chapters the the concept of partial least squares regression to help researchers and practitioners gain a closer understanding of customer behavior and other business phenomena.

Chapter 4 has also introduced the idea of interpreting logistic functions in a fuzzy logic sense and interpreting AND-gates as a multiplication of two logistic functions. Other gates, such as OR, XOR or XNOR can also be expressed in a similar fashion. These concepts could enhance the idea of customizing neural networks to specific business problem in that they give researchers greater flexibility in modeling interactions and relationships between variables. Such a study could also formalize the concept of fuzzy logic and interactions with greater mathematical rigor.

The approach for visualization presented in Chapter 5 can be enhanced by a kernel density estimator. Chapter 5 introduced the idea of truncated radial basis functions which are more efficient in terms of memory usage. However, due the fact that they are truncated, kernel density estimation is non-trivial. However, adding a kernel density estimator can support decision makers, in that it enables them to assess the frequency at which certain phenomena occur. This could enhance the usefulness of the framework presented in Chapter 5.

As mentioned above, the intervention strategies proposed in chapter 2, which are an essential ingredient of the proposed strategy of prediction and prevention of product returns, require further study. Such studies are best undertaken in the form of field experiments. The researcher can build on the prediction strategies developed in this dissertation to give a set of

customers are certain intervention whereas a control group is not given that treatment. The objective of these experiments should be to study the impact of the treatment on the rate of product returns as well as the likelihood of customers to make repeat purchases.

6.4 Conclusion

The goal of this dissertation was to introduce machine learning methods that are specifically developed for the purposes of business intelligence, particularly with regard to the issues of customization and interpretability. These approaches where then applied to the important business problem of product returns in online retail. The duality of this research approach was consistently reflected in the research design of this dissertation.

This dissertation developed a number of customizable and interpretable machine learning approaches and presented a coherent framework for applying these approaches to business intelligence problems. The dissertation also proposed a strategy for predicting and preventing product returns in online retail, provided a suitable prediction model and demonstrated the feasibility of such an approach.

References

Abbasi, A., Albrecht, C., Vance, A., and Hansen, J. 2012. "Metafraud: a meta-learning framework for detecting financial fraud," *Mis Quarterly* (36:4), pp. 1293–1327.

Abbasi, A., Zhang, Z., Zimbra, D., Chen, H., and Nunamaker, J., Jay F. 2010. "Detecting Fake Websites: The Contribution of Statistical Learning Theory," *MIS Quarterly* (34:3), pp. 435–461.

Acharya, N., Shrivastava, N. A., Panigrahi, B., and Mohanty, U. 2014. "Development of an artificial neural network based multi-model ensemble to estimate the northeast monsoon rainfall over south peninsular India: an application of extreme learning machine," *Climate Dynamics* (43:5-6), pp. 1303–1310.

Adelaar, T., Chang, S., Lancendorfer, K. M., Lee, B., and Morimoto, M. 2003. "Effects of media formats on emotions and impulse buying intent," *Journal of Information Technology* (18:4), pp. 247–266.

Adomavicius, G., Bockstedt, J., and Gupta, A. 2012. "Modeling Supply-Side Dynamics of IT Components, Products, and Infrastructure: An Empirical Analysis Using Vector Autoregression," *Information Systems Research* (23:2), pp. 397–417.

Agarwal, A., Chapelle, O., Dudík, M., and Langford, J. 2014. "A reliable effective terascale linear learning system," *The Journal of Machine Learning Research* (15:1), pp. 1111–1133.

Agarwal, R., and Dhar, V. 2014. "Editorial—Big Data, Data Science, and Analytics: The Opportunity and Challenge for IS Research," *Information Systems Research* (25:3), pp. 443–448.

Aguilar, F. X., and Cai, Z. 2010. "Conjoint effect of environmental labeling, disclosure of forest of origin and price on consumer preferences for wood products in the US and UK," *Ecological Economics* (70:2), pp. 308–316.

Aguilar-Ruiz, J. S., Giráldez, R., and Riquelme, J. C. 2007. "Natural encoding for evolutionary supervised learning," *IEEE Transactions on Evolutionary Computation* (11:4), pp. 466–479.

Alfaro, E., García, N., Gámez, M., and Elizondo, D. 2008. "Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks," *Decision Support Systems* (45:1), pp. 110–122.

Alsharif, O., Ouyang, T., Beaufays, F., Zhai, S., Breuel, T., and Schalkwyk, J. 2015. "Long short term memory neural network for keyboard gesture decoding," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 2076–2080.

Ancarani, F., Gerstner, E., Posselt, T., and Radic, D. 2009. "Could higher fees lead to lower prices?" *Journal of Product & Brand Management* (18:4), pp. 297–305.

Anderson, E. T., Hansen, K., and Simester, D. 2009. "The option value of returns: Theory and empirical evidence," *Marketing Science* (28:3), pp. 405–423.

Anglano, C., and Botta, M. 2002. "NOW G-Net: learning classification programs on networks of workstations," *Evolutionary Computation, IEEE Transactions on* (6:5), pp. 463–480.

Arnott, D., and O'Donnell, P. 2008. "A note on an experimental study of DSS and forecasting exponential growth," *Decision Support Systems* (45:1), pp. 180–186.

Asdecker, B. 2015. "Statistiken Retouren Deutschland - Definition," `http://www.retourenforschung.de/`, accessed: 2015-03-16.

Autry, C. W. 2005. "Formalization of reverse logistics programs: A strategy for managing liberalized returns," *Industrial Marketing Management* (34:7), pp. 749–757.

Autry, C. W., Hill, D. J., and O'Brien, M. 2007. "Attitude toward the customer: a study of product returns episodes," *Journal of Managerial Issues* pp. 315–339.

Bacardit, J., and Garrell, J. M. 2007. "Bloat control and generalization pressure using the minimum description length principle for a pittsburgh approach learning classifier system," in *Learning Classifier Systems*, , Springer, pp. 59–79.

Bai, X. 2011. "Predicting consumer sentiments from online text," *Decision Support Systems* (50:4), pp. 732–742.

Bandyopadhyay, S., and Paul, A. A. 2010. "Equilibrium returns policies in the presence of supplier competition," *Marketing Science* (29:5), pp. 846–857.

Bang, Y. H., Yoo, C. K., and Lee, I.-B. 2002. "Nonlinear PLS modeling with fuzzy inference system," *Chemometrics and intelligent laboratory systems* (64:2), pp. 137–155.

Banker, R. D., and Kauffman, R. J. 2004. "50th anniversary article: the evolution of research on information systems: a fiftieth-year survey of the literature in management science," *Management Science* (50:3), pp. 281–298.

Bao, H., Li, Q., Liao, S. S., Song, S., and Gao, H. 2013. "A new temporal and social PMF-based method to predict users' interests in micro-blogging," *Decision Support Systems* (55:3), pp. 698–709.

Bardhan, I., Oh, J.-h., Zheng, Z., and Kirksey, K. 2014. "Predictive Analytics for Readmission of Patients with Congestive Heart Failure," *Information Systems Research* (26:1), pp. 19–39.

Beatty, S. E., and Ferrell, M. E. 1998. "Impulse buying: modeling its precursors," *Journal of retailing* (74:2), pp. 169–191.

Beazley, D. M. 1996. "SWIG: An easy to use tool for integrating scripting languages with C and C++." *Proceedings of the 4th Annual Tcl/Tk Workshop, Monterey, CA July 6-10, 1996* pp. 1–12.

Bechwati, N. N., and Siegal, W. S. 2005. "The Impact of the Prechoice Process on Product Returns," *Journal of Marketing Research* (42:3), pp. 358–367.

Bengio, Y., LeCun, Y., et al. 2007. "Scaling learning algorithms towards AI," *Large-scale Kernel Machines* (34:5).

Benítez, J. M., Castro, J. L., and Requena, I. 1997. "Are artificial neural networks black boxes?" *Neural Networks, IEEE Transactions on* (8:5), pp. 1156–1164.

Bernadó-Mansilla, E., and Garrell-Guiu, J. M. 2003. "Accuracy-based learning classifier systems: models, analysis and applications to classification tasks," *Evolutionary computation* (11:3), pp. 209–238.

Bhattacharyya, S., Jha, S., Tharakunnel, K., and Westland, J. C. 2011. "Data mining for credit card fraud: A comparative study," *Decision Support Systems* (50:3), pp. 602–613.

Bhimani, A., and Willcocks, L. 2014. "Digitisation, Big Data and the transformation of accounting information," *Accounting and Business Research* (44:4), pp. 469–490.

Bishop, C. M. 1995. *Neural Networks for Pattern Recognition*, Oxford University Press.

Bjørner, T. B., Hansen, L. G., and Russell, C. S. 2004a. "Environmental labeling and consumers' choice—an empirical analysis of the effect of the Nordic Swan," *Journal of Environmental Economics and Management* (47:3), pp. 411–434.

Bjørner, T. B., Hansen, L. G. a., and Russell, C. S. 2004b. "Environmental labeling and consumers' choice—an empirical analysis of the effect of the Nordic Swan," *Journal of Environmental Economics and Management* (47:3), pp. 411–434.

Bonczek, R. H., Holsapple, C. W., and Whinston, A. B. 2014. *Foundations of Decision Support Systems*, Academic Press.

Bonifield, C., Cole, C., and Schultz, R. L. 2010. "Product returns on the Internet: A case of mixed signals?" *Journal of Business Research* (63:9), pp. 1058–1065.

Bower, A. B., and Maxham III, J. G. 2012. "Return shipping policies of online retailers: Normative assumptions and the long-term consequences of fee and free returns," *Journal of Marketing* (76:5), pp. 110–124.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. 1984. "Classification and regression trees," *Wadsworth, Belmont, CA* .

Brookshire, D. S., Thayer, M. A., Schulze, W. D., and d'Arge, R. C. 1982. "Valuing public goods: a comparison of survey and hedonic approaches," *The American Economic Review* (72:1), pp. 165–177.

Broomhead, D. S., and Lowe, D. 1988. "Radial basis functions, multi-variable functional interpolation and adaptive networks," Tech. rep., DTIC Document.

Browne, M., Rizet, C., Leonardi, J., and Allen, J. 2008. "Analysing energy use in supply chains: the case of fruits and vegetables and furniture," *Proceedings of the Logistics Research Network Conference* pp. 1–6.

Bryk, A. S., and Raudenbush, S. W. 1992. *Hierarchical Linear Models: Applications and Data Analysis Methods*, Sage Publications, Inc.

Buckler, F., and Hennig-Thurau, T. 2008. "Identifying hidden structures in marketing's structural models through universal structure modeling: An Explorative Bayesian Neural Network Complement to LISREL and PLS. Marketing-Journal of Research and Management," *Marketing Journal of Research and Management* (2), pp. 49–68.

Cao, Q., Ewing, B. T., and Thompson, M. A. 2012. "Forecasting medical cost inflation rates: A model comparison approach," *Decision Support Systems* (53:1), pp. 154–160.

Cao, Q., and Parry, M. E. 2009. "Neural network earnings per share forecasting models: A comparison of backward propagation and the genetic algorithm," *Decision Support Systems* (47:1), pp. 32–41.

Carbonneau, R. A., Kersten, G. E., and Vahidov, R. M. 2011. "Pairwise issue modeling for negotiation counteroffer prediction using neural networks," *Decision Support Systems* (50:2), pp. 449–459.

Cassill, N. L. 1998. "Do customer returns enhance product and shopping experience satisfaction?" *The International Review of Retail, Distribution and Consumer Research* (8:1), pp. 1–13.

Caulkins, J. P., Ding, W., Duncan, G., Krishnan, R., and Nyberg, E. 2006. "A method for managing access to web pages: Filtering by Statistical Classification (FSC) applied to text," *Decision Support Systems* (42:1), pp. 144–161.

Chan, S. W. K., and Franklin, J. 2011. "A text-based decision support system for financial sequence prediction," *Decision Support Systems* (52:1), pp. 189–198.

Chang, P.-C., Lai, C.-Y., and Lai, K. R. 2006a. "A hybrid system by evolving case-based reasoning with genetic algorithm in wholesaler's returning book forecasting," *Decision Support Systems* (42:3), pp. 1715–1729.

Chang, P.-C., Liu, C.-H., and Wang, Y.-W. 2006b. "A hybrid model by clustering and evolving fuzzy rules for sales decision supports in printed circuit board industry," *Decision Support Systems* (42:3), pp. 1254–1269.

Che, Y.-K. 1996. "Customer return policies for experience goods," *The Journal of Industrial Economics* pp. 17–24.

Chen, H., Chiang, R. H., and Storey, V. C. 2012. "Business Intelligence and Analytics: From Big Data to Big Impact." *MIS Quarterly* (36:4), pp. 1165–1188.

Chen, J., and Bell, P. C. 2009. "The impact of customer returns on pricing and order decisions," *European Journal of Operational Research* (195:1), pp. 280–295.

Chen, S., Cowan, C. F., and Grant, P. M. 1991. "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks* (2:2), pp. 302–309.

Chi, H.-M., Moskowitz, H., Ersoy, O. K., Altinkemer, K., Gavin, P. F., Huff, B. E., and Olsen, B. A. 2009. "Machine learning and genetic algorithms in pharmaceutical development and manufacturing processes," *Decision Support Systems* (48:1), pp. 69–80.

Chiang, W.-y. K., Zhang, D., and Zhou, L. 2006. "Predicting and explaining patronage behavior toward web and traditional stores using neural networks: a comparative analysis with logistic regression," *Decision Support Systems* (41:2), pp. 514–531.

Choi, T.-M., Hui, C.-L., Liu, N., Ng, S.-F., and Yu, Y. 2013. "Fast fashion sales forecasting with limited data and time," *Decision Support Systems* pp. 84–92.

Choi, T.-M., Hui, C.-L., Liu, N., Ng, S.-F., and Yu, Y. 2014. "Fast fashion sales forecasting with limited data and time," *Decision Support Systems* (59), pp. 84–92.

Choi, T.-M., Yu, Y., and Au, K.-F. 2011. "A hybrid SARIMA wavelet transform method for sales forecasting," *Decision Support Systems* (51:1), pp. 130–140.

Ciresan, D., Meier, U., and Schmidhuber, J. 2012. "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, , IEEE.

Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L., and Schmidhuber, J. 2011. "Flexible, high performance convolutional neural networks for image classification," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, , vol. 22.

Clark, T. E., and McCracken, M. W. 2001. "Tests of equal forecast accuracy and encompassing for nested models," *Journal of Econometrics* (105:1), pp. 85–110.

Clements, M. P., and Harvey, D. I. 2010. "Forecast encompassing tests and probability forecasts," *Journal of Applied Econometrics* (25:6), pp. 1028–1062.

Clottey, T., Benton, W. C., and Srivastava, R. 2012. "Forecasting product returns for remanufacturing operations," *Decision Sciences* (43:4), pp. 589–614.

Cohen, J. 1992. "A power primer." *Psychological Bulletin* (112:1), p. 155.

Corcoran, A. L., and Sen, S. 1994. "Using real-valued genetic algorithms to evolve rule sets for classification," *Proceedings of the First IEEE Conference on Evolutionary Computation, 1994* pp. 120–124.

Corradi, V., Swanson, N. R., and Olivetti, C. 2001. "Predictive ability with cointegrated variables," *Journal of Econometrics* (104:2), pp. 315–358.

Coussement, K., and Van den Poel, D. 2008. "Improving customer complaint management by automatic email classification using linguistic style features as predictors," *Decision Support Systems* (44), pp. 870–882.

Cui, G., Wong, M. L., and Wan, X. 2012. "Cost-Sensitive Learning via Priority Sampling to Improve the Return on Marketing and CRM Investment," *Journal of Management Information Systems* (29:1), pp. 341–374.

Dahl, G. E., Yu, D., Deng, L., and Acero, A. 2012. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on* (20:1), pp. 30–42.

Dalcin, L., Paz, R., Storti, M., and DElia, J. 2008. "MPI for Python: Performance improvements and MPI-2 extensions," *Journal of Parallel and Distributed Computing* (68:5), pp. 655–662.

David, M., Perkovič, M., Suban, V., and Gollasch, S. 2012. "A generic ballast water discharge assessment model as a decision supporting tool in ballast water management," *Decision Support Systems* (53:1), pp. 175–185.

De, P., Hu, Y., and Rahman, M. S. 2012. "An Empirical Investigation of the Effects of Product-Oriented Web Technologies on Product Returns," *Working Paper* pp. 1–34.

Delen, D. 2010. "A comparative analysis of machine learning techniques for student retention management," *Decision Support Systems* (49:4), pp. 498–506.

Delen, D., Zaim, H., Kuzey, C., and Zaim, S. 2013. "A comparative analysis of machine learning systems for measuring the impact of knowledge management practices," *Decision Support Systems* (54:2), pp. 1150–1160.

Dhar, R., and Wertenbroch, K. 2000. "Consumer choice between hedonic and utilitarian goods," *Journal of marketing research* (37:1), pp. 60–71.

Diebold, F. X., and Mariano, R. S. 1995. "Comparing predictive accuracy," *Journal of Business & economic statistics* (20:1), pp. 134–144.

Dissanayake, D., and Singh, M. 2008. "Managing returns in e-business," *Journal of Internet Commerce* (6:2), pp. 35–49.

Dittmar, H., Beattie, J., and Friese, S. 1995. "Gender identity and material symbols: Objects and decision considerations in impulse purchases," *Journal of Economic Psychology* (16:3), pp. 491–511.

Dreiseitl, S., and Ohno-Machado, L. 2002. "Logistic regression and artificial neural network classification models: a methodology review," *Journal of Biomedical Informatics* (35:5), pp. 352–359.

D'Souza, C., Taghian, M., and Lamb, P. 2006. "An empirical study on the influence of environmental labels on consumers," *Corporate Communications: An International Journal* (11:2), pp. 162–173.

Du Jardin, P., and Séverin, E. 2011. "Predicting corporate bankruptcy using a self-organizing map: An empirical study to improve the forecasting horizon of a financial failure model," *Decision Support Systems* (51:3), pp. 701–711.

Elmolla, E. S., Chaudhuri, M., and Eltoukhy, M. M. 2010. "The use of artificial neural network (ANN) for modeling of COD removal from antibiotic aqueous solution by the Fenton process," *Journal of Hazardous Materials* (179:1), pp. 127–134.

Fan, B., and Zhang, P. 2009. "Spatially enabled customer segmentation using a data classification method with uncertain predicates," *Decision Support Systems* (47:4), pp. 343–353.

Fang, X., Hu, P. J., Li, Z., and Tsai, W. 2013. "Predicting Adoption Probabilities in Social Networks," *Information Systems Research* (24:1), pp. 128–145.

Fernández, S., Graves, A., and Schmidhuber, J. 2007. "An application of recurrent neural networks to discriminative keyword spotting," *Proceedings of the International Conference on Artificial Neural Networks, 2007* pp. 220–229.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. 2014. "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?" *Journal of Machine Learning Research* (15), pp. 3133–3181.

Frank, I. E. 1990. "A nonlinear PLS model," *Chemometrics and intelligent laboratory systems* (8:2), pp. 109–119.

Fuchs, C. 2008. "The implications of new information and communication technologies for sustainability," *Environment, Development and Sustainability* (10:3), pp. 291–309.

Gabriel, E., Fagg, G. E., Bosilca, G., Angskun, T., Dongarra, J. J., Squyres, J. M., Sahay, V., Kambadur, P., Barrett, B., Lumsdaine, A., et al. 2004. "Open MPI: Goals, concept, and design of a next generation MPI implementation," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, , Springer, pp. 97–104.

Garbe, J.-N., and Richter, N. F. 2009. "Causal analysis of the internationalization and performance relationship based on neural networks – advocating the transnational structure," *Journal of international management* (15:4), pp. 413–431.

Garson, D. G. 1991. "Interpreting neural network connection weights," *AI Expert* (6:7), pp. 47–51.

Gerber, M. S. 2014. "Predicting crime using Twitter and kernel density estimation," *Decision Support Systems* pp. 115–125.

Granger, C. W., and Newbold, P. 1973. "Some comments on the evaluation of economic forecasts," *Applied Economics* (5:1), pp. 35–47.

Graves, A., Mohamed, A.-r., and Hinton, G. 2013. "Speech recognition with deep recurrent neural networks," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 6645–6649.

Gregor, S. 2006. "The nature of theory in information systems," *MIS Quarterly* pp. 611–642.

Grewal, D., Iyer, G. R., and Levy, M. 2004. "Internet retailing: enablers, limiters and market consequences," *Journal of Business Research* (57:7), pp. 703–713.

Hagenau, M., Liebmann, M., and Neumann, D. 2013a. "Automated news reading: Stock price prediction based on financial news using context-capturing features," *Decision Support Systems* (55:3), pp. 685–697.

Hagenau, M., Liebmann, M., and Neumann, D. 2013b. "Automated news reading: Stock price prediction based on financial news using context-capturing features," *Decision Support Systems* (55:3), pp. 685–697.

Halldórsson, Á., Kovács, G., Edwards, J. B., McKinnon, A. C., and Cullinane, S. L. 2010. "Comparative analysis of the carbon footprints of conventional and online retailing: A "last mile" perspective," *International Journal of Physical Distribution & Logistics Management* (40:1/2), pp. 103–123.

Harris, L. C. 2010. "Fraudulent consumer returns: exploiting retailers' return policies," *European Journal of Marketing* (44:6), pp. 730–747.

Harvey, D., Leybourne, S., and Newbold, P. 1997. "Testing the equality of prediction mean squared errors," *International Journal of forecasting* (13:2), pp. 281–291.

Harvey, D., and Newbold, P. 2000. "Tests for multiple forecast encompassing," *Journal of Applied Econometrics* (15:5), pp. 471–482.

Harvey, D. I., and Newbold, P. 2003. "The non-normality of some macroeconomic forecast errors," *International Journal of Forecasting* (19:4), pp. 635–653.

Harvey, D. S., Leybourne, S. J., and Newbold, P. 1998. "Tests for forecast encompassing," *Journal of Business & Economic Statistics* (16:2), pp. 254–259.

Hausman, A. 2000. "A multi-method investigation of consumer motivations in impulse buying behavior," *Journal of Consumer Marketing* (17:5), pp. 403–426.

Hecht-Nielsen, R. 1989. "Theory of the backpropagation neural network," *Proceedings of the International Joint Conference on Neural Networks, 1989* pp. 593–605.

Hedgebeth, D. 2007. "Data-driven decision making for the enterprise: An overview of business intelligence applications," *Vine* (37:4), pp. 414–420.

Heiman, A., McWilliams, B., Zhao, J., and Zilberman, D. 2002. "Valuation and management of money-back guarantee options," *Journal of retailing* (78:3), pp. 193–205.

Heiman, A., McWilliams, B., and Zilberman, D. 2001. "Demonstrations and money-back guarantees: market mechanisms to reduce uncertainty," *Journal of Business Research* (54:1), pp. 71–84.

Hinton, G. E., Osindero, S., and Teh, Y.-W. 2006. "A fast learning algorithm for deep belief nets," *Neural Computation* (18:7), pp. 1527–1554.

Hochreiter, S., and Schmidhuber, J. 1997. "Long short-term memory," *Neural computation* (9:8), pp. 1735–1780.

Hornik, K., Stinchcombe, M., and White, H. 1989. "Multilayer feedforward networks are universal approximators," *Neural Networks* (2:5), pp. 359–366.

Hu, P. J.-H., Wei, C.-P., Cheng, T.-H., and Chen, J.-X. 2007. "Predicting adequacy of vancomycin regimens: A learning-based classification approach to improving clinical decision making," *Decision Support Systems* (43:4), pp. 1226–1241.

Huang, S. H., and Xing, H. 2002. "Extract intelligible and concise fuzzy rules from neural networks," *Fuzzy Sets and Systems* (132:2), pp. 233–243.

Hunter, J. D. 2007. "Matplotlib: A 2D graphics environment," *Computing In Science & Engineering* (9:3), pp. 90–95.

Ince, H., and Trafalis, T. B. 2006. "A hybrid model for exchange rate prediction," *Decision Support Systems* (42:2), pp. 1054–1062.

Jang, J.-S. R., and Sun, C.-T. 1993. "Functional equivalence between radial basis function networks and fuzzy inference systems," *Neural Networks, IEEE Transactions on* (4:1), pp. 156–159.

Jiao, L., Liu, J., and Zhong, W. 2006. "An organizational coevolutionary algorithm for classification," *Evolutionary Computation, IEEE Transactions on* (10:1), pp. 67–80.

Joo Park, E., Young Kim, E., and Cardona Forney, J. 2006. "A structural model of fashion-oriented impulse buying behavior," *Journal of Fashion Marketing and Management: An International Journal* (10:4), pp. 433–446.

Kao, L.-J., Chiu, C.-C., Lu, C.-J., and Chang, C.-H. 2013. "A hybrid approach by integrating wavelet-based feature extraction with MARS and SVR for stock index forecasting," *Decision Support Systems* (54:3), pp. 1228–1244.

Karbowski, A., Malinowski, K., and Niewiadomska-Szynkiewicz, E. 2005. "A hybrid analytic/rule-based approach to reservoir system management during flood," *Decision Support Systems* (38:4), pp. 599–610.

Kass, G. V. 1980. "An exploratory technique for investigating large quantities of categorical data," *Applied statistics* pp. 119–127.

Keim, D., et al. 2002. "Information visualization and visual data mining," *IEEE Transactions on Visualization and Computer Graphics* (8:1), pp. 1–8.

Ketter, W., Collins, J., Gini, M., Gupta, A., and Schrater, P. 2009. "Detecting and forecasting economic regimes in multi-agent automated exchanges," *Decision Support Systems* (47:4), pp. 307–318.

Ketter, W., Collins, J., Gini, M., Gupta, A., and Schrater, P. 2012. "Real-time tactical and strategic sales management for intelligent agents guided by economic regimes," *Information Systems Research* (23:4), pp. 1263–1283.

Ketzenberg, M. E., and Zuidwijk, R. A. 2009. "Optimal pricing, ordering, and return policies for consumer goods," *Production and Operations Management* (18:3), pp. 344–360.

Khansa, L., and Liginlal, D. 2011. "Predicting stock market returns from malicious attacks: A comparative analysis of vector autoregression and time-delayed neural networks," *Decision Support Systems* (51:4), pp. 745–759.

Kim, H.-N., El-Saddik, A., and Jo, G.-S. 2011. "Collaborative error-reflected models for cold-start recommender systems," *Decision Support Systems* (51:3), pp. 519–531.

King, D. E. 2009. "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research* (10), pp. 1755–1758.

King, T., and Dennis, C. 2006. "Unethical consumers: Deshopping behaviour using the qualitative analysis of theory of planned behaviour and accompanied (de) shopping," *Qualitative Market Research: An International Journal* (9:3), pp. 282–296.

Kisilevich, S., Keim, D., and Rokach, L. 2013. "A GIS-based decision support system for hotel room rate estimation and temporal price prediction: The hotel brokers' context," *Decision Support Systems* (54:2), pp. 1119–1133.

Kotler, P., and Levy, S. J. 1971. "Demarketing, yes, demarketing," *Harvard Business Review* (49:6), p. 74.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. "Imagenet classification with deep convolutional neural networks," *1097–1105* .

Lam, M. 2004. "Neural network techniques for financial performance prediction: integrating fundamental and technical analysis," *Decision Support Systems* (37:4), pp. 567–581.

LaRose, R. 2001. "On the Negative effects of E-commerce: A sociocognitive Exploration of Unregulated Online Buying," *Journal of Computer-Mediated Communication* (6:3), pp. 1–37.

Lau, H. C. W., Ho, G. T. S., and Zhao, Y. 2013. "A demand forecast model using a combination of surrogate data analysis and optimal neural network approach," *Decision Support Systems* (54:3), pp. 1404–1416.

Lee, H., Lee, Y., Cho, H., Im, K., and Kim, Y. S. 2011. "Mining churning behaviors and developing retention strategies based on a partial least squares (PLS) model," *Decision Support Systems* (52:1), pp. 207–216.

Lee, Y.-H., Wei, C.-P., Cheng, T.-H., and Yang, C.-T. 2012. "Nearest-neighbor-based approach to time-series classification," *Decision Support Systems* (53:1), pp. 207–217.

Li, C., Ye, H., Wang, G., and Zhang, J. 2005. "A recursive nonlinear PLS algorithm for adaptive nonlinear process modeling," *Chemical engineering & technology* (28:2), pp. 141–152.

Li, D.-C., Chang, C.-C., and Liu, C.-W. 2012a. "Using structure-based data transformation method to improve prediction accuracies for small data sets," *Decision Support Systems* (52:3), pp. 748–756.

Li, N., and Wu, D. D. 2010. "Using text mining and sentiment analysis for online forums hotspot detection and forecast," *Decision Support Systems* (48:2), pp. 354–368.

Li, Q., Wang, T., Gong, Q., Chen, Y., Lin, Z., and Song, S.-k. 2014. "Media-aware quantitative trading based on public Web information," *Decision Support Systems* (61), pp. 93–105.

Li, X., and Chen, H. 2013. "Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach," *Decision Support Systems* (54:2), pp. 880–890.

Li, Y., Wei, C., and Cai, X. 2012b. "Optimal pricing and order policies with B2B product returns for fashion products," *International Journal of Production Economics* (135:2), pp. 637–646.

Li, Y., Xu, L., and Li, D. 2013. "Examining relationships between the return policy, product quality, and pricing strategy in online direct selling," *International Journal of Production Economics* (144:2), pp. 451–460.

Lin, M., Lucas Jr, H. C., and Shmueli, G. 2013. "Research commentary-too big to fail: large samples and the p-value problem," *Information Systems Research* (24:4), pp. 906–917.

Liu, N., Cao, J., Koh, Z. X., Lin, Z., and Ong, M. E. H. 2015. "Analysis of patient outcome using ECG and extreme learning machine ensemble," in *Digital Signal Processing (DSP), 2015 IEEE International Conference on*, , IEEE.

Lohmöller, J.-B. 1989. *Latent variable path modeling with partial least squares*, Heidelberg: Physica-Verlag.

Lowe, D. 1989. "Adaptive radial basis function nonlinearities, and the problem of generalisation," in *Artificial Neural Networks, 1989., First IEE International Conference on (Conf. Publ. No. 313)*, , IET.

Lu, C.-J., Lee, T.-S., and Lian, C.-M. 2012. "Sales forecasting for computer wholesalers: A comparison of multivariate adaptive regression splines and artificial neural networks," *Decision Support Systems* (54:1), pp. 584–596.

Lu, L., Lin, H., Tian, L., Yang, W., Sun, J., and Liu, Q. 2009. "Time series analysis of dengue fever and weather in Guangzhou, China," *BMC Public Health* (9:1), p. 395.

Mangiameli, P., West, D., and Rampal, R. 2004. "Model selection for medical diagnosis decision support systems," *Decision Support Systems* (36:3), pp. 247–259.

Martens, D., Provost, F., Clark, J., and de Fortuny, E. J. forthcoming. "Mining Massive Fine-Grained Behavior Data to Improve Predictive Analytics," *MIS Quarterly* (forthcoming).

Martens, J. 2010. "Deep learning via Hessian-free optimization," *Proceedings of the 27th International Conference on Machine Learning, 2010* pp. 735–742.

Martens, J., and Sutskever, I. 2011. "Learning recurrent neural networks with hessian-free optimization," *Proceedings of the 28th International Conference on Machine Learning, 2011* pp. 1033–1040.

Matsumoto, M., and Nishimura, T. 1998. "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)* (8:1), pp. 3–30.

Mc Cracken, M. W. 2000. "Robust out-of-sample inference," *Journal of Econometrics* (99:2), pp. 195–223.

Mehta, K., and Bhattacharyya, S. 2004. "Adequacy of training data for evolutionary mining of trading rules," *Decision Support Systems* (37:4), pp. 461–474.

Minsky, M. L. 1991. "Logical versus analogical or symbolic versus connectionist or neat versus scruffy," *AI magazine* (12:2), p. 34.

Moe, W. W. 2003. "Buying, searching, or browsing: Differentiating between online shoppers using in-store navigational clickstream," *Journal of Consumer Psychology* (13:1), pp. 29–39.

Mollenkopf, D. A., Rabinovich, E., Laseter, T. M., and Boyer, K. K. 2007. "Managing internet product returns: a focus on effective service operations," *Decision Sciences* (38:2), pp. 215–250.

Mukhopadhyay, S. K., and Setoputro, R. 2004. "Reverse logistics in e-business: optimal price and return policy," *International Journal of Physical Distribution & Logistics Management* (34:1), pp. 70–89.

Murtagh, F., Starck, J., and Renaud, O. 2004. "On neuro-wavelet modeling," *Decision Support Systems* (37:4), pp. 475–484.

Nauck, D., and Kruse, R. 1997. "A neuro-fuzzy method to learn fuzzy classification rules from data," *Fuzzy sets and Systems* (89:3), pp. 277–288.

Nelson, C. R. 1972. "The prediction performance of the FRB-MIT-PENN model of the US economy," *The American Economic Review* pp. 902–917.

Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W. 1996. *Applied linear statistical models*, vol. 4, Irwin Chicago.

North, C. 2006. "Toward measuring visualization insight," *Computer Graphics and Applications, IEEE* (26:3), pp. 6–9.

Oh, C., and Sheng, O. 2011. "Investigating Predictive Power of Stock Micro Blog Sentiment in Forecasting Future Stock Price Directional Movement," .

Okada, E. M. 2005. "Justification effects on consumer choice of hedonic and utilitarian goods," *Journal of marketing research* (42:1), pp. 43–53.

Olden, J. D., and Jackson, D. A. 2002. "Illuminating the black box: A randomization approach for understanding variable contributions in artificial neural networks," *Ecological Modelling* (154:1), pp. 135–150.

Olson, D. L., and Chae, B. 2012. "Direct marketing decision support through predictive customer response modeling," *Decision Support Systems* (54:1), pp. 443–451.

Özesmi, S. L., and Özesmi, U. 1999. "An artificial neural network approach to spatial habitat modelling with interspecific interaction," *Ecological Modelling* (116:1), pp. 15–31.

Oztekin, A., Kong, Z. J., and Delen, D. 2011. "Development of a structural equation modeling-based decision tree methodology for the analysis of lung transplantations," *Decision Support Systems* (51:1), pp. 155–166.

Padmanabhan, V., and Png, I. P. 1997. "Manufacturer's return policies and retail competition," *Marketing Science* (16:1), pp. 81–94.

Pao, Y.-H., and Takefji, Y. 1992. "Functional-link net computing," *IEEE Computer Journal* (25:5), pp. 76–79.

Papakiriakopoulos, D., Pramatari, K., and Doukidis, G. 2009. "A decision support system for detecting products missing from the shelf based on heuristic rules," *Decision Support Systems* (46:3), pp. 685–694.

Park, J., and Sandberg, I. W. 1991. "Universal approximation using radial-basis-function networks," *Neural Computation* (3:2), pp. 246–257.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., and Dubourg, V. 2011. "Scikit-learn: Machine learning in Python," *The Journal of Machine Learning Research* (12), pp. 2825–2830.

Pei, Z., Paswan, A., and Yan, R. 2014. "E-tailer's return policy, consumer's perception of return policy fairness and purchase intention," *Journal of Retailing and Consumer Services* (21:3), pp. 249–257.

Petersen, J. A., and Kumar, V. 2009. "Are product returns a necessary evil? Antecedents and consequences," *Journal of Marketing* (73:3), pp. 35–51.

Plaisant, C. 2004. "The challenge of information visualization evaluation," in *Proceedings of the Working Conference on Advanced Visual Interfaces*, , ACM.

Popper, K. 2005. *The Logic of Scientific Discovery*, Routledge.

Power, D. J., Sharda, R., and Burstein, F. 2002. *Decision Support Systems*, Wiley Online Library.

Pur, S., Stahl, E., Wittmann, M., Wittmann, G., and Weinfurter, S. 2013. *Retourenmanagement im Online Handel - Das Beste daraus machen*, Regensburg: ibi research an der Universität Regensburg GmbH.

Quinlan, J. R. 2014. *C4.5: Programs for machine learning*, Elsevier.

Rabinovich, E., Sinha, R., and Laseter, T. 2011. "Unlimited shelf space in Internet supply chains: Treasure trove or wasteland?" *Journal of Operations Management* (29:4), pp. 305–317.

Ramachandran, P., and Varoquaux, G. 2011. "Mayavi: 3D visualization of scientific data," *Computing in Science & Engineering* (13:2), pp. 40–51.

Rigby, D. 2014. "Online Shopping Isn't as Profitable as You Think," `https://hbr.org/2014/08/online-shopping-isnt-as-profitable-as-you-think/`, accessed: 2015-04-20.

Rosenblatt, F. 1958. "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review* (65:6), p. 386.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. 2004. "Learning internal representations by error propagation," in *Parallel distributed processing, vol. 1*, D. E. Rumelhart and J. L. McClelland (eds.), MIT Press, pp. 318–362.

Saar-Tsechansky, M., and Provost, F. 2007. "Decision-Centric Active Learning of Binary-Outcome Models," *Information Systems Research* (18:1), pp. 4–22.

Sahoo, N., Singh, P. V., and Mukhopadhyay, T. 2012. "A Hidden Markov Model for Collaborative Filtering." *MIS Quarterly* (36:4), pp. 1329–1356.

Sainath, T. N., Vinyals, O., Senior, A., and Sak, H. 2015. "Convolutional, long short-term memory, fully connected deep neural networks," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 4580–4584.

Schmidhuber, J. 2015. "Deep learning in neural networks: An overview," *Neural Networks* (61), pp. 85–117.

Schmidhuber, J., Wierstra, D., and Gomez, F. 2005. "Evolino: Hybrid neuroevolution/optimal linear search for sequence learning," *Proceedings of the 19th International Joint Conference on Artificial intelligence* pp. 853–858.

Schmidt, R. A., Sturrock, F., Ward, P., and Lea-Greenwood, G. 1999. "Deshopping-the art of illicit consumption," *International Journal of Retail & Distribution Management* (27:8), pp. 290–301.

Schumaker, R. P. 2013. "Machine learning the harness track: Crowdsourcing and varying race history," *Decision Support Systems* (54:3), pp. 1370–1379.

Searle, S. R. 2012. *Linear models*, Wiley.

Sermpinis, G., Dunis, C., Laws, J., and Stasinakis, C. 2012. "Forecasting and trading the EUR/USD exchange rate with stochastic Neural Network combination and time-varying leverage," *Decision Support Systems* (54:1), pp. 316–329.

Serrano-Cinca, C., and Gutiérrez-Nieto, B. 2013. "Partial Least Square Discriminant Analysis for bankruptcy prediction," *Decision Support Systems* (54:3), pp. 1245–1255.

Setiono, R., Leow, W. K., and Thong, J. Y. 2000. "Opening the neural network black box: An algorithm for extracting rules from function approximating artificial neural networks," *Proceedings of the Twenty First International Conference on Information Systems* pp. 176–186.

Shen, L., and Loh, H. T. 2004. "Applying rough sets to market timing decisions," *Decision Support Systems* (37:4), pp. 583–597.

Shin, H., Hou, T., Park, K., Park, C.-K., and Choi, S. 2013. "Prediction of movement direction in crude oil prices based on semi-supervised learning," *Decision Support Systems* (55:1), pp. 348–358.

Shmueli, G., and Koppius, O. 2010. "Predictive analytics in information systems research," *Robert H Smith School Research Paper No RHS* pp. 06–138.

Shmueli, G., and Koppius, O. R. 2011. "Predictive Analytics in Information Systems Research." *MIS Quarterly* (35:3), pp. 553–572.

Shulman, J. D., Coughlan, A. T., and Savaskan, R. C. 2009. "Optimal restocking fees and information provision in an integrated demand-supply model of product returns," *Manufacturing & Service Operations Management* (11:4), pp. 577–594.

Shulman, J. D., Coughlan, A. T., and Savaskan, R. C. 2011. "Managing Consumer Returns in a Competitive Environment," *Management Science* (57:2), pp. 347–362.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. 2016. "Mastering the game of Go with deep neural networks and tree search," *Nature* (529:7587), pp. 484–489.

Sinha, A. P., and May, J. H. 2004. "Evaluating and tuning predictive data mining models using receiver operating characteristic curves," *Journal of Management Information Systems* (21:3), pp. 249–280.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.-Y., Hjalmarsson, H., and Juditsky, A. 1995. "Nonlinear black-box modeling in system identification: a unified overview," *Automatica* (31:12), pp. 1691–1724.

Song, L., Cherrett, T., McLeod, F., and Guan, W. 2009. "Addressing the Last Mile Problem," *Transportation Research Record: Journal of the Transportation Research Board* (2097), pp. 9–18.

Speights, D., and Hilinski, M. 2005. "Return fraud and abuse: How to protect profits," *Retailing Issues Letter* (17:1), pp. 1–6.

Sprague Jr, R. H. 1980. "A framework for the development of decision support systems," *MIS Quarterly* pp. 1–26.

Stock, J., Speh, T., and Shear, H. 2006. "Managing product returns for competitive advantage," *MIT Sloan Management Review* (48:1), pp. 57–62.

Straub, D. W., Gefen, D., and Boudreau, M.-C. 2005. "Quantitative research," *Research in information systems: a handbook for research supervisors and their students* (1).

Su, P., Mao, W., Zeng, D., and Zhao, H. 2012. "Mining actionable behavioral rules," *Decision Support Systems* (54:1), pp. 142–152.

Sun, J., and Li, H. 2008. "Listed companies' financial distress prediction based on weighted majority voting combination of multiple classifiers," *Expert Systems with Applications* (35:3), pp. 818–827.

Sun, Z.-L., Choi, T.-M., Au, K.-F., and Yu, Y. 2008. "Sales forecasting using extreme learning machine with applications in fashion retailing," *Decision Support Systems* (46:1), pp. 411–419.

Sussillo, D., and Barak, O. 2013. "Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks," *Neural Computation* (25:3), pp. 626–649.

Suwelack, T., Hogreve, J., and Hoyer, W. D. 2011. "Understanding money-back guarantees: cognitive, affective, and behavioral outcomes," *Journal of Retailing* (87:4), pp. 462–478.

Suykens, J. A., and Vandewalle, J. P. 2012. *Nonlinear Modeling: Advanced black-box techniques*, Springer Science & Business Media.

Thomassey, S., and Fiordaliso, A. 2006. "A hybrid sales forecasting system based on clustering and decision trees," *Decision Support Systems* (42:1), pp. 408–421.

Toktay, L. B., van der Laan, E. A., and de Brito, M. P. 2004. *Managing product returns: the role of forecasting*, Springer.

Urbanke, P., Kranz, J., and Kolbe, L. 2014. "A Unified Statistical Framework for Evaluating Predictive Methods," .

Urbanke, P., Kranz, J., and Kolbe, L. 2015a. "Predicting Product Returns in E-Commerce: The Contribution of Mahalanobis Feature Extraction," *Proceedings of the 36th International Conference on Information Systems (ICIS), Fort Worth, December 13-16, 2015* pp. 1–12.

Urbanke, P., Kranz, J., and Kolbe, L. 2015b. "Predicting Product Returns in Online Retail: The Contribution of Mahalanobis Feature Extraction," .

Venturini, G. 1993. "SIA: a supervised inductive algorithm with genetic search for learning attributes based concepts," *Proceedings of the European Conference on Machine Learning, 1993* pp. 280–296.

Vinzi, V. E., Trinchera, L., and Amato, S. 2010. *PLS path modeling: from foundations to recent developments and open issues for model assessment and improvement*, Springer.

Wachter, K., Vitell, S. J., Shelton, R. K., and Park, K. 2012. "Exploring consumer orientation toward returns: unethical dimensions," *Business Ethics: A European Review* (21:1), pp. 115–128.

Walsh, G., Möhring, M., Koot, C., and Schaarschmidt, M. 2014. "Preventive product returns management systems - A review and model," *Proceedings of the European Conference on Information Systems (ECIS) 2014, Tel Aviv, Israel, June 9-11* pp. 1–12.

Wan, C., Xu, Z., Pinson, P., Dong, Z. Y., and Wong, K. P. 2014. "Probabilistic forecasting of wind power generation using extreme learning machine," *Power Systems, IEEE Transactions on* (29:3), pp. 1033–1044.

Wang, L. P., and Wan, C. R. 2008. "Comments and Replies," *IEEE Transactions on Neural Networks* (19:8), p. 1495.

Watson, H., and Wixom, B. H. 2007. "The Current State of Business Intelligence," *Computer* (40:9), pp. 96–99.

Weber, C., Hendrickson, C., Jaramillo, P., Matthews, S., Nagengast, A., and Nealer, R. 2008. "Life cycle comparison of traditional retail and e-commerce logistics for electronic products: A case study of buy. com," *Green Design Institute, Carnegie Mellon University* .

West, K. D. 1996. "Asymptotic inference about predictive ability," *Econometrica: Journal of the Econometric Society* pp. 1067–1084.

Wettschereck, D., and Dietterich, T. 1991. "Improving the performance of radial basis function networks by learning center locations," in *NIPS*, vol. 4, , Citeseer, vol. 4.

Williams, E., and Tagami, T. 2002. "Energy Use in Sales and Distribution via E-Commerce and Conventional Retail: A Case Study of the Japanese Book Sector," *Journal of Industrial Ecology* (6:2), pp. 99–114.

Wilson, D., Irwin, G., and Lightbody, G. 1997. "Nonlinear PLS using radial basis functions," *Transactions of the Institute of Measurement and Control* (19:4), pp. 211–220.

Wilson, S. W. 1995. "Classifier fitness based on accuracy," *Evolutionary computation* (3:2), pp. 149–175.

Wilson, S. W. 2000. "Get real! XCS with continuous-valued inputs," in *Learning Classifier Systems*, , Springer, pp. 209–219.

Wold, H. 1966. "Estimation of principal components and related models by iterative least squares," in *Multivariate Analysis*, P. Krishnaiaah (ed.), New York: Academic Press, pp. 391–420.

Wold, H. 1975. "Soft modeling by latent variables: the nonlinear iterative partial least squares approach," in *Perspectives in probability and statistics, papers in honor of M. S. Bartlett*, J. Gani (ed.), London: Academic Press, pp. 117–142.

Wold, S. 1992. "Nonlinear partial least squares modelling II. Spline inner relation," *Chemometrics and Intelligent Laboratory Systems* (14:1-3), pp. 71–84.

Wold, S., Kettaneh-Wold, N., and Skagerberg, B. 1989. "Nonlinear PLS modeling," *Chemometrics and Intelligent Laboratory Systems* (7:1-2), pp. 53–65.

Wood, S. L. 2001. "Remote purchase environments: the influence of return policy leniency on two-stage decision processes," *Journal of Marketing Research* (38:2), pp. 157–169.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y., et al. 2008. "Top 10 algorithms in data mining," *Knowledge and Information Systems* (14:1), pp. 1–37.

Yan, R. 2009. "Product categories, returns policy and pricing strategy for e-marketers," *Journal of Product & Brand Management* (18:6), pp. 452–460.

Yang, C.-S., Wei, C.-P., Yuan, C.-C., and Schoung, J.-Y. 2010. "Predicting the length of hospital stay of burn patients: Comparisons of prediction accuracy among different clinical stages," *Decision Support Systems* (50:1), pp. 325–335.

Yen, S. M.-F., and Hsu, Y.-L. 2010. "Profitability of technical analysis in financial and commodity futures markets — A reality check," *Decision Support Systems* (50), pp. 128–139.

Yolcu, U., Egrioglu, E., and Aladag, C. H. 2013. "A new linear & nonlinear artificial neural network model for time series forecasting," *Decision Support Systems* (54:3), pp. 1340–1347.

Yu, C.-C., and Wang, C.-S. 2008. "A hybrid mining approach for optimizing returns policies in e-retailing," *Expert Systems with Applications* (35:4), pp. 1575–1582.

Zhang, D., Zhou, X., Leung, S. C., and Zheng, J. 2010. "Vertical bagging decision trees model for credit scoring," *Expert Systems with Applications* (37:12), pp. 7838–7843.

Zhao, H., Sinha, A. P., and Bansal, G. 2011. "An extended tuning method for cost-sensitive regression and forecasting," *Decision Support Systems* (51:3), pp. 372–383.

Zhong, M. S., Pick, R. A., Klein, G., and Jiang, J. J. 2005. "Vector Error-Correction Models in a consumer packaged goods category forecasting decision support system," *Journal of Computer Information Systems* (46:1), pp. 25–34.

Zhou, H., Reid, R. D., and Benton Jr., W. 2006. "The drivers of product return in the information age," *International Journal of Internet and Enterprise Management* (4:2), pp. 100–117.

Zhou, L., Burgoon, J. K., Twitchell, D. P., Qin, T., and Nunamaker Jr, J. F. 2004. "A comparison of classification methods for predicting deception in computer-mediated communication," *Journal of Management Information Systems* (20:4), pp. 139–166.