FABER, ANDREAS D.

DATA ANALYTICS IN SUPPLY CHAIN PLANNING: APPLICATIONS IN INTERMITTENT DEMAND FORECASTING, PARTIAL DEFECTION PREDICTION AND PRICE DISCRIMINATION

Dissertation
for obtaining the degree of Doctor of Business and Economics
(Doctor rerum politicarum – Dr. rer. pol.)

at WHU – Otto Beisheim School of Management

October 24, 2019

First Advisor:	Prof. Dr. Stefan Spinler
Second Advisor:	Prof. Dr. Arnd Huchzermeier

Faber, Andreas D.: *Data analytics in supply chain planning: Applications in intermittent demand forecasting, partial defection prediction and price discrimination,*

To my wife Isabell
for her love and support.

ABSTRACT

This dissertation investigates different applications of *data analytics* in *supply chain planning*. In the last years, data analytics became more important, because of the increase of computational power and the larger availability of data. Data analytics is used in various domains to improve operations performance, increase customer satisfaction and revenues. However, both the research and the application of data analytics in supply chain management is still lacking behind other industries.

We[1] analyze the potential of data analytics in the field of supply chain planning in three exemplary fields: *demand forecasting*, *partial defection prediction* and *price discrimination*. In addition, we demonstrate how to deal with three common challenges in the field of data analytics: *the manual effort for method selection and hyperparameter tuning*, *the difficult interpretability of machine learning methods* and *the risks associated with data collection through randomized experiments*.

In the *first paper*, we develop a method selection approach in the field of *intermittent demand prediction*. Our model combines high predictive performance with automation and calculation efficiency. Unlike common practice, the prediction method gets automatically chosen for each data set without any manual selection. Our results are stable across three different data sets that come from different sources but all contain intermittent demand time series. We showcase the impact of the proposed forecasting approach with a warehouse operation simulation. We thereby prove the financial benefit with empirical data.

In the *second paper*, we deal with *partial defection prediction* in a business-to-business environment in the logistics industry. The predictions must combine predictive performance with interpretability and profit maximization. Our model uses a large variety of customer-based and time-series-based features to predict the probability of partial defection for each customer. We use a data permutation approach to make the best performing, black-box models interpretable. Furthermore, we use a profit assessment to identify the method that leads to the highest revenue through successful retention actions.

In the *third paper*, we study *price sensitivity prediction*. We do not use any randomized experiments, because the risk of loosing customers through such experiments is too high. Thereby, we address the challenge of data availability

[1]The term "we" refers to the authors of the respective chapters as denoted at the beginning of each chapter. For the abstract, this refers to the authors of Faber and Spinler (2019a,b,c).

for analytics. We find that we are able to circumvent the non-availability of experimental data by using historical observations.

The findings from all three research areas show significant improvement potential regarding the predictive performance under the special conditions that we study (automation, interpretability, data availability). We use empirical data from different research partners to assess the financial impact of the proposed methods. Overall, we demonstrate the effectiveness and value of *data analytics* in *supply chain planning*.

ACKNOWLEDGEMENTS

This thesis was written as a quasi-cumulative dissertation in fulfillment of the requirements for the doctoral degree at the Kühne Institute for Logistics Management at WHU – Otto Beisheim School of Management between 2016 and 2019.

First of all, I would like to thank my first advisor Prof. Dr. Stefan Spinler for his continuous support throughout the course of my doctoral studies. He helped to shape the research topic, provided constructive feedback whenever needed, gave methodological advice and introduced me to helpful contacts both in academia and industry. Moreover, he gave me the chance to hold multiple lections in his classes for Master and MBA students and in the course of the Digital at Scale training program for executives.

I would also like to thank Prof. Dr. Arnd Huchzermeier for serving in the capacity of my second advisor and for providing helpful perspectives on the topic of this thesis.

Further, I would like to mention the entire team at the Kühne Institute for Logistics Management for all the fruitful discussions we had every time I came to Vallendar. You made me feel part of the team even as an external doctoral student. Thank you, Anna Achenbach, Maximilian Burkhardt, Stefan Schulze-Schwering and Alexander Hess. A special thanks goes to Linda Stein for her administrative support.

I would also like to thank my research partners who provided the data for my analysis and helped to generate the findings of this thesis.

Finally, I want to thank my family, my wife and my close friends for their support throughout the years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AIC	Akaike information criterion
AICc	Akaike information criterion corrected for small sample sizes
ALBA	Active learning-based approach
ARIMA	Autoregressive integrated moving average
AUC	Area under the curve
B2B	Business-to-business
B2C	Business-to-consumer
BRNN	Bidirectional recurrent neural networks
CART	Classification and regression tree
CLV	Customer lifetime value
CRO	Croston method
DT	Decision tree
ETS	Exponential smoothing state space model
FFNN	Feedforward neural networks
FP	False positive
FN	False negative
GBM	Gradient boosting models
LIME	Local interpretable model-agnostic explanations
LinR	Linear regression
LogR	Logistic regression
MAE	Mean absolute error
MASE	Mean absolute scaled error
NN	Neural networks

PCA Principal component analysis

PRC Precision recall characteristic

RAF Royal Air Force

RBF Radial basis function

RF Random forest

RFE Recursive feature elimination

RMSE Root mean squared error

ROC Receiver operating characteristic

SBA Syntetos Boylan approximation

sBG Shifted beta geometric

SES Single exponential smoothing

SKU Stock keeping unit

SMA Simple moving average

SMOTE Synthetic minority oversampling technique

SOM Self-organizing maps

STL Seasonal and trend decomposition using Loess

SVM Support vector machines

TDL Top decile lift

TDP Top decile profit

TP True positive

TN True negative

1

INTRODUCTION

1.1 BIG DATA ANALYTICS

Big data analytics has significantly gained in importance over the last years (Wang et al. 2016), mainly because of two reasons: First, the amount of available data is much bigger nowadays. Data is even considered to be the new oil (Yi et al. 2014) that drives economic value. Second, there is now a broad understanding of the opportunities that arise through big data analytics and lead to value creation (Brown et al. 2011), more transparency (Vassakis et al. 2018), or better informed decision making (Chen and Zhang 2014). Several scholars highlight that big data analytics becomes more and more popular in academia as well as in the industry (Arora and Malik 2015, Vassakis et al. 2018). A recent report from the management consulting firm *McKinsey & Company* finds the highest potential value through analytics in *marketing and sales*, and *supply chain management and manufacturing* (Chui et al. 2018). According to their insights, future use cases cover areas such as *price and promotion, predictive maintenance, sales and demand forecast* or *partial defection reduction*.

Although the concepts of *big data* and *data analytics* are often closely linked to each other, it is important to understand the difference between both fields.

Chen and Zhang (2014) define the term *big data* as very huge data sets with a great diversity of data types that makes it difficult to analyze with traditional methods. Laney (2001) establishes the common concept of the 3V's that define big data: *volume, velocity* and *variety*. Other scholars add further characteristics such as *variability, veracity* or *visualization* (Vassakis et al. 2018). These data sets consist of structured and unstructured data from different sources (e.g., sensors, social media sites) (Sivarajah et al. 2017).

Data analytics or *data science* or *data mining* is an approach to gain knowledge from data (Vassakis et al. 2018). The underlying concept dates back to the 1950's when first tools were introduced to discover patterns in data (Vassakis et al. 2018). The focus increased since the mid-2000's with the advent of big data. Davenport (2013) distinguishes three types of analytics: 1) *descriptive analytics* to extract useful information from past data, 2) *predictive analytics* to predict future developments based on patterns in the data, and 3) *prescriptive analytics* to provide recommendations for decision-making. Among the most common

data analytics methods are *machine learning* techniques (Chen and Zhang 2014). These methods allow to extract patterns from data and thereby learn relationships that help to forecast future events (Witten et al. 2016). They are used in cases when either human expertise is not present or the human expertise cannot be easily explained (e.g., in the case of speech recognition) (Alpaydin 2009). Alpaydin (2009) states that there are several research areas such as statistics, signal processing and pattern recognition that are combined in machine learning. Its application areas cover various domains with topics such as medical diagnosis or credit risk prediction. Davenport and Ronanki (2018) distinguish between three application areas: Process automation, gaining insight through data analysis, and engaging with customers and employees.

1.2 CHALLENGES OF BIG DATA ANALYTICS

The application of big data analytics also comes with challenges. Arora and Malik (2015) state that it is still difficult to extract meaningful information from big data. Sivarajah et al. (2017) adapt the findings from Zicari (2014) and Akerkar (2013) and present three areas of challenges based on the data lifecycle: *data challenges* (related to the characteristics of the data), *process challenges* (challenges to process the data) and *management challenges* (challenges to understand and analyze the data).

The *data challenges* are directly linked to the big data concept (volume, variety, veracity, velocity, variability and visualization) and describe how difficult it is to deal with such data sets. One needs to combine data from different sources with different types of data which can be structured as well as unstructured.

The *process challenges* are structured along the different process steps (1. data acquisition and warehousing, 2. data mining and cleaning, 3. data aggregation and integration, 4. data analysis and modelling, 5. data interpretation). Important factors are the collection of the right and meaningful data in a cost efficient manner, the selection of the right analytics approach to deal with large and diverse data sets and finally the interpretation of the outcomes of the analysis. Zhou et al. (2017) highlight the common trade-off between highly transparent and interpretable models and the ones that lead to higher predictive performance but are less accessible. They propose that one should not only focus on the accuracy evaluation measures but also take other factors such as interpretability, efficiency or stability into consideration.

The *management challenges* cover aspects such as privacy, security, data governance, data and information sharing, cost and data ownership. Organizations

need to make sure that the infrastructure is compliant with all security and privacy standards. Vassakis et al. (2018) further highlight that culture and capabilities in an organization are other crucial factors that can hinder successful data analytics.

1.3 BIG DATA ANALYTICS IN SUPPLY CHAIN MANAGEMENT

In *supply chain management*, statistics and operations research are commonly used to improve supply chain performance (Tiwari et al. 2018). In contrast, the research in big data analytics in *supply chain management* only increased significantly since 2014 and the utilization in practice is still low (Waller and Fawcett 2013, Nguyen et al. 2018). Among the reasons for the low usage rate are the lack of suitable data (Schoenherr and Speier-Pero 2015), the low acceptance rate (Gunasekaran et al. 2017) and missing skill sets (Schoenherr and Speier-Pero 2015).

The main benefits of big data analytics in supply chain management cover better decision making, higher efficiencies, higher transparency and flexibility of supply chain processes as well as enhanced negotiation power towards suppliers and customers (Schoenherr and Speier-Pero 2015).

Nguyen et al. (2018) structure the existing research along the supply chain functions and find relevant work in procurement, manufacturing, transportation, warehousing and demand management. Wang et al. (2016) distinguish between strategic and operational supply chain decisions. The first cover strategic sourcing, supply chain network design and product design while the latter is similar to the described structure of Nguyen et al. (2018) that follows the supply chain functions.

Tiwari et al. (2018) identify future application prospects of big data analytics in supply chain planning in the fields of responsive and agile supply chains, reliable supply chains, sustainability and proactive risk response. A joint publication of *IBM* and *DHL* states that the potential of data analytics in the logistics industry is large, especially because of the network structure in which more efficient collaboration can unlock significant value (Gesing et al. 2018). As on-time and in-full shipments are crucial for its customers, *DHL* uses machine learning to predict delays and then initiate respective mitigation actions (e.g., choose a different airline carrier). New tools in demand prediction help to predict demand spikes and plan accordingly. A study by *McKinsey & Company* lists analytic capabilities among the 10 most prominent technologies to impact future warehouse operations (Dekhne et al. 2019). According to Columbus,

Louis (2019) machine learning will impact supply chain management in ten different areas that span from improvements in scheduling over routing optimization and lower inventories to higher visibility and lower fraud rates. In many of these cases, demand forecasting is the enabler. The higher the forecast accuracies get, the higher are the efficiency gains in the supply network.

1.4 MOTIVATION BEHIND THE DISSERTATION

We[1] contribute to the existing literature by applying data analytics in different fields.

Although data analytics becomes more popular both in academia and the industry, various research gaps subsist. Both, the supply chain functions in different industries, as well as the logistic industry itself, are not among the most popular research areas. Also, most scholars focus their research on business-to-consumer (B2C) cases. As business-to-business (B2B) relations are different in many terms (e.g., higher revenues per customers, longer relationships, higher value transactions), it is important to assess these independently. Another limitation of most of the existing research is the focus on predictive performance without considering other important goals such as interpretability or ease of implementation. We add to these gaps and provide insights regarding new application areas. In addition, we use empirical data from different research partners to translate the theoretical findings into evidence of the monetary value of the proposed prediction methods.

1.5 STRUCTURE OF THIS WORK

The dissertation builds on three papers that cover three different application areas of data analytics: *intermittent demand forecasting*, *partial defection prediction* and *price discrimination*. We further address a specific challenge in the field of data analytics in each of the papers: the *manual effort for method selection and hyperparameter tuning*, *the difficulty of interpretation of machine learning methods* and *the risks associated with data collection through randomized experiments*. The structure of this dissertation follows these three papers. While the first paper deals with a *business-to-consumer* (B2C) perspective, the other two papers cover *business-to-business* (B2B) use cases. In addition, we study a traditional supply

[1] The term "We" refers to the authors of the respective chapters as noted at the beginning of each chapter.

chain challenge at an *e-commerce* firm in the first paper but focus on the *logistics industry* in the other two papers.

- In Chapter 2, we study *intermittent demand prediction* as the basis for efficient warehouse operations. First, we use three different data sets to compare the predictive performance of various intermittent demand prediction methods including traditional statistical methods and machine learning methods. Second, we analyze the effect of time series decomposition and input data clustering on the predictive performance. Third, the various base predictions are combined with each other to further increase the forecast accuracy. We test simple combinations using the mean, mode or minimum of several forecasts as well as a learned weighted aggregation of different forecasts and a method selection scheme that selects a prediction method individually for each time series based on the respective time series characteristics. All combinatorial methods come with the advantage that no method needs to be selected manually. The last approach, method prediction, reduces the calculation time compared to the combinatorial methods, because only the chosen prediction method needs to be trained. Lastly, we use an empirical data set to evaluate the impact of improved demand forecasts on the warehouse operations performance.

- In Chapter 3, we develop a *partial defection prediction* method that combines high predictive accuracy with interpretability and profit optimization. We focus on a specific case, because we study partial defection in a continuous service delivery setting. Partial defection occurs when customers shift significant demand to another provider. This is a common challenge in the logistics industry, because the competition is high and the providers are interchangeable. Due to the fact that not every customer is equally important, we take the profitability of each customer into account. The outcome of our model is used to carry out retention actions by sales agents. These agents are interested to understand the reasons why customers are predicted to be defectors. Therefore, we compare the predictive performance of interpretable models vs. black-box models and then test different methods to make the best performing black-box models interpretable.

- In Chapter 4, we compare different prediction methods to understand which customers are *price sensitive* and which are not to discriminate in prices accordingly. We use a broad range of features including price information to predict how customers react to specific price increases at a cer-

tain point in time. Thereby, one can choose different pricing strategies for different customer groups depending on their price sensitivites. In order to avoid the risk of loosing customers through randomized experiments, we do not use any price experiments. Instead, we use historical data to study past reactions to different price change levels and train models to make predictions based on new data at later points in time. We further simulate a randomized experiment with the historical data to assess the financial impact of the proposed pricing discrimination scheme.

Chapter 5 contains the summary of the findings of all three main chapers including managerial implications and the major contributions in theory and practice. We further discuss potential avenues for relevant future research.

2

AN EMPIRICAL ASSESSMENT OF METHOD SELECTION FOR INTERMITTENT DEMAND PREDICTION[1]

2.1 INTRODUCTION

Supply chain management relies on forecasting and planning for demand (Fildes et al. 2009). Moon et al. (2003) argue that accurate demand forecasts improve customer satisfaction, reduce investments, and make companies more competitive. A special type of forecasting involves the prediction of intermittent demand. Forecasting intermittent demand differs from predicting "smooth" demand because the former involves uncertainty regarding not only the extent of demand but also the demand interval—as in the case of demand for, inter alia, spare parts (Teunter and Duncan 2009a). The topic has become more prominent with the shift from offline to online retail. Firms in the e-commerce sector use large assortments to drive additional revenue and improve customer experience (Morton 2017). However, a large assortment also results in many slow-moving products with intermittent demand (Chodak 2016). Brynjolfsson et al. (2009) show that the "long tail" of demand for niche books offered by Amazon increased significantly from 2000 to 2008, accounting for 37% of book sales in 2008. Other scholars have found that a large number of niche products in the video and music industry have almost zero sales (Elberse and Oberholzer-Gee 2006, Chellappa et al. 2007).

In academia, demand forecasting has been studied intensively (see e.g. De Gooijer and Hyndman 2006) with intermittent demand prediction as one specific application area. Scholars have recently begun to apply different machine learning methods in the field of intermittent demand prediction (Lolli et al. 2017). Other researchers apply meta-learning heuristics either to combine different forecasting methods or to select the most appropriate method for each focal product (Wang et al. 2009, Lemke and Gabrys 2010). Even so, there has been no exhaustive comparison of how the different intermittent prediction methods compare in terms of their forecasting accuracy. Nikolopoulos (2020) highlighted the fact that research on intermittent demand forecasting is still scarce and needs further attention also to predict scarce events.

[1]The following chapter is based on Faber and Spinler (2019a), unpublished working paper.

Here we study a method selection approach in the field of intermittent demand. We use three different data sets to demonstrate that this approach is widely applicable. For one of the data sets, we collaborate with a research partner in the e-commerce industry—a collaboration that allows us to assess the financial implication of various forecasting methods in the field of warehouse operations.

We contribute to the existing research in three respects. First, we provide a comparison of the predictive performance of various intermittent demand prediction methods, which include time-series decomposition and data input clustering. Thus we evaluate the relative predictive performance of methods that have not previously been compared with respect to a given set of intermittent demand data. Second, we demonstrate how method selection can improve predictive performance while minimizing the effort required for calculation. We show that our procedure for selecting a method is superior to a naïve selection under which prediction methods are chosen based on their past performance. In this context, we also analyze which time-series features should predispose decision makers to select a particular prediction method. Third, we simulate a warehouse operation in order to assess the financial effects of adopting our approach to the selection of a method for predicting intermittent demand.

The rest of our paper proceeds as follows. Section 2.2 reviews the literature, and in Section 2.3 we explain our approach to predicting intermittent demand. The case study, which includes a simulation of warehouse operations, is presented in Section 2.4. Section 2.5 reports on the results of our analysis. We conclude in Section 2.6 with a brief summary and some suggestions for future research.

2.2 LITERATURE REVIEW

2.2.1 *Intermittent demand prediction with time-series methods*

Demand forecasting is a well-researched topic and has been a focus of scholars for decades. De Gooijer and Hyndman (2006) review the research on time-series forecasting between 1982 and 2006; however, they report that few studies have addressed the subject of intermittent demand prediction. Intermittent demand is different from smooth demand, since the former is characterized by zero demand in many periods and irreguar demand in other periods (Eaves and Kingsman 2004). Thus the prediction of intermittent demand is complicated by the variability both of demand and of demand intervals (Petropoulos et al.

2013). It is for this reason that a branch of research dedicated to intermittent demand prediction has evolved.

In practice, such straightforward methods as simple moving average (SMA) and single exponential smoothing (SES) are still widely used to forecast intermittent demand (Petropoulos and Kourentzes 2015a). Although neither SMA nor SES is designed to work well with intermittent time series that include a large number of zero-demand periods, Wallström and Segerstedt (2010) find that single exponential smoothing works well in some such cases.

The most commonly adopted approach to intermittent demand forecasting is the Croston method (CRO), which splits the forecast into two parts: the inter-demand interval and the demand size. An SES model is used to forecast both parts, where those forecasts are not updated until demand has been observed (Croston 1972). Syntetos and Boylan (2005)) prove that Croston's approach is biased, so they modify it to correct that bias; the resulting method is known as the Syntetos–Boylan approximation (SBA). The methods of both Croston and Syntetos–Boylan are among the standard approaches to intermittent demand forecasting (Syntetos and Boylan 2005, Petropoulos and Kourentzes 2015a) and are often used as benchmarks for new methods (Kourentzes 2013). Several authors demonstrate that, on average, Croston (and its variants) outperform traditional methods in the case of intermittent demand (Willemain et al. 1994, Eaves and Kingsman 2004). Another modification to the Croston method, called Teunter, Syntetos and Babai method (TSB), was done by Teunter et al. (2011) to also incorporate cases with obsolescence when demand drops to zero. To do so, the demand estimates are updated in every period, not just when demand occured.

Another statistical prediction approach for intermittent demand is bootstrapping where past observations are randomly sampled to model the lead-time demand distribution (Syntetos et al. 2015). One of the most common methods is described by Willemain et al. (2004) who use a Markov Chain and transition probabilities between the states. In the analysis by Teunter and Duncan (2009b) bootstrapping performed as well as Croston whereas Syntetos et al. (2015) question whether bootstrapping is worth the complexity compared to simpler parametric methods such as Croston or SBA.

There is scant research also into intermittent time series that incorporate seasonality and trend patterns (Gamberini et al. 2010). In such cases, there are two popular families of prediction methods. One is the autoregressive integrated moving average (ARIMA), which is a family of methods that consists of autoregressive and moving average models that exploit the Box–Jenkins procedure (Box et al. 2015). Zhang and Qi (2005) and Gamberini et al. (2010) point out

that the robustness of these methods explains their capacity to deal with intermittent demand. Zhang and Qi compare ARIMA to neural networks—with and without prior data processing to de-seasonalize and de-trend the data. These authors establish that ARIMA performs better (resp. worse) than a neural network without (resp. with) data processing. Another family of predictive methods, exponential smoothing state space models (ETS), consists of additive and multiplicative combinations of level, trend, seasonality, and noise components; hence this approach is applicable to time series with different characteristics (Hyndman and Athanasopoulos 2014). According to De Gooijer and Hyndman (2006), ETS models are among the most reliable forecasting methods because of their robustness. Kourentzes et al. (2014b) use the exponential smoothing family across multiple aggregation frequencies to improve intermittent demand predictions. Their study reports that aggregated forecasts are superior to non-aggregated ones, but it does not compare ETS to other forecasting methods.

For short time series where application of machine learning methods is not possible, Athanasopoulos et al. (2017) suggest using Nearest Neighbors but they also state that the predictive performance becomes worse if the frequency of zero-demand periods increases.

Another approach in demand prediction is temporal aggregation that was explained by Nikolopoulos et al. (2011) and later further refined by Kourentzes et al. (2014b), Petropoulos and Kourentzes (2015b), Petropoulos et al. (2016). Data is first aggregated on higher frequencies to reduce the number of zero demand periods, then the forecast is calculated and lastly disaggregated again. The aggregation can be either overlapping or non-overlapping. The research of Petropoulos and Kourentzes (2015b) and Petropoulos et al. (2016) focus on intermittent demand prediction and describe how forecast accuracy can be improved by combining predictions of different temporal data aggregations over time or volume.

2.2.2 *Intermittent demand prediction with machine learning methods*

Machine learning–based methods are becoming more popular as an alternative to linear statistical approaches and also for intermittent demand forecasting. Gutierrez et al. (2008) report that these methods are well suited to dealing with nonlinear patterns in the data. Among the extensive variety of machine learning methods, some are widely used for the purpose of intermittent demand forecasting. Ahmed et al. (2010) provide a review and comparison of different machine learning methods that have been applied in the forecasting of smooth

time series, but we are not aware of any analogous research into intermittent demand prediction.

Artificial neural networks are popular machine learning methods that are often used in the field of intermittent demand forecasting. These methods are known to be flexible and to work well with nonlinearities in the data (Zhang et al. 1998), but they require large amounts of training data (Gutierrez et al. 2008) and have a tendency to overfit (Zhang et al. 1998). Overfitting occurs when the model becomes too specific for the training data and hence does not work well with any other data source. Feedforward neural networks (FFNNs) are the most widely adopted form of neural networks: a feedforward, multilayered perceptron that is trained by a backpropagation algorithm (Rumelhart et al. 1988). Zhang and Qi (2005) demonstrate that neural networks can work well for the prediction of seasonal and trend time series. Gutierrez et al. (2008) find also that neural networks outperform single exponential smoothing (SES), Croston method (CRO), and Syntetos Boylan approximation (SBA) in forecasting "lumpy" demand. Kourentzes (2013) reports inconsistent results when comparing neural networks with simple moving average (SMA), SES, and CRO to forecast intermittent demand. His findings are mixed because the neural network model's forecasting performance is worse than that of other models but the resultant savings in inventory costs is greater. Lolli et al. compare other forms of neural networks—including time-delay and recurrent networks in addition to applied extreme learning machines—as an alternative learning approach. These authors show that backpropagation yields better results but requires more computational effort than do extreme learning machines.

Support vector machines (SVMs) make use of a transformation of features into a high-dimensional feature space, which allows a linear model to be used in the new space that represents a nonlinear decision boundary in the original space. A penalty for complex models is added to the error function. Support vector machines are based on the principal of minimizing structural risk (i.e., minimize an upper bound of the generalization error) instead of minimizing the empirical error (Mukherjee et al. 1997). Different kernels can be used for the inner products; common choices are a linear kernel, a polynomial kernel, and the Gaussian radial basis function. Two advantages of SVMs are that (i) they generalize well and (ii) their solution excludes local minima (Bao et al. 2004). Bao et al. and Hansen et al. (2006) test the SVM approach to (intermittent) demand forecasting and find that it performs better than either the Croston or autoregressive integrated moving average (ARIMA) method.

2.2.3 *Time-series clustering for machine learning demand forecasting*

The discussion in Chen et al. (2016) suggests that a combination of input data clustering and machine learning forecasts may improve forecast accuracy. Toward that end, the full data set is divided into groups consisting of observations with similar characteristics and then, for each such cluster, separate machine learning models are trained. Chen et al. use self-organizing maps, "growing hierarchical" self-organizing maps, and k-means to cluster the data. Thomassey and Fiordaliso (2006) use k-means combined with C4.5 decision trees to generate forecasts for new products; they argue that k-means is one of the most common clustering methods for which results are generally robust.

2.2.4 *Meta learning via forecast combinations*

A popular approach to increasing forecast accuracy is to combine forecasts (Timmermann 2006). Studies have documented the success of forecast combinations, especially with regard to model building and selection (Kourentzes et al. 2014a). According to Andrawis et al. (2011), simple combinations (e.g., averaging) often work best; these authors also find that the underlying models should be diverse enough to capture different patterns in the data. A key requirement is to select appropriate base forecasting methods, since no combination can overcome the deficiencies of an inaccurate component model (Andrawis et al. 2011). In their study of forecast combinations for predicting intermittent demand, Petropoulos and Kourentzes (2015a) show that simple combinations of base forecasting methods do not perform well whereas combinations of temporally aggregated time series lead to more accurate forecasts.

A related example of meta learning is based on the sequential aggregation of experts. This model uses a polynomial "potential aggregation" rule, with different learning rates for each expert, that computes weights for the combination of different methods. The model is optimized using squared loss (Gaillard and Goude 2016). So far, its main application has been in predicting electricity consumption (Devaine et al. 2013).

2.2.5 *Meta learning via method selection*

As an alternative to forecast combinations, a meta-learning approach can be used to select a forecasting method for a specific time series. Using a variety of forecasting methods allows one to consider different sales patterns for different

products at different times. One forecasting method might yield good results for one product but not work well for another one (and vice versa). Hence overall forecasting accuracy may improve if different methods are used to forecast different types of demand patterns. This method selection is based on the work of Collopy and Armstrong (1992), whose model uses time-series characteristics and expert judgments to select a forecasting approach.

Automated feature extraction has been used to select from a range of forecasting methods, which include machine learning models (Wang et al. 2009, Lemke and Gabrys 2010). The results of these papers are mixed. On the one hand, Lemke and Gabrys's selection algorithm distinguishes between using single versus combination methods but is not designed to select a particular method. On the other hand, Wang et al. use a decision tree to identify the best forecasting method yet choose from only four different forecasting methods.

2.2.6 *Improving warehouse operations performance*

Accorsi et al. (2014) identify the two chief performance optimization areas in warehouses: warehouse design and warehouse operations. The first involves mid- and long-range decisions, such as the size and layout of a warehouse. The second covers all operational activities—of which *order picking* is perhaps the most important, since it accounts for the majority of labor and other costs (Strack and Pochet 2010). Most warehouses comprise two areas: the forward area, which is set up for fast and cheap order picking; and the reserve area, which is used to store large quantities. The forward area is capacity constrained because quick and cheap order picking is possible only in an area of limited size. Inventory from the reserve area can be replenished after it is used to increase stock in the forward area. The question of which stock keeping units (SKUs) to store in which part of the warehouse is known as the *forward-reserve problem* (Strack and Pochet 2010).

Hackman et al. (1990) discuss the allocation of products to either the capacity-constrained picking area or the unconstrained storage area. They use a "greedy knapsack"–based heuristic to solve this problem, where the solution amounts to a ranking that accounts both for picking costs and for internal replenishment costs. In a subsequent study, Frazelle et al. (1994) extend the Hackman et al. approach by considering the picking zone's area to be adjustable; these authors minimize warehouse operation costs while using material handling costs as well as equipment costs. Van den Berg et al. (1998) address the same problem under the condition of unit load replenishment. The authors model two types

of periods—busy and idle—with the goal of replenishing in idle periods only. Here, too, a greedy knapsack–based model is used to minimize the overall labor time devoted to order picking and inventory replenishment.

2.3 METHODOLOGY

Our analysis follows the three-step approach illustrated schematically in Figure 2.1. After training a broad set of base forecasting methods (step 1), we test a clustering approach for machine learning methods (2); thus we combine data from multiple products that have similar characteristics. Next, we test two meta-learning models: method combination (3A) and method selection (3B). As their inputs, these models use the predictions—of both clustered and unclustered base forecasting methods—with regard to the validation data set. For the method combinations (3A), we compare simple combination models to a more complex, learned aggregation model. The simple combinations work as follows. First, performance measures are calculated on the validation data set for all tested base prediction methods; the methods are then ranked in terms of their accuracy, and the best-performing methods are selected. With each of these methods, one predicts demand in the test period and then combines the different predictions using their mean, mode, or minimum. We test two differ-

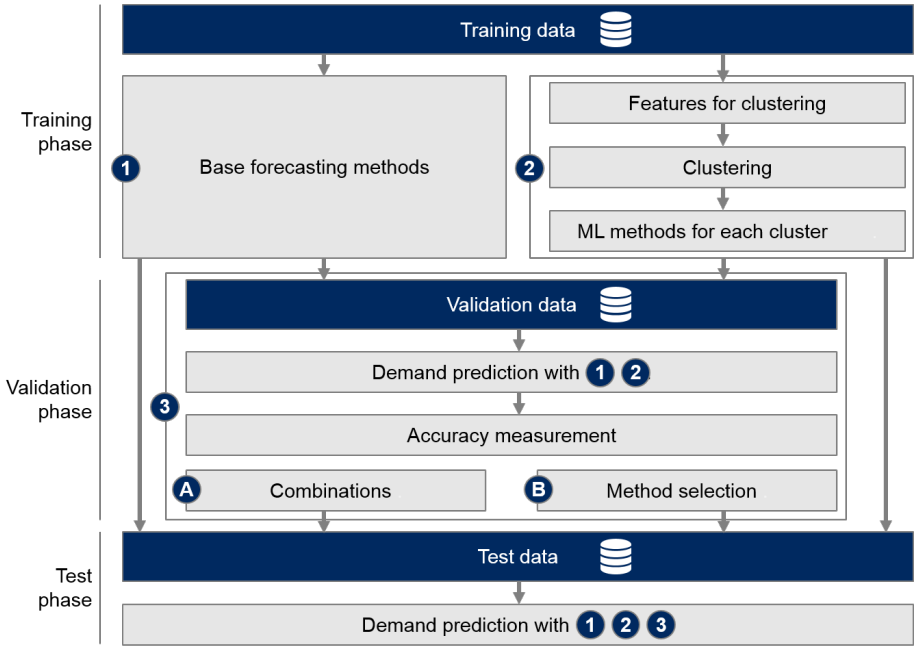

Figure 2.1: Demand prediction research approach

ent approaches to selecting which methods to combine: by comparing accuracy based on the validation data either across all SKUs or separately for each stock

keeping unit (SKU). In the former case, the same forecasting methods are selected for all SKUs; in the latter, the selected methods can differ for each SKU. One advantage of comparing across SKUs is that learning is then based on a larger data set, which reduces overfitting. Using validation data for each SKU separately is a more focused approach in that it learns from one SKU only. The "learned combination" method follows the same logic as that underlying simple combinations. In this approach, no prediction methods are selected; instead, one learns combination weights based on the measured predictive performance of the validation data. As mentioned in Section 2.2.4, the model employs a polynomial potential aggregation rule that features different learning rates for each expert and computes weights for the combination of different methods. The model is optimized using squared loss (Gaillard and Goude 2016). As before, we test one model that learns the weights across all SKUs and another model that learns the weights for each SKU separately.

As an alternative approach, we use method selection (3B). To do so, we train a random forest model using the time-series features of the training data set as the predictors while using the forecast accuracies of the validation data set as the response variable. For the final prediction on the test data set, we use the time-series features from the combined training and validation data set. We compare three different method selection strategies. For the first approach, the accuracies of all forecasting methods applied to the validation data set are compared against each other for each SKU and then coded in a binary way. Thus, for each SKU, the best method is coded with a 1 and all others with a 0. In this case, the prediction is a multiple binary classification problem with one classification model for each method. The method most likely to yield an outcome of 1 is used to calculate the forecast for the test data set. The second approach is based on a ranking of the methods for each SKU. A regession model predicts the rank for each SKU and method, and the model with the highest predicted rank is chosen for the test data set. The third approach uses a regression to predict the predictive perfomance for each SKU and each forecasting method. Here, the method with the best predicted predictive perfomance is used for the demand prediction of the respective stock keeping unit.

Finally, we compare all four approaches (1, 2, 3A, and 3B) using a separate test data set. The rationale behind splitting our data into three parts (training, validation, and test data) is to avoid any bias while building the models and selecting or combining models. Hence the training set is used to fit the model parameters, and the validation set is used to determine the rules for selecting or combining methods.

We use a TBATS model—that is, an exponential smoothing state space model with Box–Cox transformation and ARMA errors as well as trend and seasonal components—to establish the existence or absence of any seasonality patterns in the time-series data (De Livera et al. 2011). If this procedure does identify seasonality patterns for a significant share of the data set's time series, then we apply a decomposition method to separate the trend and seasonality prediction from the remainder. In particular, we use the seasonal and trend decomposition using Loess (STL), which is based on a locally weighted regression smoother and iteratively finds the seasonal and trend component in moving windows of data (Cleveland et al. 1990). This approach splits the time series into three parts: the trend component, the seasonal component, and the remainder. For each component, we employ a different forecasting approach. The seasonal component is forecast using a naïve method that takes the seasonal value from the same period in the previous year. For the trend component, we use the latest observed value as a naïve prediction; the remainder is forecast using the methods described in Section 2.2. The sum of these three separate forecasts corresponds to the final demand forecast. For all prediction models, we compare the forecast accuracy of the decomposed time series against the performance using the original time series.

2.3.1 *Accuracy measure*

We use both the root mean squared error (RMSE), as given in Eq. (2.1), and the mean absolute scaled error (MASE), as in Eq. (2.2), to measure and compare the predictive performance of our tested models. We cannot use some other well-known measures, such as the mean absolute percentage error (MAPE), because they do not work well in the case of intermittent demand: with these measures, periods of zero demand can lead to infinite or undefined values. See Hyndman et al. (2006) for a detailed explanation of performance measures in the context of intermittent demand. The RMSE is a scale-dependent measure because the error term is on the same scale as the data. In contrast, the MASE is a scale-free measure and directly compares the forecast error of a tested method with that of a naïve forecasting model. For period t, we use y_t and $\hat{y}_t$ to denote

(respectively) actual and predicted demand. Our measures are defined formally as follows:

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}; \tag{2.1}$$

$$\text{MASE} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{|y_t - \hat{y}_t|}{\frac{1}{T-1}\sum_{t=2}^{T}|y_t - y_{t-1}|}\right). \tag{2.2}$$

2.3.2 *Prediction methods*

For our intermittent demand forecasting approach, we use the methods listed in Table 2.1. Space limitations dictate that we simply refer readers to the cited authors for detailed explanations of these methods.

Table 2.1: Forecasting methods

Method	Abbreviation	Reference
Simple moving average	SMA	Makridakis et al. 2008
Single exponential smoothing	SES	Makridakis et al. 2008
Croston's method	CRO	Croston 1972
Syntetos–Boylan approximation	SBA	Syntetos and Boylan 2005
Autoregressive integrated moving average	ARIMA	Box et al. 2015
Exponential smoothing state space model	ETS	Hyndman et al. 2002
Support vector machines	SVMs	Mukherjee et al. 1997
Random forest	RF	Breiman 2001
Feedforward neural networks	FFNNs	Rumelhart et al. 1988
Bidirectional recurrent neural networks	BRNNs	Schuster and Paliwal 1997
Gradient-boosting models	GBMs	Ridgeway 2007

We use the simple but well-established models (SMA, SES, CRO, SBA) as a benchmark against which to compare the more complex and meta-learning models. In addition, we use a wide variety of different methods because extant research offers no clear guidance on which methods are superior; recall that previous studies report mixed findings along with different (and sometimes

contradicting) recommendations. We therefore conclude that the choice of a forecasting method should depend on the data set to which it will be applied. We address this issue by way of meta-learning methods, which automatically select different methods or method combinations based on different data inputs.

Besides the methods discussed in Section 2.2.2, we test three other approaches that we suppose will work well for intermittent demand prediction. Random forest (RF) is a machine learning method that consists of an ensemble of regression or classification trees. It combines "bagging" with random feature selection, and it does not involve the pruning of any trees. To avoid overfitting, the prediction results of multiple trees are averaged (Breiman 2001). The random forest approach has been used to predict smooth demand. For instance, Kane et al. (2014) find that RF outperforms ARIMA models and Herrera et al. (2010) reports that random forests perform better than feedforward neural networks but worse than support vector machines. We expect RF to work well in the field of intermittent demand prediction because the method is known to be robust and is relatively insensitive to parameter values (Dudek 2015). Bidirectional recurrent neural networks (BRNNs) are a modification of FFNNs; they are trained simultaneously in the positive and negative time direction, so they can use all information in the training data. Thus BRNNs overcome the limitation of using only information that pertains to the period *preceding* a specific input state (Schuster and Paliwal 1997). The BRNN model takes context into account because both past and future states are used to train it. When demand is intermittent, any sequence of zero-demand periods is a key factor—which is why we expect predictive performance to improve when models are trained on demand in subsequent periods. Finally, gradient-boosting models (GBMs) are examples of an ensemble learning method that builds on weak prediction models. A basis function is thereby improved in a greedy fashion to reduce the loss or error function, and each iteration uses randomly sampled data with replacement. This approach gained substantial attention owing to its good results, especially in the field of load forecasting (Taieb and Hyndman 2014). Gradient boosting is a robust method that we expect to work well with the unstable input of intermittent demand.

We use the following parameters for the prediction methods. For SMA, we use both a five-period and a nine-period average. For SES, the smoothing factor α is determined using a mean squared error minimization. For both ARIMA and STL we use the *corrected* Akaike information criterion (AICc) to automatically select the appropriate model. The Akaike information criterion (AIC) is a model selection criterion based on the Kullback–Leibler distance between the

candidate model and the true model; the AICc is the AIC with a correction for small sample sizes. For the machine learning methods compared here, all input features are scaled and also centered. We use a grid search approach to select the best "hyperparameter" combinations (i.e., machine learning model parameters that are set *before* the learning process) and perform three repetitions of our fivefold cross-validation (see the Appendix A.1 for a list of all tested hyperparameters). The parameters, which are selected independently for each trained model, consist of the chosen SKU, forecasting method, and time period.

For the simple combinations, we test outcomes from combining the top 3 and top 5 methods as determined by their performance on the validation data set.

We use k-*means clustering* to combine the training data of several SKUs and to train the forecasting models with the data for each cluster (i.e., instead of with the data for each individual SKU). The number of clusters reflects the percentage of variance that can be explained in terms of that number. This percentage is determined by calculating the within-cluster sum of squares (wss) for different numbers of clusters—after which the location of a "bend" is identified by plotting wss against k, the number of clusters (Sugar and James 2003). We use different indices (e.g., the gap statistic, the silhouette method) to determine the optimal number of clusters in a data set; for a detailed explanation of these indices, the reader is referred to Charrad et al. (2014).

2.3.3 *Time-series feature extraction*

We build our model with information from time-series data on product demand. Price or promotion features are irrelevant here because the revenue from long-tail products is too low to merit consideration of any promotion or price campaign.

Our selection of time-series features builds on preceding research in the field of intermittent demand prediction with machine learning methods (Gutierrez et al. 2008, Mukhopadhyay et al. 2012). Thus, we first control for autocorrelation in the time-series data and identify which lags are significantly correlated at the 95% level. In addition to those lags, we also use the (cumulative) number of zero-demand periods (Gutierrez et al. 2008); this factor is the basis for our "separating_zero" and "successive_zero" features defined in Table 2.2. The feature set of the *decomposed* time series differs because the rest of that time series contains no zero demand values; hence the separating_zero and successive_zero features are not used.

Table 2.2: Time-series features used for machine learning prediction

Feature	Description
Lags	Lagged demand for previous periods
separating_zero	"the number of periods separating the last two non-zero demand transaction at the end of the immediately preceeding period" (Gutierrez et al. 2008)
successive_zero	"the cumulative number of successive periods with zero demand" (Mukhopadhyay et al. 2012)

The time-series clustering and our method selection approach are both based on the time-series features described in the Appendix A.2. To determine which of these features are most vital to selecting a accurate prediction method, we use the "feature importance assessment" afforded by a trained random forest model. Thus we iterate through all features and calculate, for each one, the difference between the mean squared error of an out-of-bag data sample and that of the same data after permuting the chosen variable. The features are then ordered by the size of those calculated differences.

2.4 EXPERIMENTAL DESIGN

2.4.1 *Data*

Our analysis is based on three data sets. The first is from our research partner: a leading e-commerce company in Germany that sells mostly children's products, books, and multimedia. The context of this retailer's demand prediction approach is described in Section 2.4.3. The second data set, which is from the Royal Air Force, has been used by several other scholars (Syntetos et al. 2009, Teunter and Duncan 2009a, Nikolopoulos et al. 2011, Petropoulos and Kourentzes 2015a). Our third data set is the output of an intermittent demand simulation. Each data set contains time series for 4,000 to 5,000 different products.

We report the following descriptive statistics related to demand: the demand size or nonzero-demand vector; the average inter-demand interval; and the demand per period, including zero-demand periods. These statistics are reported for each time series in the data set from our research partner (Table 2.3), from the Royal Air Force (Table 2.4), and from our simulations (Table 2.5).

Table 2.3: Descriptive statistics: Research partner's data set

	Demand size		Inter-demand intervals		Demand per period	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
Min.	1.00	0.00	1.02	0.15	0.04	0.19
1st Q	1.32	0.66	1.45	1.25	0.29	0.62
Median	1.83	1.30	2.33	2.60	0.79	1.23
Mean	2.61	2.21	3.71	4.53	1.53	2.09
3rd Q	3.23	2.94	4.71	6.26	2.14	2.76
Max.	21.28	38.27	25.43	51.34	10.84	26.98

Table 2.4: Descriptive statistics: Royal Air Force (RAF) data set

	Demand size		Inter-demand intervals		Demand per period	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
Min.	1.00	0.00	3.82	0.00	0.04	0.19
1st Q.	1.56	0.82	7.27	5.43	0.15	0.54
Median	3.83	3.07	9.00	6.93	0.37	1.45
Mean	13.68	12.90	9.78	7.13	1.44	5.87
3rd Q.	11.33	9.35	11.57	8.62	1.15	4.45
Max.	668.00	874.42	24.00	16.46	65.08	275.71

Table 2.5: Descriptive statistics: Simulated data set

	Demand size		Inter-demand intervals		Demand per period	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
Min.	3.74	1.84	2.57	1.81	0.95	1.89
1st Q.	9.89	6.86	3.15	2.50	2.93	5.89
Median	12.49	9.43	3.32	2.73	3.73	7.70
Mean	13.31	10.59	3.34	2.77	3.99	8.43
3rd Q.	15.75	13.06	3.51	2.99	4.71	10.10
Max.	52.40	61.16	4.68	4.22	16.04	38.74

The results in Table 2.3, 2.4 and 2.5 show that the RAF data set has both the highest average demand size (for nonzero demand periods) as well as the highest average inter-demand intervals, thus being highly intermittent. In contrast, the demand size and inter-demand intervals are significantly lower for the data set from our research partner whereas the simulation data set has similar demand sizes (on average) but lower average inter-demand intervals compared to the RAF data set.

We further present two exemplary time series out of the data from our research partner in Figure 2.2 and 2.3. Both show the full time series with 357 weekly observations of the demand size with a significant share of zero demand periods.

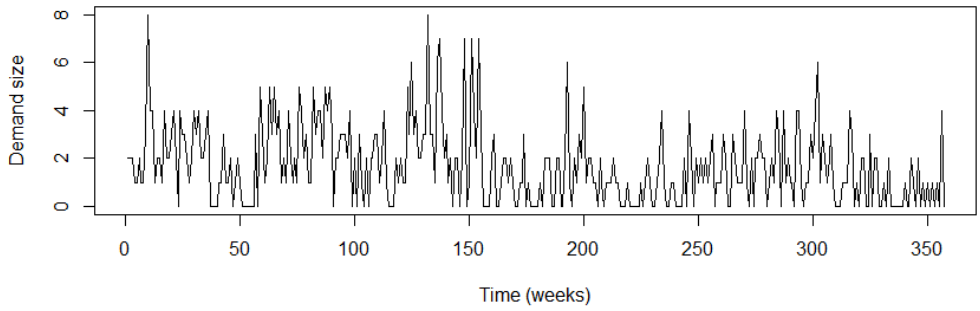

Figure 2.2: Time series example 1/2

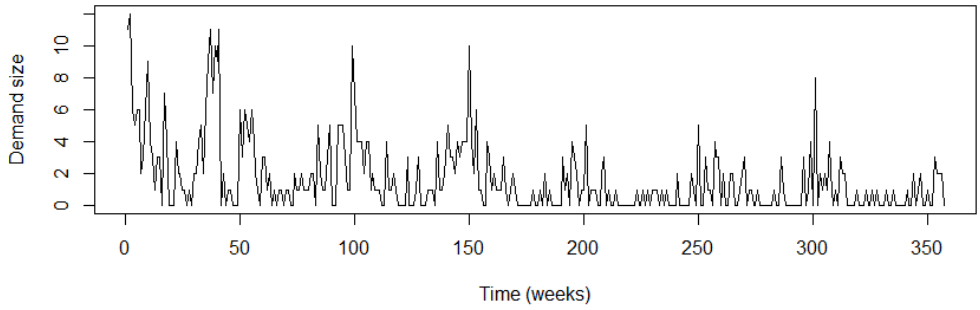

Figure 2.3: Time series example 2/2

We split each data set into three parts: a training set, a validation set, and a test set. As in Mukhopadhyay et al. (2012) and Lolli et al. (2017), we use 65% of the periods for the training set. Because our demand features are lagged, we

start the training of machine learning models in the period after the longest lag. Each validation set contains 20% of the periods in its respective data set and is used to train, evaluate, and select the meta-learning models (including method combinations and method selection); the remaining 15% of observations in each test data set are used to assess the predictive performance of our tested models.

For each data set we focus on rolling one-period demand forecasts. Thus we calculate forecasts for each time series in each data set for each period that is included in the validation or test data set. The reason for single one-period forectasts is the fact that our research partner uses such forecasts in the operational planning for its warehouse—in particular, when deciding which products to store in which area of the warehouse so as to minimize overall warehouse operation costs. Transfers of inventory stock are planned on a weekly basis.

2.4.2 *Software implementation*

The analyis for the research paper is done with the software R, using the caret package (short for classification and regression training) that utilizes a number of R packages for machine learning analysis (Kuhn 2015). We further use the tsintermittent package to calculate standard intermittent demand forecasts such as Croston (Kourentzes and Petropoulos 2016). R has the advantage of being open source and widely used for machine learning purposes (Lantz 2013.

2.4.3 *Case study context*

Our research partner's time-series data consist of daily demand for more than 4,000 SKUs between January 2010 and September 2016. Although these data do not include out-of-stock information, we assume that stockouts are rare for intermittent demand products and thus not relevant to our analysis. The data are aggregated on a weekly basis (356 weekly periods in total) to reduce intermittance and variability and to fit the predictive time frames employed.

The online retailer is interested in achieving more accurate forecasts in order to make warehousing operations more efficient and thereby reduce costs. There is only one warehouse in place, and steady demand growth has led to its capacity being almost fully used. Increasing that capacity is not possible at short notice and comes at a high cost. The retailer's current focus is therefore on increasing its existing warehouse's efficiency. The warehousing process can be briefly summarized as follows. Products can be stored in a "high-bay" bay warehouse or in a shelf-picking area; the former option is characterized

by greater storage capacity—but also by higher picking costs—than the latter option. Hence the retailer aims to store higher-demand articles in the picking area and thus to minimize both picking and replenishment costs.

We simulate warehouse operation costs by using demand forecasts (of the various predictive methods described previously) as input for the forward-reserve problem. We analyze not only the forecast accuracy measures but also the effect of forecast errors on inventory costs. To simulate storage costs, we apply a simplified model of the online retailer's actual warehousing approach. More specifically, we adopt the forward-reserve problem approach described in Section 2.2.6 but with a modification that allows for replenishments from the high-bay (reserve) area to the picking (forward) area. To avoid the case of additional replenishments within a planning period, we assume that each SKU is stored *either* in the picking area or in the high-bay warehouse. Our simulation covers the test periods, and each product's storage location is determined—using the demand prediction for each SKU—before the respective period starts. A more detailed description of warehouse operations, including cost factors, is given in the Appendix A.3.

In addition to using demand forecasts as input, we test a simple heuristic for deciding which SKUs should be stored in which area. For that purpose we undertake an "ABC analysis", as is often performed in the field of inventory control, to classify the SKUs. Thus, using mean demand as a criterion, we place 20% of the SKUs in class A, 30% in class B, and 50% in class C. Then the storage strategy consists of allocating items that have exhibited the highest mean demand (i.e., those in class A, followed by class B) to the shelf-picking area. Each SKU is assigned to a storage area only once, and no stock transfers take place.

2.5 RESULTS AND MANAGERIAL IMPLICATIONS

2.5.1 *Results of base forecasting methods*

We report the accuracy of the base forecasting methods in Table 2.6. This table reports the mean RMSE and MASE values across all SKUs in the respective data set, and each method's column rank in given in parentheses.

The GBM method does not work with the RAF data set because the number of observations in its training set is too small. In fact, we find that there is no method that works well for all data sets: the methods' rankings differ between the data sets and also between the two accuracy measures. According

Table 2.6: Predictive performance results of base forecasting methods

	Research partner data		RAF data		Simulation data	
	RMSE	MASE	RMSE	MASE	RMSE	MASE
SMA5	1.42 (10)	0.91 (5)	3.76 (8)	1.03 (3)	15.25 (11)	0.92 (13)
SMA9	1.56 (12)	1.00 (9)	3.73 (6)	1.08 (4)	14.68 (8)	0.92 (12)
SES	1.29 (1)	0.85 (3)	3.65 (4)	1.14 (5)	14.18 (3)	0.89 (6)
CRO	1.37 (8)	1.06 (12)	3.70 (5)	1.22 (7)	14.13 (2)	0.90 (11)
SBA	1.34 (7)	1.01 (10)	3.64 (3)	1.14 (6)	14.12 (1)	0.90 (10)
ARIMA	1.29 (1)	0.89 (4)	3.74 (7)	1.32 (8)	14.32 (5)	0.89 (7)
ETS	1.29 (1)	0.85 (2)	3.77 (9)	1.39 (9)	14.19 (4)	0.90 (8)
SVM-linear	1.29 (1)	0.82 (1)	3.61 (2)	0.90 (2)	15.54 (12)	0.68 (1)
SVM-radial	1.37 (8)	0.96 (6)	3.42 (1)	0.86 (1)	15.06 (10)	0.73 (2)
RF	1.32 (5)	0.98 (8)	4.54 (10)	1.47 (10)	14.56 (7)	0.88 (5)
FFNN	1.60 (13)	1.17 (13)	5.63 (12)	2.12 (12)	14.72 (9)	0.86 (4)
BRNN	1.49 (11)	1.04 (11)	5.27 (11)	2.09 (11)	15.59 (13)	0.90 (9)
GBM	1.33 (6)	0.97 (7)	—	—	14.44 (6)	0.85 (3)

the RMSE measure, SBA and SES are the most accurate; they are followed by CRO, ETS, SVM-linear, and ARIMA. Yet our MASE results indicate that both the SVM methods perform best, followed by SES, ARIMA, and SES. The SVM-linear approach is highly predictive for all data sets and for both measures; the only exception is for the RMSE applied to the simulated data, which we suppose reflects that average demand in this case is much higher than in the other data sets. Hence its average absolute error is also higher, which significantly increases the RMSE value—which also explains why all of the simulated data set's RMSE values are significantly higher than that measure's value for the data from our research partner and the RAF. Overall, we conclude that (i) the simple, traditional methods (e.g., SBA and SES) work well yet (ii) complex machine learning methods (e.g., SVM) can be effective also. Methods in the moderately complex prediction families ETS and ARIMA lead to average results that nonetheless have the virtue of being stable across all the data sets. Other tested machine learning methods (RF, FFNN, BRNN, GBM) seem unable

to cope with the data's high variability. We assume that this deficiency stems from the limited amount of information in the feature space, which contains only lagged demand and information regarding the previous inter-demand interval.

2.5.2 *Results of base forecasting methods with trend and seasonality decomposition*

Table 2.7 presents the forecast accuracies for time series decomposed by trend and seasonality. We calculate demand forecasts using the decomposed time series for all methods except for CRO and SBA (since these two methods are stymied by negative values, which occur throughout the decomposition). Neither do we use that approach for the RAF data set, which contains only a few items with seasonal demand patterns.

Table 2.7: Predictive performance results of base forecasting methods with time-series decomposition

	Research partner data		Simulation data	
	RMSE	MASE	RMSE	MASE
SMA5	1.24 (10)	0.87 (11)	13.30 (11)	0.81 (11)
SMA9	1.26 (11)	0.87 (10)	12.81 (10)	0.79 (9)
SES	1.15 (8)	0.79 (8)	12.09 (4)	0.77 (6)
ARIMA	1.11 (1)	0.76 (3)	11.99 (1)	0.76 (3)
ETS	1.14 (5)	0.77 (5)	12.01 (2)	0.77 (4)
SVM-linear	1.11 (1)	0.73 (1)	12.09 (3)	0.71 (2)
SVM-radial	1.13 (4)	0.75 (2)	12.28 (8)	0.69 (1)
RF	1.18 (9)	0.85 (9)	12.55 (9)	0.8 (10)
FFNN	1.14 (5)	0.78 (6)	12.27 (7)	0.78 (7)
BRNN	1.12 (3)	0.76 (4)	12.11 (5)	0.77 (5)
GBM	1.14 (5)	0.79 (7)	12.16 (6)	0.78 (8)

In all tested cases, predictive performance increases under time-series decomposition. The SVM-radial, FFNN, and BRNN methods achieve the highest predictive gain, whereas predictive power increases only slightly for SES, ARIMA, ETS, and RF. Because ETS and ARIMA models can deal with seasonality and trend patterns, they do not require any decomposition pre-processing. That said, the decomposed ARIMA and ETS models outperform all other tested

methods—with and without time-series decomposition—followed by SVM-linear. We therefore conclude that ETS and ARIMA are both able to extract information from the autoregressive structure of the decomposed time series. It seems that the linear SVM approach works well because it can avoid overfitting yet still handle complex data.

We apply three methods that are seldom used in the field of intermittent demand forecasting: ETS, BRNN, and GBM. The exponential smoothing state space method works well for the original time series because the model can deal with both trended and seasonal time series; however, the predictive performance of ETS is only average for time series that are decomposed. The BRNN model performs better than does the FFNN model—probably because the former is better able to learn (i.e., since both past and future states are considered when it is trained). Autocorrelation in the time-series data is what explains the FFNN's comparative disadvantage. Although the GBM model is known to be robust, it does not perform well for intermittent time series. We assume that this outcome reflects the inability of weak learners to cope with the high variance typical of intermittent data sets.

2.5.3 *Results of combined clustering and machine learning–based forecasting*

We cluster the input data to aggregate training features across similar time series. This clustering approach is tested only for the data set from our research partner, since the required calculation time is too long for our other two data sets (one must train a model for each cluster, where each model includes all of the cluster's SKUs). These circumstances also explain our inconclusive results for the clustered RF and SVM-radial methods. Our analysis is based on 14 clusters (following the approach described in Section 2.3.2 with the feature set described in Section 2.3.3). We use trend and seasonality decomposition to create the training data sets for all clustered models.

The only method for which the cluster approach yields a negative effect is SVM-linear; its RMSE increases from 1.67 to 1.68, and its MASE rises from 0.65 to 0.67). For all other methods, the accuracy improves: FFNN (RMSE declines from 1.74 to 1.68 and MASE from 0.71 to 0.69); BRNN (RMSE from 1.71 to 1.69, MASE constant at 0.69); and GBM (RMSE from 1.75 to 1.68, MASE from 0.78 to 0.69). Thus we see a slight advantage to using clustered input data rather than SKU-specific data. However, the effect is rather small—except for the GBM approach, where we see a dramatic improvement in predictive performance. It seems that the GBM ensemble of weak prediction models benefits the most

from a larger data set that contains a wide variety of input cases. In contrast, the other methods work best with data that are case specific.

2.5.4 *Results of meta learning: Forecast combinations and method selection*

Table 2.8 reports the predictive performance of our meta-learning models; as before, column rankings are given in parentheses. For the simple combinations of minimum, mode, and mean, we distinguish between two approaches: one in which we select the base forecasting methods across all SKUs; and one where a separate selection is made for each SKU (as described in Section 2.3.2). The latter approach is indicated via the prefix "sku_".

The predictive performance of the best-performing meta-learning methods surpasses that of the best base forecasting methods with time-series decomposition. This result is in line with prior findings that demonstrate the superiority of forecast combinations—that is, given their greater robustness and ability to accommodate different data characteristics. Once again, the ranking of the tested methods differs as a function of the focal data set and the accuracy measure used. According to the RMSE measure, the mean_5 and mean_3 combinations work best, followed by weighted_mean_all and the selection_ranking; the MASE measure assigns the best predictive performance to sku_min_3 and min_5. If we take the average of all rankings across all data sets and performance measures, then both min_5 and selection_ranking are the clear winners. However, one downside of the "minimum" combinations is that, for highly intermittent time-series data with many zero-demand periods, the resulting forecast often predicts zero demand. That generalization holds in the case of our RAF data set, for which the min_3 and min_5 forecasts result in average predictions of (respectively) 0.45 and 0.41 even though the actual average demand is 1.44.

The results show also that the naïve approach, under which selection of the forecasting method is based on how the different models perform when applied to validation data, does not work well. This outcome reflects that time-series characteristics change over time, which explains why prediction methods that worked well in the past may not yield good results in the future.

We observe that the method selection (ranking_selection) works well across all data sets. The standard deviation of the ranking scores for different data sets and different accuracy measures is 3.22, which is smaller than that for other relatively accurate methods (e.g., standard deviation of the min_5 rankings is 5.09). In effect, then, time-series characteristics allow our selection of a

Table 2.8: Predictive performance of meta-learning methods

	Research partner		RAF		Simulation	
	RMSE	MASE	RMSE	MASE	RMSE	MASE
sku_min_3	1.13 (14)	0.72 (4)	3.48 (5)	0.70 (3)	12.13 (15)	0.67 (4)
sku_min_5	1.13 (16)	0.71 (1)	3.37 (2)	0.63 (1)	12.22 (17)	0.67 (2)
sku_mode_3	1.13 (15)	0.75 (14)	3.90 (17)	0.89 (8)	12.11 (12)	0.69 (10)
sku_mode_5	1.12 (13)	0.74 (11)	3.64 (11)	0.90 (9)	12.06 (11)	0.70 (14)
sku_mean_3	1.12 (12)	0.75 (15)	3.85 (16)	0.95 (10)	12.05 (10)	0.70 (12)
sku_mean_5	1.11 (10)	0.74 (12)	3.75 (14)	0.99 (12)	12.03 (7)	0.72 (16)
min_3	1.11 (7)	0.72 (3)	3.37 (3)	0.97 (11)	12.04 (9)	0.67 (3)
min_5	1.11 (9)	0.71 (2)	3.35 (1)	0.68 (2)	12.12 (13)	0.67 (1)
mode_3	1.10 (5)	0.73 (7)	3.61 (9)	1.20 (18)	12.00 (4)	0.68 (7)
mode_5	1.10 (4)	0.74 (10)	3.62 (10)	1.07 (13)	12.00 (5)	0.70 (11)
mean_3	1.10 (3)	0.73 (8)	3.58 (6)	1.16 (17)	12.00 (3)	0.69 (9)
mean_5	1.10 (1)	0.74 (9)	3.60 (7)	1.09 (14)	11.99 (2)	0.71 (15)
weighted_ mean_sku	1.11 (6)	0.75 (13)	3.84 (15)	1.14 (16)	12.13 (14)	0.76 (18)
weighted_ mean_all	1.10 (2)	0.75 (16)	3.68 (13)	1.10 (15)	11.97 (1)	0.76 (17)
naive_ selection	1.14 (17)	0.76 (18)	4.00 (18)	0.89 (7)	12.14 (16)	0.69 (8)
selection_ ranking	1.11 (8)	0.73 (6)	3.46 (4)	0.79 (5)	12.01 (6)	0.70 (13)
selection_ regression	1.11 (11)	0.73 (5)	3.65 (12)	0.75 (4)	12.04 (8)	0.68 (5)
selection_ classification	1.14 (18)	0.75 (17)	3.61 (8)	0.86 (6)	12.35 (18)	0.68 (6)

single forecasting method to be independent of the particular data set at hand. In comparison with other method combination schemes, method selection is also much more efficient because only the selected prediction method needs to be trained. For the other meta-learning methods, *all* base forecasting approaches must be assessed when ranking them in terms of the validation data set—which is prerequisite to selecting the method that should be applied to the

test data set. Depending on the size of the data set and the required frequency of forecast updates, the time savings can easily exceed hours or even days.

With regard to assessing feature importance, if there are trend and seasonality patterns involved then the method selection builds on the trend and seasonality measures but also on the *skewness* of the remainder component. In the absence of seasonality and trends, the most important features of the time series are its mean and standard deviation. We present details of our feature importance assessment and ranking method selection in the Appendix A.2.

2.5.5 *Research partner data: Comparison with current forecasting method*

We use our research partner's data set to compare the accuracy of that online retailer's current forecast with predictions based on the models described in this paper. Although confidentiality concerns preclude our knowing which forecast approach the retailer currently uses, we do have access to selected past demand forecasts that can be compared with the tested methods. Because the SES model without time-series decomposition is the forecast approach whose predictions are most similar to the firm's internally forecasted values, we use this model as a reference against which to compare the tested methods. The SES forecast for the test period has a mean RMSE of 1.29 and a mean MASE of 0.85, both of which are significantly worse than the values derived when we use the ranking method to select an approach (RMSE of 1.11, MASE of 0.73; each value is about 14% less than its counterpart under SES).

2.5.6 *Research partner data: Results of storage cost simulation*

In addition to evaluating forecast accuracy, we run a warehouse simulation for the data set from our research partner to compare—among all the tested forecasting methods—the operation costs that would occur. This simulation covers periods in the test data set as well as the main cost factors: picking and transferring stock. Results from these cost calculations based on the various forecasting models are reported in Table 2.9.

Costs are lowest for the meta learning with learned combination rules. The method selection approach, which yields the best performance across all data sets in terms of our two accuracy measures, has slightly higher costs—although they are lower than the costs of our base forecasting methods. The heuristic approach with ABC clustering results in significantly higher costs. The models' rank ordering does not change much when the shelf-picking area's capacity

Table 2.9: Results of warehouse operation costs simulation for selected prediction methods

Forecasting method	Cost[a]
Meta learning: learned combination per SKU	100.00
Meta learning: learned combination across SKUs	100.01
Meta learning: method selection (ranking)	100.17
ARIMA	102.05
SES	102.80
ABC-analysis	104.80

[a] 100 = cost of the best-performing method

changes from 10% to 5%, 20%, or 30%. That ordering is little affected also when the ratio of picking-area transfer costs to high-bay–warehouse transfer costs declines from 10:1 to 8:1, 4:1, or 1:1. In light of our results from Section 2.5.5, we hypothesize that the actual operation costs are the same as those for the SES method. It follows that warehouse operation costs can be reduced by nearly 3% in the short term under the proposed method selection model.

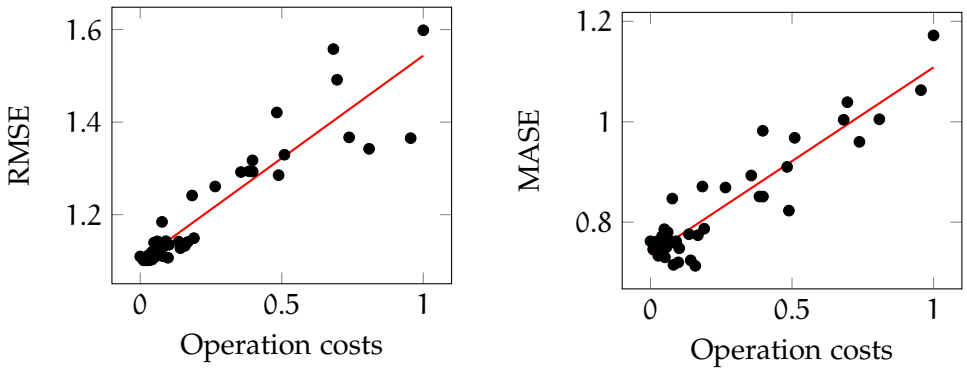

Figure 2.4: Correlation between the forecast accuracy (RMSE, left panel; and MASE, right panel) and the warehouse operation costs (normalized to range between 0 and 1)

The correlation between our RMSE/MASE results and the cost calculation is plotted in Figure 2.4, where we use a red line to mark the trend derived via linear regression. There is a clear positive correlation between our predictive performance results and the cost calculation. Thus we confirm that more accurate forecasts translate into more efficient operations and lower costs.

2.6 CONCLUSION AND FUTURE RESEACH

We find that the method selection approach works as well as the best forecast combination methods. In both cases, predictive performance is superior to using a single base forecasting method. We further observe that the forecast accuracy significantly improves by using more sophisticated machine learning methods instead of simple standard economic methods. The ranking selection approach leads to an RMSE value of 1.11 and MASE of 0.73 for the data set of the research partner whereas the result with one of the best economic models, ARIMA, only reaches an RMSE of 1.29 and MASE of 0.89. Similar improvements are achieved for the two other data sets. We observe that this improvement is mostly achieved by using the meta learning approach as the base machine learning methods reach similar results compared to the standard economic prediction methods. However, machine learning methods are also more complex and require more time to calculate due to the training of the model. Method selection offers the benefit of significantly reducing this calculation time, since only the selected method needs to be trained with historical data. In this way our approach combines high predictive accuracy of a meta learning model with efficient calculation reducing the complexity compared to a standard combination approach. Moreover, we demonstrate that method selection works well across different data sets; such flexibility is not evidenced by most of the base forecasting methods, which work well with one data set but not with another. We also observe that a prediction method's performance changes over time because the time-series characteristics themselves change. When that happens, the method selection approach is able to automatically select an alternative prediction method as needed.

The managerial implications of the findings are threefold: First, it would be a straightforward matter for our research partner to implement the method selection approach, which means that our calculated 3% savings on the overall warehousing operation costs can be realized short-term without any change in the warehouse setup. At the same time, the approach is efficient regarding calculation times and thus demand predictions can be updated with high frequencies. Second, this approach can be applied not only to the warehouse storage decision but also to the decisions made by other departments, such as procurement and marketing to further reduce warehouse operation costs by optimizing purchase volumes or marketing campaign focus. By doing so, the purchase frequencies and inventory levels can be improved on a SKU level.

Lastly, another possible use is in optimizing the warehouse layout when space must be allocated between the picking area and the high-bay warehouse.

The one caveat to adopting any of the tested prediction methods is that they work only if there is sufficient historical demand data available to train the forecasting models. In the case of new products, for which no demand history is available, one must use different demand prediction methods and forecast that product's demand separately. Ferreira et al. (2015) propose such an approach in the field of fashion sample sales.

There remain several related avenues of research worth pursuing. First, one could extend the one-period forecast to multiple periods and thereby consider, for example, the costs of transferring stock over several periods. A longer time horizon would be beneficial also in cases where the forecast is used to plan order volumes or to optimize warehouse layout. Second, our model could be augmented with more features—such as price points, weather data, and behavioral data from the retailer's website—to improve predictive performance. Third, method selection could be expanded to choose not just one of the base prediction models but also between both base models and from combinatorial models.

3

INTERPRETABLE PREDICTION OF PARTIAL DEFECTION: A CASE STUDY IN THE B2B PARCEL LOGISTICS INDUSTRY[1]

3.1 INTRODUCTION

The management of customer relationships has become a topic of increasing interest in the academic and business worlds (Lu et al. 2014). The objective is to create long-term profitable customer relationships that yield significant value over time (Frow and Payne 2009). Companies want to retain their existing customers because it is much more expensive to attract new ones (Huang and Kechadi 2013). Studies have estimated the costs of attracting new customers to be 5–6 times higher than the costs of retaining existing ones (Bhattacharya 1998, Nie et al. 2011). Dingli et al. (2017) state that the most common reasons for customers churn are due to competitive offerings with cheaper prices, negative word of mouth, better service offered by competitors or re-location of customers. Predicting defection is therefore meant to detect potential churners early on and to plan retention actions aimed at customers who are about to switch to another provider or supplier. Therefore, often historic buying patterns are analyzed to predict behavior in the future (Dingli et al. 2017). The prediction model usually provides a defection probability for each customer, which allows us to rank customers in terms of their churn probability. One can then decide how many and which of these customers to target in retention campaigns. Van den Poel and Lariviere (2004) state that even small improvements in retention may lead to significant increases in future revenue. In addition to the high costs of acquiring new customers, there are other advantages to retaining current customers. Long-term customers provide higher revenues, require lower costs to be served, and can spread positive word of mouth (Reichheld 1996). Yet owing to different levels of profitability within the customer base, it is more important to retain some customers than others (Lemmens and Gupta 2017). Therefore, profitability should be considered when deciding which customers to target with retention actions (Bahnsen et al. 2015).

[1]The following chapter is based on Faber and Spinler (2019b), unpublished working paper.

The literature offers various definitions of *defection*. A common distinction is between "defection" (churn) and "partial defection". In the former, the customer has been unequivocally lost (as when, e.g., a contract is terminated); in the latter, the customer becomes inactive or switches some of its demand to another provider (Buckinx and Van den Poel 2005).

In many industries, companies must deal with high churn rates. In telecommunications, churn rates are reported reaching from 45% to 70% in some years (Mattison 2001). The implication is that more than 3.75% of all customers cancel their contracts every month.

Our study makes three main research contributions in the field of predicting defection. First, we widen the focus in business-to-business (B2B) contractual settings so as to encompass not only complete defection but also partial defection. Thus we demonstrate how companies that use contracts without fixed volumes can, early on, identify (a) defectors who slowly switch to a competitor over time and (b) which customers should be targeted with retention actions. To maximize the firm's future profit after such retention actions, we must learn (i) which customers are most likely to exhibit partial defection and (ii) how much revenue is at risk in each case. The latter question is key because there is a significant difference between a customer that reduces its demand by (say) half and a customer that abandons the firm completely; hence the absolute level of revenue prior to defection is a chief concern. We therefore test the usefulness of combining defection classification methods (e.g., logistic regression, neural networks, gradient boosting) with regression methods that estimate future levels of demand.

Second, whereas the extant literature invariably defines partial defection as a demand decrease of more than 50%, we investigate how the predictive performance changes with different thresholds or lead times for predicting partial defection. We present a sensitivity analysis for two variations. (i) We vary the threshold that determines which customers are defined as partial defectors. The lower the threshold, the clearer the difference between partial defectors and loyal customers who may simply have fluctuating demand over time. Yet a lower threshold translates into a smaller number of classified partial defectors, which tends to reduce predictive power. (ii) We modify the lead time between the date of prediction and the prediction period. The longer the lead time, the more time there is for retention actions but the greater the prediction error. Given the results of both sensitivity analyses, managers are equipped to select a combination of threshold and lead time that results in high predictive performance while accommodating business requirements (e.g., the firm's capacity to engage in retention actions).

Our third research contribution is to develop an approach that renders interpretable even the more complex models of predicting defection. Current research in this field seeks prediction methods that are interpretable. In contrast, we develop a data permutation approach that can be applied in the context of any data-mining model and that leads to high predictive performance. One can use this approach to identify case-specific reasons why, irrespective of the model, a customer is likely to exhibit partial defection.

The rest of our paper proceeds as follows. Section 3.2 reviews existing research and introduces the prediction methods we consider: clustering, ensembling, profit-based predictions, and interpretable models. In Sections 3.3 and 3.4 we explain the chosen model approach, our case study, and the data set. Results and managerial implications are then presented in Sections 3.5 and 3.6. We conclude in Section 3.7 with a summary and suggestions for future research.

3.2 LITERATURE REVIEW

In this section we introduce data-mining models and survival probability models for defection prediction, including combinatorial methods that use clustering and ensembling. We also discuss models that focus on profit optimization and interpretability instead of solely on predictive performance.

3.2.1 *Defection defined*

Most of the research on predicting defection is in one of two streams: optimizing predictive performance or emphasizing model interpretability (De Caigny et al. 2018). Scholars typically focus on the accuracy needed to maximize prediction performance (Verbeke et al. 2012), and some researchers use financial metrics to plan retention actions in a way that maximizes profit (Jahromi et al. 2014). Often, the best results are achieved with rather complex models that are difficult to understand and interpret (De Bock and Van den Poel 2012). Hence other researchers focus on a model's interpretability toward the end of understanding and interpreting its predictions. The insights gained from that research can help identify the drivers of specific customer behavior, after which interventions can be planned in accordance with those factors (Gustafsson et al. 2005). The combination of these research streams addresses the trade-off between accuracy and interpretability (De Caigny et al. 2018). Various learning models have been proposed for predicting defection. Among the most common

of these are logistic regression (Neslin et al. 2006), decision trees (Wei and Chiu 2002), neural networks (Au et al. 2003b), random forests (Larivière and Van den Poel 2005), and support vector machines (Coussement and Van den Poel 2008) as well as survival probability models (Fader and Hardie 2007). In addition, some studies combine two or more these models with clustering (Huang and Kechadi 2013) or in an ensembling approach (Lemmens and Croux 2006).

Within churn research, one can further distinguish between contractual and non-contractual settings. In the first case, churn can easily be detected by the termination of a contract whereas this is not possible in the latter case. In latter situations, the company has to derive the status of the customer based on observable behavior. We focus on the first situation as our research partner has contractual relationships to the customers who consume the offered services. Another difference lies in the focus of the analysis: The prediction can either focus on cohorts of similar customers or on individual customer level.

The existing literature on churn prediction focuses primarily on business-to-consumer (B2C) interactions and consists of examples and case studies from telecommunications (Lemmens and Croux 2006, Huang and Kechadi 2013, Lu et al. 2014), the financial industry (Kumar and Ravi 2008), and online subscriptions for gaming or music (Runge et al. 2014). In contrast, defection in B2B settings remains a niche topic; only a few researchers (Rauyruen and Miller 2007, Jahromi et al. 2014, Chen et al. 2015a) have examined it. Jahromi et al. call for more empirical research, especially in the B2B field. They argue that there is less data available for B2B than for B2C settings and that insufficient use has been made of the data that are available. There are two major differences between B2C and B2B markets. First, the customers in B2B settings are usually fewer but larger, and they make larger and more regular purchases and therefore tend to be more valuable (Rauyruen and Miller 2007). Because of these large customers in B2B settings, each lost customer accounts for a significant share of the focal firm's total revenue. It follows that churn prevention is essential for maintaining revenue and profit. Second, the cost of switching to another provider are often low in B2B markets and so customers can easily move to a competitor. The rise of information technology enabled better customer access to information, which made it easier and cheaper than before to switch to a competitor (Wiersema 2013). Chen et al. (2015a) documents the strong competition in these markets and also points out that competitive advantages in the logistics industry (e.g., cost, speed of delivery) can easily be imitated by other market players. For these reasons, there is a fairly high risk of losing customers.

Some authors define defection as an explicit cancellation of contracts (a.k.a. churn), while others use a broader definition that also includes customers who are inactive or have partially defected. The latter definition is common in non-contractual settings wherein one can observe user activity only over time, which makes it impossible to identify complete defection. In cases of partial defection, customers purchase also from another provider but do not shift all of their demand to that provider. According to Buckinx and Van den Poel (2005), however, even a short-term decline in revenue can lead to complete defection in the long run—as occurs when customers switch to another provider gradually over time. Buckinx and Van den Poel define partial defection in terms of loyal and unloyal customers, groupings that are based on those customers' behavioral attributes. The *loyal* group consists of customers characterized by an above-average purchase frequency and a below-average coefficient for variation in the interpurchase time. Ahn et al. (2006) study partial defection in the contractual setting of the telecommunications industry. These authors define partial defection as either nonuse or suspension of a contract. Thus they define three different customer states: active, partially defected, and churned.

We study the particular case of partial defection in a contractual setting. Even though a contract is in place, a customer can gradually shift its demand to other service providers (i.e., it can partially defect). The reason is that shipping volumes are not fixed and so can easily be adjusted. Given that the parcel shipping services operate continuously, we use demand changes to classify defection. More specifically: a customer whose shipments decline by a prespecified threshold percentage is classified as a partial defector. We focus the analysis on individual customers given the availability of data on an individual level.

3.2.2 *Models used to predict defection*

García et al. (2017) perform a large literature review to review and anlyze different methods to predict churn. They state that the choice of method is very dependent on the problem at hand but argue that both - a broad comparison of methods as well as method combinations should be considered. The authors show that standard methods such as regression are still widely used but machine-learning methods are on the rise. Besides the statistical method, logistic regression, the two most common types of defection prediction methods are *data-mining* models and *survival probability* models. The former amounts to finding patterns in large data sets. The latter method combines (a) a simple probability distribution (e.g., Poisson, binomial, exponential) that characterizes

past observed behavior (e.g., purchase history) and (b) a cross-sectional heterogeneity to account for variation in customer characteristics across the customer base. Among the methods briefly described next, decision tree, random forest, neural networks and support vector machines are data-mining models and shifted-beta-geometric distribution is a survival probability model.

Statistical models are widely used with **logistic regression** (LogR) being the most frequently adopted approach in the literature because it combines accuracy, interpretability, and rapid calculation (Lu et al. 2014). Although the accuracy is sometimes less than that delivered by more complex models, LogR is still preferred because of its easy and transparent application (Runge et al. 2014). Nie et al. (2011) and Coussement and Van den Poel (2008) find that logistic regression achieves higher predictive performance than do decision tree models. **Decision tree** (DT) models are another widely used method whose results are easy to calculate and interpret. Hadden et al. (2006) and Au et al. (2003a) both use CART, whereas Chen et al. (2015a) use C4.5. However, all these studies find that DT models yield better results than those obtained via neural networks or logistic regression. That said, Hung et al. (2006) report that neural networks perform better than does the C5.0 DT.

Out of the machine-learning models, **Random forest** (RF) is often used to predict partial defection (Buckinx and Van den Poel 2005). Larivière and Van den Poel (2005) find that RF higher predictive performance than does LogR, and Coussement and Van den Poel (2008) report that RF outperforms both logistic regression and support vector machines. The analysis of Xie et al. (2009) shows that RF performs better than decision trees, neural networks, or support vector machines. However, Buckinx and Van den Poel (2005) find no significant differences in the results derived via LogR or neural networks from those based on RF. **Neural networks** is also frequently used to predict defection. According to Au et al. (2003b) and Hung et al. (2006), NN perform better than do DT; Vafeiadis et al. (2015) find that NN perform no worse than DT and that both perform better than either support vector machines or naïve Bayes. In contrast, Tsai and Chen (2010) argue that the DT model works better than the tested NN model. **Support vector machines** (SVM) are also popular for predicting defection, although they usually require longer calculation times (Lessmann and Voß 2009). Zhao et al. (2005) and Xia and Jin (2008) both report that SVM performs better than LogR, DT, NN, or naïve Bayes. Coussement and Van den Poel (2008) remark that only an optimal parameter selection for SVM leads to it performing better than LogR. Vafeiadis et al. (2015) show that a DT model and an NN model both work better than the tested SVM approach.

An alternative to the statistical and machine-learning methods, **probability models** are based on probability distributions used to predict future customer behavior. In the case of defection, this distribution is a survival function that—for any moment in time—returns a probability that a customer will remain active or will defect (where that likelihood depends, in part, on the focal customer's contract length). Fader and Hardie (2007) develop a model for contractual settings that builds on a "shifted beta geometric" distribution. Tamaddoni et al. (2016) compare several data-mining models with probability models; these authors find that probability models outperform data-mining models only when the sample size is extremely small.

In summary, one can see that the results and rankings of the different prediction methods differ from one analysis to the next; there are even cases of reported results that directly contradict each other. Thus there is no "silver bullet" and so one must find the best method for each particular data set (Verbeke et al. 2012). A disadvantage shared by all machine learning models (RF, NN, SVM) is the lack of interpretability. All such models are viewed as "black boxes" that enable analysis of overall feature importance but can generate no specific insights regarding a single prediction.

3.2.3 *The use of clustering to predict defection*

Huang and Kechadi (2013) use unsupervised learning to cluster their input data and then use a supervised method to predict defection. They find that a hybrid model (using k-means clustering and rule induction prediction) is more accurate than six other classification techniques (including LogR, DT, and SVM). According to Huang and Kechadi, a hybrid approach allows the firm to compile training sets (for its machine learning algorithms) that include customers with similar behavior patterns and hence that result in greater predictive accuracy. De Caigny et al. (2018) develop a different hybrid approach that combines DT-based clustering and LogR. These authors report that the combined approach works better than either LogR or DT and is no worse than more advanced methods such as RF. Another hybrid approach is tested by Fathian et al. (2016), who use self-organizing maps (SOM) to cluster their data and apply four different learning models: DT, NN, SVM, and k-nearest neighbor. These authors use principal component analysis (PCA) to reduce the feature space and also use bagging and boosting (see Section 3.2.4). They find that a combination of SOM, PCA, and heterogeneous boosting performs best. Lu et al. (2014) describe a

clustering approach that combines boosting with LogR to predict defection for any given cluster.

3.2.4 *The use of ensembling to predict defection*

Researchers have often combined models in their efforts to improve predictive performance. Popular choices of such combinations include bagging, boosting, and stacking (Abbasimehr et al. 2014).

Bagging starts with randomly sampled sets of the training data with replacement; as a result, a new training set is created for each classifier. The classifiers are then trained individually on their respective training sets. Different classifiers are typically chosen for each of those training sets. The final prediction is made by averaging the single predictions of all of the trained classifiers.

Boosting is, in contrast, an iterative process. Prior to each iteration, the data are reweighted to reflect the predictive accuracy observed in the previous run. In addition, the classifiers are combined by using weights in proportion to performance. The most commonly used implementations of boosting are AdaBoost (Freund et al. 1996), LogitBoost (Friedman et al. 2000), and gradient boosting (Friedman 2001).

Stacking is a two-step approach. First, base classifiers are trained with the training data; second, another data set is used to combine the predictions of the base classifiers. In this way, the output of all the different base models is used as input to a new classifier that combines the outputs into a final classification.

In the analysis of Abbasimehr et al. (2014), boosting achieves better results than bagging, stacking, or voting with four base classifiers: DT, NN, SVM, and *reduced incremental pruning to produce error reduction*. Vafeiadis et al. (2015) test and compare several different machine learning techniques (NN, SVM, DT, naïve Bayes) as well as LogR for defection prediction and find that DT and NN work best. In a second step, they apply AdaBoost boosting and find significant improvements for all tested models. Lemmens and Croux (2006) find that both bagging and stochastic gradient boosting improve on the results of DT; they also emphasize that the choice of which method to use depends on the given data set. Ge et al. (2017) find that a gradient boosting approach is superior to LogR and RF in predicting defection for a "software as a service" company. In a recent kaggle challenge (WSDM Cup 2018) that addressed defection prediction for a leading music streaming service in Asia, Gregory (2018) used a gradient boosting approach to win the competition. However, Chen et al. (2012) argue that more complex SVM models work better than boosting.

3.2.5 *Combined defection prediction and profit optimization*

The firm's goal of predicting defection through classification accuracy can be misaligned with its profit-maximizing ambitions (Bahnsen et al. 2015). The reason is that misclassification costs are strongly affected by the profitability of customers (Glady et al. 2009). Misclassification costs occur in two cases: when defectors are wrongly classified as non-defectors and when non-defectors are wrongly classified as defectors. In the first case, the costs are equal to the loss in profit due to defected customers. In the latter, costs are incurred for undertaking retention actions and for the retention offers (e.g., discounts) themselves. The profit generated by a retention campaign depends, in turn, on (a) the likelihood of customers accepting the offer and (b) the incentive-driven change in customer lifetime value (Lemmens and Gupta 2017). The optimal target size of a retention campaign is often determined by the trade-off between retention costs and the potential loss through defection (Lemmens and Gupta 2017). Instead of targeting customers with the highest predicted defection probability, the firm should target those for whom retention actions are expected to yield the highest return. That expected return is a function of three factors: the revenue from—or customer lifetime value of—the focal customer, the cost of the retention offer, and the offer's acceptance rate (Verbraken et al. 2013). Bahnsen et al. (2015) propose the following profit-based measure:

$$\text{Expected profit} = \text{TP}(\gamma(\text{CLV} - C_o - C_a) + (1-\gamma)C_a) + \text{FP}(-C_o - C_a) \quad (3.1)$$

The expected profit depends on two different cases, the correct prediction of a defector (TP = true positive) or the false prediction (FP = false positive). In both cases, there is a retention offer cost (C_o) and an administrative cost of contacting the customer (C_a). If the customer is an actual defector (i.e., the TP case) then that customer accepts the offer with probability γ. In this case, the firm's profit is the customer lifetime value (CLV) minus the costs. A customer that is not a defector (i.e., the FP case) will always accept the offer and thus only costs occur. So instead of maximizing profits, the firm can change the sign of the formula and minimize costs (Bahnsen et al. 2015). One can also add the cases of false negative predictions (where the cost is CLV) and true negative predictions (in which case there are no costs).

Larivière and Van den Poel (2005) test different defection prediction models and investigate the outcomes with regard to both profitability and classification accuracy. The effect on profitability is tested by using, as dependent variables, profit *evolution* (the profitability change compared to the last observation) and profit *drop* (a binary value that indicates whether a customer has become less

profitable since the previous observation). However, these authors do not optimize profit; they only assess accuracy in the context of forecasting future profit development. Bahnsen et al. (2015) develop a cost-sensitive modeling approach that accounts for CLV and that differs in the cases of correct versus incorrect classification. They use DT, LogR, and RF models—with and without cost-proportionate sampling–and report that their cost-sensitive approach increases profitability by as much as 26.4%. Lemmens and Gupta (2017) use a profit-based loss function to take profitability into account and to make better predictions for high-profit customers than for others. These authors compare the profit-based loss model with standard misclassification-based loss models that reflect the profit increase calculated using (a) the retention action response probability and (b) the change in CLV due to the incentive. Bahnsen et al. show that their proposed approach yields an average increased profit of 62%. Verbeke et al. (2012) develop a novel profit-based performance criterion by which CLV is used to measure the potential maximum profit that can be achieved via a retention campaign. They maximize the potential profit across all customers in order to select the model with the highest incremental profit and to find the optimal share of customers to include in the campaign.

3.2.6 *Interpretable models*

Most of the research on predicting defection focuses on predictive accuracy. However, there are many cases in which defection predictions must also be interpretable and understandable—so that such predictions can be more easily trusted (Freitas 2014). Especially when the prediction runs counter to expectations, it is crucial to understand the model or at the least the reason for the prediction. Freitas (2014) highlights the importance of comprehensible data-mining models in the medical and military domains, where one must disentangle cause–effect relationships. Yet comprehensibility is vital also for predictions of defection because the reason for defecting might well differ among customers, in which case customer-specific retention actions should likewise differ to ensure a high success rate. Acting in response to a specific prediction requires that one understands the predictive model and also the customer's reason for defecting. Sales agents can then use this information to target each customer (or group of customers) and to devise measures for retaining them.

Freitas (2014) compares models that are widely viewed as delivering comprehensible results—including decision trees, nearest neighbors, and Bayesian network classifiers—and discuss their respective advantages and disadvantages.

Verbeke et al. (2011) use two novel data-mining models, Ant-Miner+ and an *active learning-based approach* (ALBA), to combine accuracy with interpretability. In their research, the ALBA approach that incorporates a nonlinear SVM works best even when compared with traditional rule induction techniques (e.g., C4.5 DT).

Besides these directly interpretable models, one can also aim to extract comprehensible models or simple rules from more complex models. Mashayekhi and Gras (2015) introduce a method for extracting rules from RF models via a hill-climbing algorithm that sharply reduces the number of rules. This method improves comprehensibility without compromising the level of accuracy. Ribeiro et al. (2016) present an approach to explaining predictions of any classifier through *local interpretable model-agnostic explanations* (LIME). This approach proceeds as follows. First, the observation to be explained is permuted n times. Second, the selected classifier is used to predict the outcome of the permuted observations. Then the (Euclidean) distance of all permuted observations to the actual data is calculated and transformed into a similarity score. Next, the m features that best explain the selected classifier from the outcome of the permuted observations are selected and used to train a simple model with the permuted data; these features are weighted to reflect the previously calculated similarity score. Finally, one can extract the feature weights from the simple model to explain the selected classifier.

3.2.7 *Data balancing*

Defection is usually a rare event. Hence the data are often highly unbalanced because only a small share of the firm's customers are defectors (Lemmens and Croux 2006). This imbalance can lead to poor predictive performance because it is more difficult to anticipate events that are relatively rare. In describing several different approaches to balancing data, He and Garcia (2009) identify three state-of-the-art solutions for unbalanced learning: sampling methods, cost-sensitive methods, and kernel-based or active learning methods.

Sampling methods aim to achieve a balanced data set. One option is upsampling, in which case a random sample from the minority class is replicated. In contrast, downsampling removes random observations from the majority class. The *synthetic minority oversampling technique* (SMOTE) is an alternative that combines downsampling of the majority class with creation of new minority instances by interpolation based on feature space similarities between existing minority examples (Chawla et al. 2002).

Cost-sensitive learning builds on a cost matrix that applies different misclassification costs to different observations. Through weighting, a heavier cost is imposed when errors are made in the minority class.

Active learning methods are the least common approach. They involve kernel modifications and/or active learning methods for support vector machines.

3.3 APPROACH TO MODELING

3.3.1 *Classification models and data balancing*

We use the most common classification models for predicting defection, as described in Section 3.2.2: LogR, DT, SVM with radial kernel, RF, and NN. The DT model used for our analysis is based on the well-known C5.0 approach. We also test a gradient boosting models (GBM) approach (Ridgeway 2007). In addition to these data-mining models, we use a survival probability model to characterize the observed behavior of customers, which in turn enables predictions about the expected duration of a customer relationship. The contractual nature of our setting explains our decision to use the shifted beta geometric (sBG) probability model (Fader and Hardie 2009). Table 3.1 briefly summarizes these prediction models. For the machine learning methods, one must select hyperparameters

Table 3.1: Prediction methods used in this analysis

Abbreviation	Method	Source
LogR	Logistic regression	2013
RF	Random forest	Breiman (2001)
NN	Neural networks	Rumelhart et al. (1988)
GBM	Stochastic gradient boosting model	Ridgeway (2007)
C5.0 DT	C5.0 decision tree	Quinlan (1986)
SVM	Support vector machines	Scholkopf and Smola (2001)
sBG	Shifted beta geometric probability model	Weinberg and Gladen (1986)

that "tune" the learning approach so as to minimize the generalization error. For example, an SVM with a radial basis function (RBF) kernel requires that both the parameter of the soft-margin cost function (c) and the free parame-

ter of the Gaussian RBF (γ) be predefined. One can use different approaches (e.g., random selection, grid search) to select the best hyperparameters for a particular data set. Since the computational power required for grid search is not too large in our case and since the task can be split into parallel subtasks, we adopt a five-fold cross-validation approach with a grid search to select the hyperparameters that work best (Hsu et al. 2003). The tested hyperparameter combinations are listed in Table 3.2. Given the skewed distribution of defectors

Table 3.2: Hyperparameters considered in this analysis

Classifier	Hyperparameter	Candidate settings
RF	Randomly selected predictors	2, $\sqrt{\#features}/2$, $\sqrt{\#features}$
NN	Weight decay	0.001, 0.01, 0.1
	Hidden units	1, 2, 3, 5, 10
GBM	Max. tree depth	1, 2, 3
	Boosting iterations	50, 100, 200, 500
	Shrinkage	0.01, 0.1
	Min. terminal node size	10, 25
C5.0	Trials	1, 10, 20

versus non-defectors in the data set, we test four different data-balancing methods (upsampling, downsampling, SMOTE, and weighting; see Section 3.2.7) for all models *except* for the one based on probability distribution.

3.3.2 *Combined unsupervised and supervised classification*

In addition to the classification approach using the models just described, we use a combined clustering and classification approach. For this purpose, we combine an unsupervised model to cluster the customers with a supervised model to predict, within each cluster, each customer's likelihood of defecting. We use the well-known k-means clustering approach to split the data into clusters before we train the models individually on each cluster (cf. Hartigan and Wong 1979). The number of clusters is determined by the "elbow" method (Thorndike 1953), which is based on minimizing the total within-cluster sum of squares.

3.3.3 *Ensembling of different classifiers*

Besides the GBM boosting approach (one of our base classification models), we test two other ensembling methods that combine different classifiers. First, we use heterogeneous bagging. We check the accuracy on the validation data set to select the models with the highest accuracy (based on AUC-ROC; see Section 3.3.8), after which we combine the predictions of the selected models to estimate the defection probabilities for the test data set. We test different numbers of models to combine (all of them or the top 3, 5, or 10). Second, we use a stacking approach. The predictions of the various classifiers on the validation data set are used as input for the final model training. We test and compare all described classifiers for the stacking model.

3.3.4 *Classification sensitivity analysis: Partial defection thresholds and prediction lead times*

We use sensitivity analysis to assess how different prediction conditions affect predictive performance.

First, we test different partial defection thresholds. The dependent variable is the share of the combined demand over four months compared to the same months during the previous year; if demand decreases then this value is between 0 and 1. We use a threshold value to classify a customer as a partial defector or not. If the calculated share is below the threshold, then the customer is classified as a partial defector. Thus the higher the partial defection threshold, the greater number of customers classified as partial defectors (and vice versa). Table 3.3 reports the share of defectors for each tested threshold. In addition to

Table 3.3: Share of defectors in the data set for five partial defection thresholds

Threshold	Share of defectors (%)
0.25	0.02
0.50	0.07
0.60	0.11
0.70	0.17
0.80	0.25

the fixed threshold across all customers, we also test a signal-to-noise approach.

In this test, we use different threshold values for each customer depending on the variation in their previous demand: the lower the demand variation, the higher we set the threshold value. In this way, we aim to distinguish between a temporary and a structural decline in demand for each customer. Some B2B customers exhibit fluctuating demand that is due to their business models (e.g., promotion or price effects), and these customers should not be classified as defectors. Others have more stable demand, and it is with respect to these customers that even a small reduction in demand is indicative of an imminent switch to a rival; hence they should be viewed as potential defectors. We use the coefficient for variation to cluster the customers and then apply different threshold levels for each cluster. The cases of two and four clusters are both tested, and we also checked the effects of five different threshold variations (viz., 0.8, 0.7, 0.6, 0.5, and 0.25).

Second, we changed the lead time between the feature observation (i.e., time of prediction) and the prediction period. The greater this time span, the earlier one can identify potential defectors. However, a longer time span does reduce the method's predictive performance.

3.3.5 *Regression models*

We use regression as an alternative to the classification approach to address partial defection. Regression models are used to predict future demand and hence to identify possible reductions in revenue and profit. Customers for which predicted demand is below the partial defection threshold are considered to be defectors. If the aim is to maximize profits, then the firm must know how *much* revenue and profit is at risk for each customer. The potential profit loss resulting from a 50% reduction in a large customer's demand could exceed even the complete loss of a smaller customer. Hence we compare different regression methods to predict each customer's future demand. The same methods and data are used here as for the classification approach but with three differences: we use linear regression (LinR) instead of logistic regression as well as CART regression trees instead of C5.0 classification trees; and we do not use a probability model.

3.3.6 *Profitability analysis*

Our research partner is naturally interested in the financial effects of better partial defection predictions. How do better forecasts translate into savings or

additional profit? In contrast to most published accounts, our research partner does not offer discounts to retain customers. Instead, sales agents seek to resolve customers' complaints and to offer them improved service. Hence each retention call or visit comes with a fixed cost. Discussions with our research partner led us to set this cost at €100. No published data are available for acceptance rate, thus we follow the literature (see e.g. Lemmens and Gupta 2017,Neslin et al. 2006) and use 30% for the retention offer acceptance rate. We discussed this approach with our research partner who confirmed this rate based on internal analysis and experience for our case. The increased profit resulting from successful retention actions is equal to the prevented profit *reduction*, which in turn depends on the chosen partial defection threshold.

The goal is to engage proactively with customers whose loss would reduce profits the most and thus whose retention would yield the greatest uplift. There are three options when prioritizing customers so that retention strategies maximize the firm's profit. The first option is to multiply the calculated partial defection probability of the tested classifiers (classification approach) by the current profit of each customer. Thus customers that are both highly profitable and likely to defect are listed first. The disadvantage of this option is that it fails to account for the specifics of partial defection; so whether a customer is about to reduce its demand by 50% or 90%, a score based solely on the defection probability and the profit will remain the same. For that reason, the second option subtracts the calculated profit forecast (regression approach) from the current profit to obtain the expected decline in profit. Here customers are sorted—from high to low—according to that expected decline (in this case, the extent of the demand decrease is considered).

The third option is to combine the classification result with the expected profit decline using regression. We distinguish between two subcases. "Option 3(a)" consists of multiplying the calculated defection probability by the calculated expected profit decline and then sorting the customers (from high to low) according to the results. Under "option 3(b)", all customers are ranked independently based on both the classification result and the regression result; then the customers are sorted by the average of those two rankings.

For all prioritization approaches, we test each classification and regression model as well as the combinations described in Sections 3.3.1 and 3.3.5.

In addition, we test another approach that could further improve forecast accuracy and thus enhance the profit uplift. More specifically, we use ABC clustering to analyze separately the customers with high and low profits. We assign the 20% of customers with the highest profits to class A, the next 30% to class B, and the remaining customers (with the lowest profits) to class C.

Our focus is then to optimize predictions for class A. Toward that end, we test two different approaches. First, we use the ABC clusters much as we did the k-means clusters described in Section 3.3.2. We train the prediction models independently for each cluster. In the second approach, we assign weights to the customers based on their respective class assignments (this method is similar to weighted data balancing). We test two different class weights. Customers in classes A, B, and C are first assigned weights of (respectively) 3, 2, and 1; then we increase the relative significance of classes A and B by changing their weights to 10 and 3, respectively.

3.3.7 *Interpretability*

Prediction accuracy is more helpful, of course, when the classification results can be easily interpreted. There are two reasons why our research partner seeks to identify the features most important to customers that are predicted defectors. First, its sales agents need to know *why* the focal customer was classified as a partial defector. Second, that information is used by sales agents when making retention calls. Agents adapt their approach to the retention discussion so that it reflects the reasons underlying the customer's predicted defection.

The easiest way to derive a customer-specific ranking of features is to use directly interpretable models such as LogR or DT. We therefore start by comparing the predictive performance of such models with more complex ones. The choice of which model to use is normally based on the trade-off between accuracy and interpretability. In a second step, we aim to make complex models more interpretable and thereby to combine high accuracy with transparency. For that purpose we first follow the LIME approach (see Section 3.2.6), which is based on fitting linear models to permutations of the original training set. The output is a list of the features most responsible for a given observation's classification. To evaluate the LIME approach, we use it in combination with the LogR model and compare the ranking of the extracted features with the ones obtained directly from LogR. A high correlation between these two feature rankings indicates a good explanatory model.

We also test another approach that builds on LIME but reduces the method to the permutation part (i.e., without fitting another model to the permuted data). For each observation, we resample each feature value using all other values of the same feature in the data set. We then calculate the distance between the new defection probability to the actual probability and sort the features according to those distance values. Once again, we evaluate this approach by combining it

with the LogR model and then comparing the ranking of the extracted features to the ranking of features that are derived directly from logistic regression.

3.3.8 *Accuracy measures*

We first introduce the accuracy measures used for classification methods and then describe those used for regression methods. Finally, we explain our approach to evaluating the financial uplift that results from using any of the classification or regression methods.

The performance measure of classification models is based on a so-called confusion matrix, shown in Figure 3.1, that contains the number of correctly and incorrectly predicted cases. In the case of a *true positive* (TP), an actual defector is correctly predicted as a defector; similarly, *true negative* (TN) refers to a correct classification of a non-defector. The misclassification of a defector as a non-defector constitutes a *false negative* (FN), and the false prediction of a non-defector as a defector is a *false positive* (FP). We evaluate the performance

		Predicted	
		Positive	**Negative**
Actual	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 3.1: Confusion matrix for binary classification

of the classification models using the *top decile lift* (TDL), the *area under the curve of the receiver operating characteristic* (AUC-ROC), and the *area under the curve of the precision recall characteristic* (AUC-PRC).

The **top decile lift** factor compares the proportion of defectors in the top customer decile (according the predicted partial defection probability) with the corresponding proportion in the full data set. A score higher than 1 indicates that there is a higher density of defectors in the top decile than in the full data set. This measure helps the firm's sales agents decide which customers to target in a retention campaign (Neslin et al. 2006), since the share of customers to be contacted is usually determined beforehand. For this reason, the TDL is considered to be especially helpful in planning retention actions.

The **receiver operating characteristic** curve is a two-dimensional representation of how the true positive rate (3.2) is related to the false positive rate (3.3) for various cut-off values (Hanley and McNeil 1982):

$$\text{True positive rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \tag{3.2}$$

$$\text{False positive rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{3.3}$$

The **precision recall characteristic** curve represents precision, as defined by Equation 3.4, in terms of recall, as defined in Equation 3.5—again for different cut-off values. This curve is often used to evaluate the predictive performance of models tested on unbalanced data sets (He and Garcia 2009).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}; \tag{3.4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3.5}$$

Example ROC and PRC curves are shown in Figure 3.2. The AUC is a common score-based measure that distills the curve's information into a single value (Krzanowski and Hand 2009). A random classifier leads to a diagonal ROC curve, which yields a value of 0.5 for AUC-ROC; whereas the AUC-PRC's value resulting from a random classifier is approximately the class ratio (Keilwagen et al. 2014).

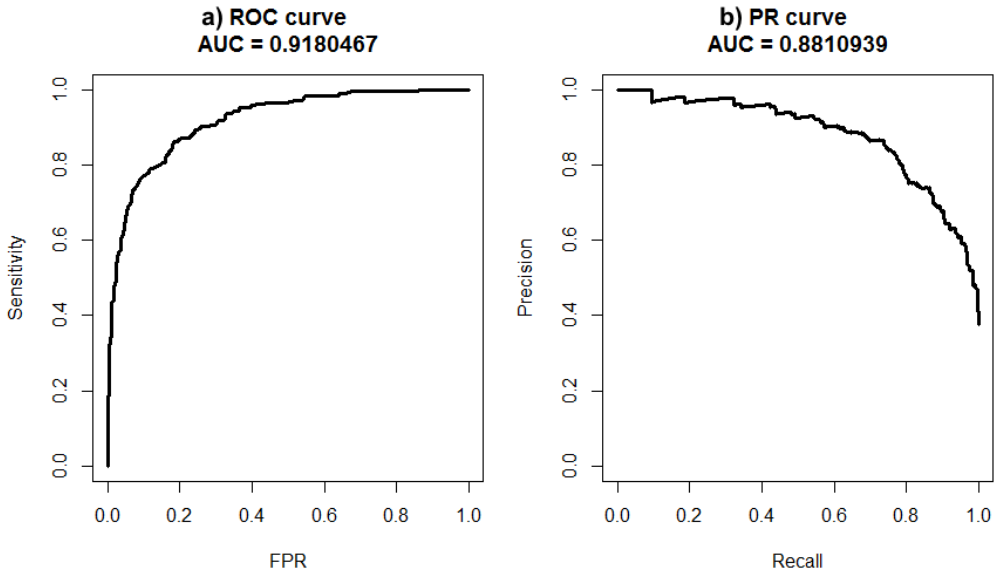

Figure 3.2: ROC curve (left panel) and PRC curve (right panel)

When assessing the regression models, our metrics are the *mean absolute error* (MAE) and the *root mean-squared error* (RMSE) measures. These measures are formally defined in Equations 3.6 and 3.7, respectively:

$$\begin{aligned} y_t &= \text{Actual demand,} \\ \hat{y}_t &= \text{Predicted demand;} \end{aligned}$$

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T} |y_t - \hat{y}_t|, \tag{3.6}$$

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T} (y_t - \hat{y}_t)^2}. \tag{3.7}$$

To estimate the expected profit, we adapt the retention campaign contribution formula of Neslin et al. (2006). We simplify matters by using the actual profit decline with a short-term focus—that is, instead of undertaking a more complicated, long-term CLV calculation. Also, we take into consideration the fact that the research partner does not offer any discounts to retain customers but focuses on targeted calls to the respective customers by the sales agents which comes with a fix cost. We change the formula so that it is no longer necessary to define up front the fraction of customers to be targeted by the retention campaign. Instead, we adopt an approach similar to that used for the TDL. Thus we first sort the customers, as described in Section 3.3.6, by using: (a) the current profit multiplied by the defection probability (for the tested classification models); (b) the predicted profit decline (for the tested regression methods); or (c) a combination of both classification and regression predictions. Next we use (3.8) to calculate the profit P_k for each decile k of customers:

N = number of customers in the decile,

α = success rate (= share of customers that remain due to the retention offering),

c = retention costs (= the cost of contacting a customer), and

d = actual profit decline (aggregated profit in the target months compared to the same months in the year before) *without* retention action.

$$P_k = \sum_{i=1}^{N} \alpha d_i - cN; \tag{3.8}$$

We then report profits for the top decile, P_1, in the *top decile profit* (TDP) and create a *profit index* to cover the results of all deciles. We adapt the index approach of Ling and Li (1998) for use in the context of a defection model's clas-

sification performance. So letting P_k denote the profits derived from decile k, we write

$$\text{Profit index} = 1P_1 + 0.9P_2 + 0.8P_3 + \cdots + 0.1P_{10}. \tag{3.9}$$

Table 3.4 summarizes the accuracy measures that we use.

Table 3.4: Overview of measures

Purpose	Abbreviation	Explanation
Accuracy measures	TDL	Top decile lift
(classification)	ROC	Receiver operating characteristic
	PRC	Precision recall characteristic
Accuracy measures	MAE	Mean absolute error
(regression)	RMSE	Root mean-squared error
Profitability measures	TDP	Top decile profit
	Profit index	Profit index

3.4 CASE STUDY AND DATA

3.4.1 *Case study*

We work with a large parcel logistics provider in Germany that offers both B2C and B2B services. Our research focuses on parcel delivery in the B2B customer segment. According to Esser and Kurte (2020) more than 2 million B2B customers were served with B2B parcel services in Germany in 2019. They further state that 60 % of all company branches use parcel services daily. The parcel volume increased by 3.8 % in 2019 to a total of 3.65 billion shipments whereas 44 % are B2B related. The authors further state that the share of B2B parcel shipments is significant and B2B services are the backbone of the German society and economy. The total revenue of the parcel industry is estimated to be 20.4 billion Euro in 2019 (Esser and Kurte 2020). The German market is highly competitive with high price sensitivity among customers (Esser and Kurte 2020). Thus, churn prevention is a strategic necessity to avoid losing market share to competitors.

The data comprise monthly observations of 30,000 random customers over the years 2012 to 2017. The descriptive statistics of the time series in focus of

the analysis can be found in Table 3.5. We present both the mean and the standard deviation across all time series.

Table 3.5: Descriptive statistics - demand size

	Mean	Standard deviation
Min.	30.53	0.00
1st Q.	183.74	86.22
Median	370.71	175.64
Mean	642.58	329.70
3rd Q.	812.32	381.92
Max.	23902.42	25156.43

We further present two exemplary time series in Figure 3.3 and 3.4. Both show the full time series with 67 monthly observations since 2012. Figure 3.3 shows a time series with seasonality patterns but without any (partial) defection whereas Figure 3.4 shows a time series where demand drops to zero in period 61 and following.

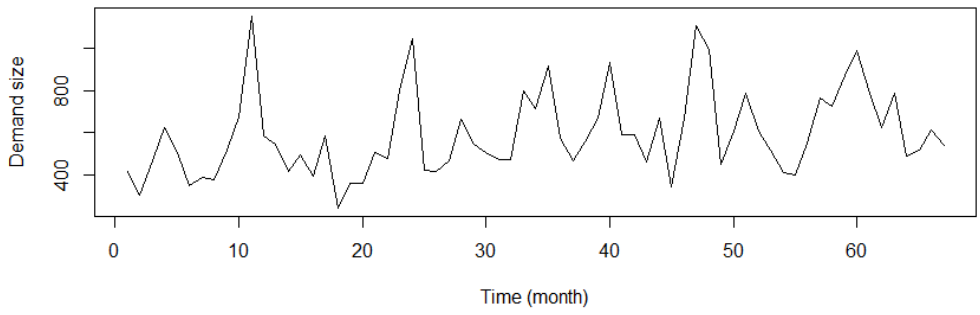

Figure 3.3: Time series example without partial defection

We refer to the independent variables associated with these observations as *features*, which are captured for all customers at three yearly intervals: in January 2015, January 2016, and January 2017. Despite the contractual setting, we assess partial defection because the shipment volumes are not fixed. We observe that customers often switch to another provider slowly over time. It is therefore not enough to take only complete defectors into account because then defectors would not be identified until almost all demand had already shifted to a competitor. Hence customers are classified as defectors ($= 1$) or non-defectors ($= 0$)

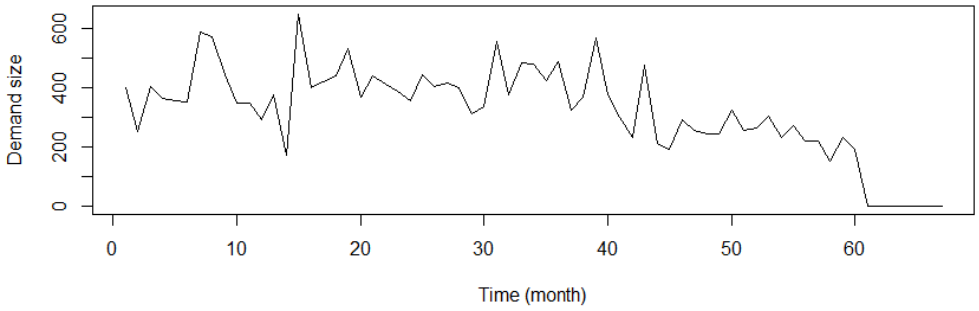

Figure 3.4: Time series example with (partial) defection

depending on the difference between the total demand in a four months time frame in the observation year and demand during the same four months of the previous year. We test different prediction time horizons to balance the time required to initiate retention actions and the predictive performance. We use a four-month total to minimize the influence of short-term fluctuations that could dominate observations limited to a shorter period. To define a customer as a defector, we test different defection thresholds which translate into a reduction in demand as compared with the same period in the previous year.

We split our data into three parts: (i) the training data, which are used to train the prediction models; (ii) the validation data, to evaluate prediction performance and identify the most ensemble-appropriate methods; and (iii) the separate test data set, which we use to measure predictive accuracy. We start by randomly assigning 65% of the customers to the training set, 20% to the validation set, and 15% to the test set. Then we take the 2015 and 2016 observations of the chosen customers for the training and validation set and the 2017 observations of the test customers for the test set. Thus we simulate the actual analytics approach at the logistics provider, which can use its past data (here, for years 2015 and 2016) to train and evaluate the models intended for application to its current data (here, for 2017). We ensure that both the training and validation data sets include the same share (7%) of defectors. Yet because the share of defectors may change over time, we use a different share (12%) for the test set.

3.4.2 *Description and preprocessing of the data set*

The data at hand consists of the monthly time series data as described above plus additional features mesaures at three points in time (2015/2016/2017). We use a set of nearly 500 features, from various sources at the logistics provider, that can be grouped into three thematic clusters:

- transactional data (e.g., volumes, peak-volume month, prices for different products, contract duration, volume share of selected weekdays);
- customer master data (e.g., size, industry, location, contract length, usage of specific services such as CO2 compensating shipments or warehousing, profitability, distance to fulfillment center of competitive parcel service providers); and
- customer service information (e.g., icoming/outgoing calls, personal meetings, complaints per topics such as pricing/delays).

For time-series–based features, such as revenue and volume, we use several different time horizons and reference values. Table 3.6 summarizes the main features that are directly derived from each customer's monthly revenue time series.

Next we use the *seasonal and trend decomposition using Loess* (STL) approach (Cleveland et al. 1990) to divide the revenue time series into a seasonal component, a longer-term trend, and a remainder component. We use these three parts of the time series to create the additional features listed in Table 3.7.

In addition, we calculate the time-series–based features described in Table 3.8. We also use the *auto-regressive integrated moving average* forecasting method to predict demand during the four-month target period. The resulting feature is calculated by dividing the obtained forecast by the demand from the same periods in the previous year.

There are two disadvantages to a data set that, like ours, has a wide variety of features. First, calculations take longer when a larger data set is involved; second, interpretive difficulties increase with the number of available features. However, data complexity can be reduced by three different types of feature selection methods: filter, wrapper, and embedded (Liu et al. 2010). Filter models (e.g., chi-squared test, information gain) analyze the data's general characteristics and use an assessment criterion such as distance to rank the features (Guyon and Elisseeff 2003). Wrappers (e.g., recursive feature selection or Boruta) embody a specific learning approach and use performance on those terms as their assessment criterion (Liu et al. 2010). Embedded methods (e.g.,

Table 3.6: Overview of revenue time-series features

Feature	Explanation
fluc_1m	Revenue of last month compared to same month in previous year (same also for two and four months)
revenue_old	Revenue in the four-month target period in previous year
share_1m_revenue	Share of revenue in last month compared to full year (same for 4 and 6 months)
share_december_revenue	Share of revenue in December compared to full year (same for other months)
share_Q1_revenue	Share of revenue in first quarter compared to full year (same for other quarters)
share_summer_revenue	Share of revenue in summer months compared to full year (same for other seasons)
share_saturday_revenue	Share of revenue on Saturdays compared to full week (same for other weekdays)

Table 3.7: Overview of time-series features using STL

Feature	Explanation
trend_4m	Trend of last 4 months compared to the 4 months before (same also compared to the same 4 months 1 and 2 years earlier)
remainder_4m	Remainder of last 4 months compared to the 4 months before (same also compared to the same 4 months 1 and 2 years earlier)
trend_vs_remainder_12m	Trend compared to remainder of last 12 months
seasonal_vs_remainder_12m	Seasonal compared to remainder of last 12 months

LASSO) incorporate feature selection into the classifier and thus select the features while the model is being created; the objective function is then used to evaluate the performance of a specific feature subset (Liu et al. 2010). Each of these methods has its respective advantages and disadvantages. Filter methods are easy to apply and require limited calculation effort, but their predictive

Table 3.8: Overview of time-series–based features

Feature	Explanation
skewness	Measures the asymmetry of a time series
kurtosis	Measures the tailedness of a time series
breakpoints	Number of structural breaks in a time series
teraesvirta	Teraesvirta neural network test for the nonlinearity of a time series
auto_corr	Ljung–Box test for serial correlation of a time series
hurst	Hurst exponent; measures the long-term memory of a time series

performance often falls short of results derived using wrapper and embedded methods (Guyon and Elisseeff 2003).

We test two different wrapper feature selection methods. *Recursive feature elimination* (RFE) is based on iteratively training new models and, after each iteration, removing the data set's worst-performing feature. The model with the best performance is ultimately used to select the chosen features. With this method, cross-validation resampling is used to avoid overfitting; typical implementations use RF and 10 times cross-validation (Kuhn 2012). The second method, *Boruta*, selects relevant features by comparing the feature importance to random permutations of the original feature. It proceeds as follows: all features are duplicated by "shadow" features, which are randomly shuffled copies of the original features. After fitting a random forest model, features that are less important than its shadow duplicates are removed. This iterative approach continues until a predefined number of iterations has been reached or until all features have been classified as either important or unimportant (Kursa et al. 2010).

3.5 RESULTS

In this section, we first report our findings on the predictive performance of the tested classification models (Section 3.5.1) and then describe a sensitivity analysis in which we compare the classification effects of different thresholds and lead times (Section 3.5.2). Next, we present the results of regression models (Section 3.5.3) and of combinations of classification and regression models (Section 3.5.4). We then discuss insights derived from the profitability focus

approach (Section 3.5.5) and the interpretability focus approach (Section 3.5.6). We conclude with information related to feature importance (Section 3.5.7).

3.5.1 *Results of classification models*

Table 3.9 reports accuracy results of the tested classification models—including all data-balancing methods but without feature selection. The profit measures TDP and profit index are based on ranking customers according to the product of (a) the firm's current profit and (b) the defection probability calculated using the respective prediction model.

Table 3.9: Predictive performance of all models: No feature selection (column rankings in parentheses)

	Balancing	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
LogR	none	0.710 (24)	0.275 (24)	2.630 (25)	549,594 (13)	674,061 (20)
	weights	0.723 (17)	0.281 (21)	2.777 (22)	490,559 (26)	676,341 (17)
	down	0.709 (25)	0.263 (25)	2.667 (24)	494,021 (24)	649,259 (27)
	up	0.719 (21)	0.277 (23)	2.762 (23)	489,422 (27)	676,335 (18)
	smote	0.723 (18)	0.288 (18)	2.930 (17)	510,489 (21)	667,578 (23)
RF	none	0.748 (9)	0.353 (4)	3.222 (5)	556,405 (8)	696,460 (10)
	weights	0.750 (8)	0.354 (3)	3.281 (4)	551,367 (12)	697,282 (8)
	down	0.761 (2)	0.322 (8)	3.018 (14)	552,402 (11)	697,036 (9)
	up	0.745 (10)	0.310 (12)	3.054 (11)	548,003 (14)	698,635 (7)
	smote	0.759 (3)	0.315 (11)	3.061 (9)	555,892 (9)	699,148 (6)
NN	none	0.689 (26)	0.200 (30)	1.410 (30)	466,863 (29)	634,502 (29)
	weights	0.739 (14)	0.296 (14)	2.959 (15)	543,444 (15)	693,044 (11)
	down	0.722 (19)	0.286 (19)	2.798 (21)	497,939 (23)	654,373 (26)
	up	0.716 (22)	0.291 (17)	2.908 (18)	535,051 (18)	692,038 (13)
	smote	0.722 (20)	0.279 (22)	2.937 (16)	505,577 (22)	669,937 (22)
GBM	none	0.758 (5)	0.348 (6)	3.295 (2)	553,888 (10)	687,225 (15)
	weights	0.758 (4)	0.355 (2)	3.310 (1)	596,499 (2)	711,573 (2)
	down	0.765 (1)	0.359 (1)	3.208 (6)	582,315 (4)	706,289 (3)

Continued on next page

Table 3.9: Predictive performance of all models: No feature selection (column rankings in parentheses) (continued)

	Balancing	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
	up	0.757 (6)	0.349 (5)	3.288 (3)	605,735 (1)	713,729 (1)
	smote	0.741 (13)	0.322 (9)	3.098 (8)	583,425 (3)	703,097 (4)
C5.0	none	0.657 (30)	0.236 (27)	2.294 (28)	489,182 (28)	664,981 (24)
DT	weights	0.675 (27)	0.217 (29)	2.126 (29)	379,699 (30)	632,953 (30)
	down	0.744 (11)	0.324 (7)	3.127 (7)	562,160 (5)	701,805 (5)
	up	0.711 (23)	0.283 (20)	2.835 (20)	539,103 (17)	646,655 (28)
	smote	0.729 (15)	0.304 (13)	3.032 (13)	558,164 (7)	679,605 (16)
SVM	none	0.661 (29)	0.236 (26)	2.528 (26)	492,183 (25)	661,589 (25)
	weights	0.663 (28)	0.233 (28)	2.462 (27)	527,444 (20)	671,652 (21)
	down	0.741 (12)	0.293 (16)	2.901 (19)	529,984 (19)	689,885 (14)
	up	0.753 (7)	0.315 (10)	3.061 (9)	559,041 (6)	692,041 (12)
	smote	0.726 (16)	0.293 (15)	3.047 (12)	539,296 (16)	675,092 (19)
sBG	n/a	0.563 (31)	0.125 (31)	1.188 (31)	182,623 (31)	418,594 (31)

The best prediction models reach an AUC-ROC of up to 0.765, an AUC-PRC of up to 0.359 (both GBM down), and a TDL of up to 3.310 (GBM weights). With regard to the profit measures, the best models achieve a TDP of more than €605,000 and a profit index of more than €713,000 (both GBM up). Overall, GBM provides the best predictive performance, followed by RF. Across all methods, the downsampling data-balancing approach has on average the highest values for the accuracy measures (AUC-ROC, AUC-PRC, and TDL) whereas the upsampling balancing approach yields the highest profit measures (TDP and profit index). The probability model sBG performs significantly worse compared to all other methods; in fact, its results are only slightly better than would follow from random guessing. We also observe a high correlation between the accuracy measures and profit measures (Pearson's $\rho > 0.85$ between AUC-ROC and both TDP and the profit index).

We use Boruta and RFE feature selection to reduce the number of features in the data set. That number is reduced from 225 to 154 with Boruta and to 63 with RFE. (The accuracy measures for prediction methods using Boruta or RFE feature selection are tabulated in the Appendix B in Tables B.1 and B.2,

respectively.) The accuracy of the GBM and RF models using Boruta and RFE feature selection are both close to the ones without any feature selection; thus the AUC-ROC of GBM orig is 0.758 both with and without Boruta feature selection and is 0.753 with RFE. Yet the NN and LogR models, which exhibit better performance than any of the two tested feature selection methods, improve still further: the AUC-ROC for NN orig increases from 0.689 to 0.733 with Boruta and to 0.758 with RFE. The same observation regarding the performance of LogR and NN holds true for the profit measures, especially when one compares results of the full-featured models with results from models with RFE (i.e., the TDP for NN orig increases from €466,863 to €540,348). On average across all classifiers, there is almost no difference among the three accuracy measures when we compare Boruta or RFE feature selection with the results of models that use all features. In contrast, the profit measures do improve slightly, on average, with either Boruta or recursive feature elimination.

Following our application of different prediction models, data-balancing methods, and feature selection mechanisms, we test the combination of unsupervised and supervised classification in a clustering approach. The elbow method (see Section 3.3.2) indicates that three is the appropriate number of clusters for our data set. Next, we assign the customers from both the training and the test data set to these clusters and train separate models for each of the clusters. One cluster contains 16% of all observations, and each of the other two clusters contains 42%. We present the results of the GBM classification method as an example. Note that we cannot use the downsampling data-balancing method because there are so few defectors in one of the training clusters. Results of the clustered approach are reported in Table 3.10.

Table 3.10: Predictive performance of the clustered approach with GBM method

Data balancing	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
orig	0.754	0.354	3.178	540,311	690,716
weights	0.762	0.347	3.215	578,667	703,789
up	0.756	0.336	3.164	568,516	692,895
smote	0.739	0.306	2.937	544,958	686,026

We find that the results of all three accuracy measures for all tested classifiers with clustering are similar to the results without clustering. However, the same cannot be said of the profit measures. In almost all cases (except the

profit index for orig), performance of the non-clustered approach is better than that of the clustered approach (e.g., respective TDLs for orig of €553,888 and €540,311). We assume that these outcomes reflect the training data's being—in the non-clustered approach—broader and more diverse, which facilitates the generalizing performed by algorithms.

We now present results of the ensembling methods that combine different classification models. Table 3.11 reports findings related to the heterogeneous bagged approach for combinations with different numbers of base classifiers (all, top 3, top 5, top 10), where rankings are determined by the classifiers' predictive performance on the validation data set. The results when using different stacking classification models (LogR, RF, NN, GBM, C5.0 DT, SVM) are given in Table 3.12.

Table 3.11: Predictive performance of the bagging approach

Method	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
mean_all	0.765	0.352	3.288	558,679	706,081
mean_top3	0.760	0.345	3.281	598,988	712,847
mean_top5	0.764	0.355	3.317	599,260	713,887
mean_top10	0.766	0.358	3.295	588,760	709,928

Table 3.12: Predictive performance of the stacking approach

Method	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
LogR	0.768	0.354	3.273	578,060	704,851
RF	0.734	0.328	3.193	541,184	687,338
NN	0.766	0.350	3.251	566,774	702,637
SVM	0.604	0.234	2.404	459,725	660,658
C5.0 DT	0.767	0.351	3.332	567,227	702,379
GBM	0.740	0.314	2.974	561,999	618,679

The results in Table 3.11 show an improvement in predictive performance compared to most of the base classifiers. An ensemble of the best five models (according to their accuracy on the validation data set) leads to slight improvements, especially in the profit measures, over the downsampled GBM base

classifier: the AUC-ROC rises from 0.764 to 0.760. However, the AUC-PRC falls from 0.355 to 0.335, the TDL from 3.317 to 3.135, the TDP from €599,260 to €589,256, and the profit index from €713,887 to €706,218). In contrast, the stacked models in Table 3.12—when compared to the best base classifiers—yield similar results for the accuracy measures but worse results for the profit measures. Across all measures, the LogR, NN, and C5.0 DT stacking models work best.

3.5.2 *Sensitivity analysis for classification models*

Table 3.13 presents results of our sensitivity analysis regarding lead time: the amount of time between the prediction and the customer's expected action.

Table 3.13: Predictive performance of the GBM model with different lead times

Lead time	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
1	0.786	0.359	3.645	566,760	592,713
2	0.744	0.325	3.107	560,928	696,123
3	0.743	0.322	3.087	529,257	652,263
5	0.710	0.336	2.681	626,703	898,443
7	0.661	0.324	2.347	640,982	1,041,990

There is a clear correlation between lead time and prediction accuracy. All accuracy measures confirm that, the shorter the lead time, the better the predictive performance. The only exception is the AUC-PRC result for a lead time of five months that is worse than the result with a lead time of seven months. In contrast, the profit measures improve significantly with longer lead times. But these outcomes come with a major caveat because they are not truly comparable given the different periods examined. In particular: using a lead time of two months places the focus on total demand from April to July; whereas a lead time of seven months will focus on total demand from September to December. As the customers of our research partner switch slowly to another provider, the demand of defectors declines over time. Hence the further one looks into the future, the larger the demand gap becomes. Since change in demand is the basis for our profit measures, it follows that these measures tend to improve the further we look into the future.

We present the prediction accuracies for different partial defection thresholds in Table 3.14. The threshold value is calculated as demand in the target period divided by demand during the same period in the previous year. A customer is classified as a defector if that ratio falls below the threshold value.

The results in Table 3.14 show that, the higher the threshold value, the worse the AUC-ROC and TDL but the better the AUC-PRC. These outcomes can be explained by the increasing number of defectors in data sets with higher threshold values, which reduces data imbalance. Overall, profit measures based on the 0.25 threshold perform significantly worse than those based on any of the other four tested thresholds. In such comparisons, one must distinguish between the different data-balancing methods. For instance, profit results of the orig and SMOTE improve under higher thresholds whereas those of the weighted and upsampling balancing methods decline. These differences, too, reflect the data set's changing exent of imbalance. The greater the imbalance (and hence the fewer defectors in the data), the more accurate are the results for the weighted and upsampling methods of balancing.

The results of our signal-to-noise approach—in which we apply different thresholds for different customer groups depending on their past demand variations—do not differ significantly from the results described previously.

3.5.3 *Results of regression models*

Results of the regression models are reported in Table 3.15, where MAE and RMSE are the performance measures and TDP and profit index are the profit measures. Both of these profit measures are based on ranking customers according to the expected profit loss—that is, the difference between the current profit and the future profit as predicted using the respective regression model.

The GBM model is the most accurate for three of the four measures; only NN has a higher profit index result. The SVM model is extremely accurate when measured by MAE but performs worse than the other models in terms of the RMSE measure. This finding suggests that there are more large errors for the SVM model that are weighted higher using the RMSE measure. The results of the profit measures for all methods are significantly lower than the outcomes of the classification models.

Table 3.14: Predictive performance of the GBM model with different partial defection thresholds (column rankings in parentheses)

TH	Balancing	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
0.25	none	0.755 (11)	0.227 (25)	4.017 (5)	528,321 (25)	678,172 (25)
	weights	0.785 (1)	0.258 (23)	4.440 (1)	566,170 (18)	698,330 (19)
	down	0.780 (2)	0.260 (22)	4.261 (3)	561,388 (21)	696,340 (22)
	up	0.778 (3)	0.264 (21)	4.408 (2)	563,571 (19)	696,740 (21)
	smote	0.776 (4)	0.241 (24)	4.261 (3)	571,202 (14)	695,322 (23)
0.50	none	0.758 (7)	0.348 (19)	3.295 (7)	553,888 (23)	687,225 (24)
	weights	0.758 (6)	0.355 (17)	3.310 (6)	596,499 (3)	711,573 (4)
	down	0.765 (5)	0.359 (16)	3.208 (9)	582,315 (10)	706,289 (14)
	up	0.757 (8)	0.349 (18)	3.288 (8)	605,735 (1)	713,729 (2)
	smote	0.741 (19)	0.322 (20)	3.098 (10)	583,425 (9)	703,097 (18)
0.60	none	0.749 (15)	0.405 (14)	2.829 (14)	575,032 (13)	697,952 (20)
	weights	0.752 (12)	0.412 (13)	2.877 (13)	567,094 (15)	708,859 (9)
	down	0.756 (10)	0.429 (11)	2.993 (12)	560,861 (22)	707,394 (12)
	up	0.757 (9)	0.422 (12)	2.999 (11)	583,668 (8)	709,453 (7)
	smote	0.733 (24)	0.387 (15)	2.660 (15)	594,027 (5)	703,363 (17)
0.70	none	0.750 (13)	0.499 (7)	2.513 (18)	579,495 (11)	706,865 (13)
	weights	0.748 (16)	0.504 (6)	2.521 (17)	566,630 (16)	708,923 (8)
	down	0.746 (17)	0.494 (9)	2.464 (19)	586,460 (7)	708,612 (10)
	up	0.749 (14)	0.498 (8)	2.532 (16)	576,461 (12)	711,884 (3)
	smote	0.733 (23)	0.476 (10)	2.350 (20)	598,366 (2)	709,851 (6)
0.80	none	0.735 (22)	0.580 (4)	2.160 (24)	587,958 (6)	714,009 (1)
	weights	0.742 (18)	0.583 (3)	2.171 (23)	553,377 (24)	705,641 (15)
	down	0.740 (20)	0.583 (1)	2.185 (22)	562,199 (20)	705,326 (16)
	up	0.740 (21)	0.583 (2)	2.188 (21)	566,207 (17)	707,886 (11)
	smote	0.724 (25)	0.561 (5)	2.059 (25)	594,400 (4)	710,319 (5)

Note: TH = threshold.

Table 3.15: Predictive performance of regression models (column rankings in parentheses)

Model	MAE	RMSE	TDP	Profit index
LinR	0.485 (5)	1.650 (3)	493,677 (4)	501,269 (5)
RF	0.454 (3)	1.646 (2)	468,487 (6)	458,478 (6)
NN	0.472 (4)	1.727 (6)	511,276 (3)	630,592 (1)
GBM	0.437 (1)	1.643 (1)	536,263 (1)	577,902 (2)
CART DT	0.490 (6)	1.678 (5)	493,056 (5)	574,516 (3)
SVM	0.437 (1)	1.668 (4)	514,720 (2)	561,175 (4)

3.5.4 *Results of combinations of classification and regression models*

After testing the described classification and regression approaches, we combine both predictions with the aim of further increasing the profit lift through retention actions (as described in Section 3.3.6). For this purpose, we test and compare two different methods. Under approach A, we sort customers according to the product of the predicted defection probability and the predicted profit loss; in approach B, customers are sorted by the average of their classification and regression rankings. As an example, we present the results for approach A with the GBM classification models. The TDP results are given in Table 3.16, and the profit index results are reported in Table 3.17.

Table 3.16: Top decile profit for combinations of GBM classification and different regression models (row rankings in parentheses)

Data balancing	LinR	RF	NN	GBM	SVM	CART DT
orig	475,345 (4)	412,395 (6)	476,628 (3)	490,521 (2)	458,810 (5)	519,100 (1)
weights	500,092 (5)	454,995 (6)	508,585 (4)	546,142 (2)	513,578 (3)	563,123 (1)
down	497,466 (5)	460,328 (6)	514,113 (4)	547,739 (2)	518,851 (3)	564,524 (1)
up	496,179 (5)	457,221 (6)	512,440 (3)	545,391 (2)	511,183 (4)	562,749 (1)
smote	485,293 (5)	446,961 (6)	510,470 (3)	529,706 (2)	499,769 (4)	564,546 (1)

Table 3.17: Profit index for combinations of GBM classification and different regression models (row rankings in parentheses)

Data balancing	LinR	RF	NN	GBM	SVM	CART DT
orig	483,621 (5)	433,931 (6)	642,807 (1)	561,407 (3)	546,709 (4)	581,528 (2)
weights	481,354 (5)	436,709 (6)	645,431 (1)	572,731 (3)	558,989 (4)	593,941 (2)
down	488,775 (5)	443,383 (6)	640,335 (1)	575,264 (3)	559,866 (4)	595,014 (2)
up	480,782 (5)	436,762 (6)	643,777 (1)	573,015 (3)	558,045 (4)	595,138 (2)
smote	482,139 (5)	439,511 (6)	646,396 (1)	571,583 (3)	555,994 (4)	593,372 (2)

The results show that, regardless of the data-balancing method, the TDP is highest when GBM classification is combined with CART regression whereas the profit index is highest when GBM classification is combined with NN regression. Although most of the profit measures results are better (higher) than those obtained using the pure regression approach presented in Section 3.5.3, their performance is significantly lower than that of the base classifiers presented in Section 3.5.1. Our analysis of all the other classifiers (i.e., besides GBM) yield the same qualitative results.

Approach B, under which customers are sorted according to the average rank of both the classification and regression predictions, leads to strongly similar results that are likewise worse than those derived via the pure classification approach.

3.5.5 *Results of profitability focus*

In addition to evaluating predictive performance, we seek to improve that performance with respect to high-profit customers and thereby maximize overall profit (as described in Section 3.3.6). We present the results for two different approaches. First, we use clustering. Table 3.18 reports performance measures for the ABC clusters that are based on customer-specific profits. As an example, we present the results of the GBM models; Table 3.19 gives the accuracy measure results for each of the three clusters. Second, we use weighting. Table 3.20 reports results of the weighted approach (with weights of 3 for customers in cluster A, 2 for those in cluster B, and 1 for cluster C).

Table 3.18: Predictive performance of the GBM method with ABC clustering by profits

Data balancing	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
orig	0.738	0.345	3.186	518,149	666,137
weights	0.747	0.339	3.135	601,713	701,849
down	0.760	0.340	3.127	602,660	706,689
up	0.743	0.341	3.244	570,176	686,206
smote	0.750	0.330	3.091	595,536	705,372

Table 3.18 reveals that almost all accuracy measures perform worse than does the base classifier without ABC clustering; only the results for the model with SMOTE data balancing are better. The TDP is higher with ABC clustering for

weights, down, and SMOTE, and the profit index is slightly better for down and SMOTE. In all other cases, the results of the base GBM classifier outperform the ABC clustered approach. Here we present only GBM as an example but find comparable results for the other methods. We obtain similar findings with the same approach but when using revenue clusters instead of profit clusters.

Table 3.19: Predictive performance of the GBM method by cluster without data balancing

Cluster	AUC-ROC	AUC-PRC	TDL
A	0.771	0.386	4.012
B	0.769	0.366	3.655
C	0.719	0.329	2.827

The results for each of the clusters in Table 3.19 show that the accuracy is greatest for cluster A—that is, for customers that generate the most profits. This finding is in accord with our approach to calculating profit (with TDP and profit index measures) because we use the profit also to sort customers and only then calculate the profit measures.

Table 3.20: Predictive performance of ABC weighting by profits

Method	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
LogR	0.692	0.278	2.747	362,246	570,163
RF	0.751	0.356	3.281	547,579	695,089
NN	0.753	0.313	2.966	565,547	695,958
SVM	0.691	0.252	2.440	548,271	687,758
GBM	0.756	0.348	3.361	560,341	695,724
C5.0 DT	0.699	0.275	2.952	514,622	670,232

The results of the weighted approach (see Table 3.20) generally show no improvement over the various base classifier results. In particular, the results of the profit measures are worse. We shall present results only for the approach using weights 3, 2, and 1; similar results are obtained using the respective weights 10, 3, and 1. Finally, the same qualitative results are obtained when we use revenue (rather than profits) to determine the ABC weights.

3.5.6 *Results of interpretability focus*

Our research has documented that the directly interpretable models LogR and C5.0 DT are each significantly less accurate than the more complex models GBM and bagged ensembling. Since the goal should be to achieve high accuracy *and* interpretability, we test two methods for explaining the predictions of complex and otherwise non-interpretable models: LIME and data permutation. We evaluate the effectiveness of both methods by comparing customers' ranked features based on LIME and data permutation on the LogR model with their ranked features based on our estimated coefficients for the LogR model. The Kendall rank correlation coefficient is a measure that can be used to compare the two rankings.

The average correlation of all customers in the test data set between the feature rankings based on LogR and the permutation approach is 0.82; for all customers, this value is significant at the 1% level. In contrast, the correlation between the ranking based on LogR and the one based on the LIME approach is not signifcant for most customers. Hence we conclude that our permutation approach enables the extraction of customer-specific feature importance. This approach can easily be used for other non-interpretable models as well. It follows that the selection of a model for predicting defection can be based solely on predictive performance—that is, irrespective of any interpretability constraints.

Table 3.21: Overview of selected customers

id	rank	rev	profit	prob	reason_1	reason_2	reason_3
234	367	6384	638	0.81	share_dec (H)	volume _v_av (H)	longest _contract (L)
75	368	8773	263	0.76	trend_12m (L)	share _december (H)	parcel_vol (L)
9627	369	21964	4393	0.64	first_m _active (L)	arima (L)	dist_hub (H)

We use the information so extracted to build an "information board" for the sales agents (see Table 3.21), which contains customers' identifies and ranks based on their respective defection probabilities. The revenue and profit over

the preceding four months indicates a customer's potential financial effect on the firm. Subsequent columns in the table show the predicted defection probability and the top three reasons for defecting (as given by the extracted feature ranking for each customer). The notation (H) or (L) indicates whether the value of a customer's focal feature is, respectively, higher or lower than the average across all customers. Sales agents who have knowledge of why a customer is predicted to defect can use that knowledge during the retention call.

3.5.7 *Feature importance*

It is helpful to understand not only the customer-specific features that indicate reasons for partial defection but also the features across all customers that drive defection. Hence we compile, in Table 3.22, the leading features from the GBM model *without* feature selection. We restrict the list to the top 20 features as ranked by their importance in the upsampled model that has the highest profit measures among the GBM models (see Table 3.9). Table 3.22 reports mean values that are scaled from 0 to 100.

Most of the top features in Table 3.22 are transactional and reflect the relative value of certain months (e.g., summer revenue, December share of sales). Other features, such as the coefficient for varation (cv2), are time series–based. It is noteworthy that the autoregressive integrated moving average (ARIMA) forecast is one of the most predictive features. A comparison of rankings for the different data-balancing methods shows that they are closely matched, especially with regard to the top three features they identify. We observe the greatest difference across rankings for the SMOTE balancing approach (trend_12m is ranked at position 70).

Table 3.22: Variable-importance GBM models (column rankings in parentheses)

Feature	none	weights	down	up	smote
share_4m_volume	100 (1)	100 (1)	100 (1)	100 (1)	100 (1)
trend_6m	42.65 (2)	40.55 (3)	36.47 (3)	39.79 (2)	14.78 (7)
share_1m_volume	22.67 (5)	41.11 (2)	44.35 (2)	39.62 (3)	11.87 (9)
share_6m_volume	36.77 (3)	16.41 (6)	23.47 (5)	17.2 (4)	7.22 (13)
cv2	6.91 (16)	17.43 (5)	23.58 (4)	16.95 (5)	7.44 (12)
share_Q2_revenue	20.48 (6)	20.23 (4)	15.61 (6)	15.97 (6)	3.50 (23)
trend_vs_remainder _12m	8.13 (11)	14.64 (7)	10.93 (7)	13.94 (7)	1.69 (34)
first_month_active	1.89 (56)	11.71 (8)	7.08 (11)	12.13 (8)	4.82 (20)
cv2_12m	6.50 (17)	8.49 (12)	7.06 (12)	8.86 (9)	1.23 (47)
fluc_1m	15.21 (9)	9.59 (9)	7.43 (10)	8.46 (10)	2.62 (26)
longest_contract	1.46 (66)	9.46 (10)	2.96 (21)	7.11 (11)	0.53 (94)
share_march	15.69 (8)	8.27 (13)	5.95 (14)	6.87 (12)	3.29 (24)
arima	16.19 (7)	7.09 (16)	4.28 (18)	6.84 (13)	2.34 (28)
trend_24m	2.85 (36)	7.82 (15)	2.90 (22)	6.41 (14)	2.45 (27)
volume_parcel	7.05 (13)	9.24 (11)	7.48 (9)	5.50 (15)	1.73 (33)
share_december	6.93 (14)	7.95 (14)	5.70 (15)	5.44 (16)	9.95 (10)
summer_revenue	22.70 (4)	4.81 (18)	7.50 (8)	5.23 (17)	5.96 (17)
trend_12m	6.93 (15)	5.43 (17)	1.79 (26)	5.23 (18)	0.79 (70)
delivery_month_range	0.24 (126)	3.50 (20)	6.95 (13)	4.48 (19)	0.14 (154)
defected_before	5.34 (18)	1.77 (34)	0.51 (44)	2.36 (20)	16.58 (6)

3.6 SUMMARY OF RESULTS AND MANAGERIAL IMPLICATIONS

Of the base classifiers, the GBM models work best, followed by RF. Bagged ensembling yields results similar to those under the GBM method. Thus our results show the advantage of more complex classification models (e.g., GBM, bagged ensembling) in comparison with the simpler LogR and DT methods (the AUC-ROC is 0.764 for bagged ensembling of the top 5 models, versus 0.695 for the best LogR model and 0.744 for the best C5.0 DT model). We can see also that, on average and across all tested methods, the downsampled approach leads to the highest AUC-ROC and AUC-PRC; that method's effectiveness is closely

followed by upsampling and SMOTE. For the profit measures, upsampling yields the best results on average.

If the goal is to achieve greater financial gain through retention activities, then we find that the firm should prefer using a classification model to adopting either the regression approach or the combined classification and regression approach. However, we do not consider retention offerings in form of discounts due to the fact that the research partner does use such measures. Using discounts will change the expected profitability due to the impact on the customer lifetime value.

In addition, we find that the tested feature selection methods reduce the number of features without compromising predictive performance. In fact, accuracy of the LogR and NN models *improves* with fewer features. However, no significant differences arise when we compare the Akaike information criterion for models with and without feature selection. We remark also that predictive performance is improved neither by stacking nor by the combination of unsupervised and supervised clustering.

In light of the superior results delivered by complex yet non-interpretable models, one must usually prioritize either predictive performance or interpretability. However, we demonstrate that our permutation approach makes it possible to achieve both goals. The advantage of this approach is that it renders any learning model interpretable—which, in the context of our study, leads to identifying customer-specific reasons for partial defection. The firm's sales agents can use this information to understand the classification outcome's rationale and then to adapt their retention strategy accordingly.

For the threshold sensitivity analysis, there is a clear correlation between the chosen threshold value and the predictive performance measures. The higher the threshold value, the higher the AUC-ROC and the lower the AUC-PRC. These relations hold because the measures depend on the extent of class imbalance, which is defined by the partial defection threshold. A higher threshold is associated with a lower class imbalance because more customers are classified as partial defectors. Results for the profit measures show that a threshold of 0.25 leads to a significantly lower profit (after retention actions) than do all the higher tested thresholds. As before, this outcome reflects the small number of partial defectors in the case of a low threshold, which makes prediction more difficult. For our thresholds above 0.25, the choice of a threshold-appropriate data-balancing method is more important than is the choice of the threshold itself. There is no improvement when we use different thresholds for customers grouped by demand variability; the reason, we suppose, is that past demand variation is an unreliable metric for distinguishing between temporary

and structural declines in demand. The results tend to validate our decision to use partial rather than complete defection—that is, given the higher profits (after retention actions) that can be achieved when the threshold exceeds 0.25. In the event of complete defection the threshold would be 0, in which case the results should be even worse than when threshold is 0.25. Our decision to use data-balancing methods is also supported: the lower the threshold, the more essential these methods are to achieving good results.

A critical managerial decision is to select the forecast horizon. We confirm the intuition that, the nearer (in time) the change in customer behavior, the greater the prediction accuracy. Thus we find a huge gap in predictive performance when the lead time changes from one month to two months and also when it changes from three to five months and from five to seven months. When choosing a lead time, management should account for possible changes in accuracy as well as the relevant business requirements. The latter are driven, for example, by the firm's capacity to carry out retention actions and thus the time needed to engage with all predicted partial defectors.

Managers must also decide on the number of customers that should be targeted with the retention campaign. The profit measures used in our research offer insights on each customer decile. Although we report only the TDP and the profit index, the decile profit is positive for the first four deciles of all tested models. However, profit declines significantly with each decile (i.e., GBM without data balancing TDP €553,888, 2nd decile €139,254, 3rd decile €74,089, 4th decile €33.860). A profit of more than €800,000 could be achieved if there were sufficient capacity to contact each customer in the first four deciles.

The feature importance of the trained models shows that revenue- or volume-based transactional features play a key role. This finding is in line with the observation that customers switch to competitors slowly over time. The earlier this demand shift can be identified, the sooner that retention actions can commence. Recall that the ARIMA forecast further improves predictive performance, which again is calculated with time-series data. Because the feature rankings are similar for all tested balancing methods, we conclude that the result is stable.

3.7 CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This paper illustrates the importance of predicting B2B partial defection prediction: our research partner can increase its profit (after retention actions) by more than €500,000 in the short term by targeting the first decile of customers

with the highest defection probabilities—even though the retention acceptance offer rate is only 30%. We demonstrate the advantage of using partial defection predictions in cases when customers switch their provider slowly over time, a behavior observable in contractual settings when the volume of products or services to be delivered is not fixed. The feature importance information extracted from the trained models reveals that transactional features are the most predictive because they reflect defection over time. Although partial defection covers cases from significant demand reduction to complete defection, our results show that the classification approach is superior to a regression or combined classification and regression approach. We also establish that the GBM learning model yields better results than do more complex approaches such as stacking or combinations of unsupervised and supervised learning.

The results of our sensitivity analysis document the relation between using different thresholds and lead times for predicting partial defection and for predictive performance more generally. We confirm the appropriateness of predicting partial (rather than complete) defection and also that the ability to predict defection is relatively stable for lead times between one and three months. These results provide the basis for managerial decisions concerning which threshold and lead time should be used in the specific context of a focal company. One paramount decision criteria might be the capacity to carry out the required number of retention calls within the chosen lead time. Our permutation approach allows one to explain the individual classifications made by complex models, thereby resolving the trade-off between accuracy and interpretability. This approach can improve the success of retention actions because sales agents are able to respond to the actual needs of each customer. Also, the cost of retention actions might decrease because sales agents can learn the reason for a customer's dissatisfaction *before* directly addressing that customer.

The market conditions faced by our research partner—namely, large customers, strong competition, and low switching costs—are typical in B2B settings. One can therefore follow the same approach used to predict partial defection also for other B2B players who operate in contractual settings but without fixed volumes.

There are several research directions whose pursuit would likely yield further knowledge about the prediction of partial defection in B2B settings. First, additional insights could be gleaned from sales agents (e.g., their discussion points from the previous customer contact) or from industry performance predictions—either of which might increase forecast accuracy. Second, a better understanding of the actual reasons for defection (e.g., price, service) would make it easier to plan appropriate and context-specific retention actions. Also, one could use

these reasons to train a multiclass classification approach that might perform better. Third, the profit calculation approach could be strengthened if more were known about different possible retention actions as well as their respective costs and success rates. The research partner could also introduce retention discounts that would change the profit calculation as these discounts would impact the expected profit significantly. If this is the case, one can extend the expected profit calculation to a customer lifetime value calculation. In this context, one could also analyze whether the success of retention actions depends on the lead time (i.e., is there a long-term or only a short-term effect). Finally, different time-series lengths could be used when calculating the dependent variable (here, aggregated four-month demand) to see whether a particular length is reliably associated with more accurate predictions.

4

MACHINE LEARNING–BASED PRICE SEGMENTATION BASED ON PRICE SENSITIVITY: A CASE STUDY IN THE B2B PARCEL LOGISTICS INDUSTRY[1]

4.1 INTRODUCTION

Pricing is the basic tool in revenue management, and intelligent pricing is the easiest and quickest way to increase profits (Phillips 2005). Yet success requires identifying the price that maximizes revenue without losing customers. Customers differ in their willingness to pay, and finding the optimal price for each customer is a challenging problem. One possible solution is *price discrimination*: offering different prices to different customers (or groups of customers). Such "segmented" pricing has a long history to increase both revenue and profit (Miao et al. 2019). Thus customers are usually clustered into groups based on their demand patterns, after which individual pricing strategies are applied to each group. In practice, however, the extent of customer diversity makes it difficult to distinguish usefully among the customer segments so defined (Chen et al. 2015b). It is only within the last few years that scholars have begun to examine pricing at the individual level rather than the customer segment level (Ban and Keskin 2019). Research into this "personalized" price discrimination or segmentation has been boosted by newly available data sources, such as detailed customer demographics and sales histories (Ban and Keskin 2019, Dubé and Misra 2017). Such information can help companies find, for each customer, the price that maximizes overall revenue. Most of the literature in this field incorporates price experiments designed to reveal unknown demand functions and to stipulate the corresponding optimal price changes (Qu et al. 2016). Companies use such price experiments to learn about the demand functions of their customers (den Boer 2015). The extant research mainly addresses business-to-consumer (B2C) settings with applications for e-commerce (Gupta and Pathak 2014, Miao et al. 2019), service subscriptions (Shiller et al. 2013, Dubé and Misra 2017), transportation (Chen et al. 2015b), and hospitality (Vives et al. 2018). Most of these papers deal with perishable inventory and finite selling seasons (Fisher et al. 2017).

[1]The following chapter is based on Faber and Spinler (2019c), unpublished working paper.

Another research stream in pricing is "dynamic" pricing with inventory effects. One can distinguish between selling a finite amount of perishable inventory and joint pricing and inventory problems (den Boer 2015). In the first case, one follows the ambition—especially in the airline and hotel sectors—to balance supply and demand (Miao et al. 2019). den Boer (2015) highlight that the inventory at the end of the selling hoizon gets lost. Thus, the price depends on both the available inventory and the time left in the selling period. Another application area is perishable goods with the ambition to reduce spoilage at the end of the shelf life (Adenso-Díaz et al. 2017). The second case considers models where the ambition is not only revenue driven but also analyzing the required capacity and deciding e.g., on different fare classes to offer or deciding on inventory capacities (Adenso-Díaz et al. 2017, den Boer 2015).

There are two ways in which the pricing situation we study differs from most of the previous research. First, we focus not on the B2C setting but rather on the business-to-business (B2B) setting. The B2B sector plays a prominent economic role because it accounts for nearly half of all transactions in the United States (Dwyer and Tanner 2002). Nonetheless, B2B settings appear in an extremely small fraction of research worldwide: only 3.4% of the articles in the top four marketing journals (LaPlaca and Katrichis 2009). There is an important difference between B2B and B2C. In particular, under B2B contracts the prices are commonly negotiated for each customer individually and can be changed from one purchase or price negotiation to another (Zhang et al. 2014). Also, B2B contracts are associated with longer-lasting business relationships; it follows that pricing is a crucial aspect of retaining customers. Qu et al. (2016) highlight the virtues of the price discrimination that B2B conventions enable—for instance, retaining the high revenue of individual customers that could be lost if a higher proposed price exceeds their willingness to pay. In addition, the authors argue that B2B customers are more diverse than are B2C customers.

The second difference between this study and previous work is that our research partner provides a continuous service over time, which means that neither inventory status nor the time of a sale is a factor. In most situations related to pricing strategies, the product or service is offered at a certain price; customers then decide whether to buy or not to buy. However, we study long-lasting business relationships during which regular price adjustments are made. Thus customers already consume the service at a certain price, and the question is how they will react to price increases. Customers can either accept the new price—and so leave their demand unchanged—or shift some or all of that demand to another provider. Hence this study of price discrimination is directly linked to the topic of *churn*. Price and price changes are certainly factors that

customers consider when deciding on a service provider. We therefore test for whether churn predictions correctly identify which customers are price sensitive and which are not.

Our research makes three contributions to the literature. First, we analyze how price changes affect demand in a B2B service setting. Thus we train a demand prediction model with a broad set of features, including price change information, and identify which of those features have a significant effect on demand. Second, we assess and predict price sensitivities but without conducting any price experiments. We instead train our model with data from past price changes and the subsequent demand reactions—that is, because price experiments involve a risk of losing customers. Third, we demonstrate that the simulation of an A/B test with different price change levels allows one to evaluate the accuracy of models for predicting price sensitivity and to identify the revenue potential when a firm's pricing strategy is properly aligned with the identified price sensitivities.

The rest of our paper proceeds as follows. Section 4.2 reviews the relevant literature, after which our approach to modeling is explained in Section 4.3. We then present the case study and describe the data set in Section 4.4. The results of our analysis are reported in Section 4.5 and the managerial conclusions are drawn in Section 4.6. Section 4.7 concludes with a summary and the outlook for related research opportunities.

4.2 LITERATURE REVIEW

Segmented pricing has become a popular research field in the last decade, especially with the advent of online retailing and its attendant new data sources (Miao et al. 2019). Applying price theories in practice requires knowledge of the underlying demand function (den Boer 2015). Early work assumed that the demand function is known to the seller (for an overview, see Bitran and Caldentey 2003, Elmaghraby and Keskinocak 2003). Real-world sellers, however, seldom know their customers' demand functions—especially in industries where market conditions change rapidly (Miao et al. 2019). Hence sellers must estimate the demand curve via price experiments. Den Boer (2015) reviews this field's history and current research. The author distinguishes between two literature streams in monopolist settings: one where the demand function changes dynamically over time; and another where the demand function is static but where pricing depends on the inventory level. We review the first stream because B2B services, the focus of our study, are not inventory driven.

There is an extensive literature that approaches segmented pricing from the perspective of identifying unknown demand functions. Ban and Keskin (2019) group this research into two different fields: discriminating pricing in *Bayesian* settings with iterative updates of a prior belief function (e.g., Araman and Caldentey 2009, Qu et al. 2016, Cheung et al. (2017)); and *frequentist* settings with sequential price experiments (Besbes and Zeevi 2009, Keskin and Zeevi 2014). The papers cited here all presuppose continuous prices, but Miao et al. (2019) discuss studies that assume discrete price points (Ferreira et al. 2018) or focus on segmented pricing in a changing environment (Keskin and Zeevi 2016).

Recent research (see e.g. Chen et al. 2015b, Cohen et al. 2016, Qiang and Bayati 2016) examines how considering the characteristics of individual customers can make learning more flexible with regard to those customers and to different market settings. There are many cases featuring the availability of rich data sets with customer information (e.g., browsing data), competitor information, and/or product characteristics (e.g., ratings). Of course, this information can change over time (Miao et al. 2019). Ban and Keskin (2019) go one step further by considering feature-dependent price sensitivity.

Another approach to pricing decisions is described in the literature on clustering. Ferreira et al. (2015) cluster products based on their demand characteristics before application of a pricing strategy based on those characteristics. Cheung et al. (2017) use k-means clustering to generate a set of different demand functions. For each product, a specific pricing approach is chosen based on the selected demand function.

There have been only a few applications of machine learning methods in the field of price discrimination. Gupta and Pathak (2014) propose a combined model of unsupervised and supervised learning with which to identify price ranges for different customer segments. Shiller et al. (2013) use machine learning to predict—based on Web browsing histories—the demand for Netflix subscriptions. These authors report that exploiting demographic data increases profits by only 0.30%, as compared with a 14.55% increase when Web browsing histories are used. Dubé and Misra (2017) combine machine learning with a large-scale price experiment; they find that incorporating customer demographic information into price decisions yields significant profit increases. Schlosser and Boissier (2018) assess a range of different machine learning models in terms of predicting the sales probabilities of various products. In a second step, these authors employ a dynamic programming model to identify superior pricing strategies.

Another under-researched field is price discrimination in B2B settings. Qu et al. (2016) develop a model based on approximate Bayesian inference. Zhang et al. (2014) use a hierarchical Bayesian approach and distinguish between two different customer states of trust in the seller. Both of these papers point to a difference between B2C and B2B pricing decisions that reflects the difference in their respective customer relationships. Zhang et al. (2014) also emphasize that B2B customers must make more than a binary, buy–not buy decision; they must also decide how *much* to buy.

Because frequent price experiments are not always feasible, Cheung et al. (2017) seek to achieve high performance while minimizing their reliance on them. Schlosser and Boissier (2018) underscore the need for pricing simulations—given that experiments are difficult (if not hazardous) in competitive settings.

Price sensitivity is another factor that affects churn (Shaaban et al. 2012, Zhang et al. 2012). Dominique-Ferreira et al. (2016) report that loyal customers are less price sensitive than are nonloyal customers, but these findings are not statistically significant. Stock (2005) documents an inverse relationship between customer satisfaction and price sensitivity, and Tanford et al. (2012) find that price sensitivity is a reliable indicator of future customer defection.

4.3 MODEL APPROACH

Following the research of Chen et al. (2015b), Cohen et al. (2016) and Qiang and Bayati (2016), we use customer characteristics when analyzing price sensitivity. That approach is appropriate when one considers the diversity of B2B customers in our data set. Thus we combine price-related information with a broad set of features characterizing each customer.

We use the random forest prediction method, which Bajari et al. (2015) has shown works well for predicting demand. This method's foundation dates to Breiman et al.'s (1984) introducing the classification and regression tree (CART) algorithm. A root node creates binary splits—until either (a) splitting no longer adds value to the prediction or (b) all of each node's observations have the same value. Although the method is known to be fast and remains popular, it does have some limitations. Most importantly, decision trees tend to overfit and so yield accurate predictions only with training data (Hastie et al. 2009). To overcome this limitation, the random forest approach uses an "ensembling" method: combining many different CART decision trees (see Breiman 2001), thereby reducing the variance and thus the risk of overfitting. Each of the

ensembled decision trees is a randomized variant because of the three factors described next.

First, bootstrap aggregation (a.k.a. bagging) is used—which means that the training set for each tree is a "resample with replacement" from the initial training set. Second, a random selection of the features (a.k.a. feature bagging) is used at each splitting node. Third, the predictions of all trees in the ensemble are either averaged (in the case of regression) or selected by majority vote (in the case of classification). Because the trees can be fitted in parallel, this method is quite fast (for a detailed description, see Breiman 2001). We use a grid search with three repeats of five-fold cross-validation to optimize the model's hyperparameters. In our case, the only hyperparameter requiring optimization is the one that defines the feature bagging step's number of randomly selected features. For this hyperparameter, we test values of 2, $p/2$, and p, where p is the reference data set's number of features. Thus we test both extremes: a low number of selected features that have the strongest effect; and a high number of features, which includes variables that are less influental yet may nevertheless contribute to the prediction.

Our modeling approach consists of a two-step process. First, we test for whether price has any effect at all on demand. Toward this end, we train models to predict demand either with or without price features and then compare the performance of these models. Second, we assess predictions of price sensitivity. Thus we compare different approaches to distinguishing between customers with low versus high price sensitivity to price.

4.3.1 *Assessment of price feature importance*

We test the predictive performance that results from using different feature combinations (price features, demand time-series features, and customer features) in order to identify which model delivers the highest forecast accuracy. To compare the performance of demand prediction models with and without price-related information, we use two standard metrics: the *mean absolute error* (MAE) and the *root mean-squared error* (RMSE); these performance measures

are calculated as in Eq. (4.1) and Eq. (4.2), respectively. Here, for period t, y_t captures actual demand and $\hat{y}_t$ represents predicted demand:

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T}|y_t - \hat{y}_t|; \tag{4.1}$$

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}. \tag{4.2}$$

We use the Wilcoxon signed-rank test to test for whether the forecast accuracy differs significantly between models using different feature sets. This test is a distribution-free, nonparametric technique that is widely used to compare forecast accuracies (Demšar 2006).

To learn more about the significance of particular price features, we analyze the "variable importance" ranking as determined by the best-performing model. Doing so requires iterating through all the features and calculating, for each feature, the difference between the mean squared error of an out-of-bag data sample and that of the same data after permuting the chosen variable. The last step consists of ordering the features in terms of the size of those differences (for a more detailed description of this approach, see Hastie et al. 2009).

4.3.2 *Assessment of price sensitivity classification*

In assessing price sensitivities, we test different approaches to distinguishing between customers with low and high price sensitivities; see Table 4.1 for an overview. We do not use methods designed to learn individual demand functions because the behavior of customers in our data set is highly time dependent: at different times, customers react differently to the same price change. We forgo identifying individual demand functions also because there are so few observations per customer—that is, since the price usually changes only once each year. In light of how much time elapses between two price changes (and thus between two observations), other factors affecting the demand function might themselves have changed.

The *naïve* approaches are each based on a single customer characteristic. We formulate a hypothesis about the correlation between each characteristic and the associated price sensitivity and then sort the customers accordingly. First we assume that customers under a long-term contract are relatively less price sensitive because they have been loyal in the past. Second, we assume that high demand volumes indicate low price sensitivities; that is, we hypothesize that customers that ship more have higher switching costs than do those that ship

Table 4.1: Approaches to predicting price sensitivity

Name	Description
Naïve (contract length)	Sort the customers according their contract lengths
Naïve (demand volume)	Sort the customers according their demand volumes
Price simulation	Simulate the effect of different price changes on the demand prediction
Churn prediction	Sort the customers according their predicted churn probabilities
Sensitivity prediction	Predict the expected price sensitivities (using regression or classification)

lower volumes. We use the total demand volume in the three months preceding the time of prediction to compare shipment volumes between customers. The customers are sorted from high to low—in terms of contract length or demand volume, as applies—and in this way we order them from (respectively) low to high expected sensitivity to price.

The *churn prediction* approach is similar to both naïve methods. In this case, we expect that customers with high churn probabilities are also extremely price sensitive, and we sort the customers accordingly. Since there were few instances of a customer totally abandoning our research partner, we define churn as *partial* defection. Thus a customer is classified as a churner only if its demand declines to less than half the demand volume observed during the same period one year earlier. We use a separate machine learning model—one with the same features and the same lead time as the demand prediction model—to predict churn probabilities. We also replace our random forest model with a gradient boosting model because Gregory (2018) has documented the latter method's superiority in the field of churn prediction. Stochastic gradient boosting is an ensembling method that combines weak prediction models. In each run, a new classifier is trained and the data are then reweighted to reflect the preceding run's predictive performance (see Friedman 2001). Just as in the demand prediction approach, we test the model with and without price features to establish whether (or not) accuracy improves when price information is taken into account.

The *price simulation* method is an extension of the simple demand prediction model that we use to assess price feature importance. Here we use permutations of the actual price feature values, as illustrated by Figure 4.1's example.

In each permutation, we change the price features to reflect a different price change level. For each of the 30 replacements for every customer, we obtain a simulated but updated demand prediction. Next, for each customer we fit a linear regression based on the log-transformed price change steps—between 1% and 30%—and the calculated demand predictions. The slope of the regression line then equals the customer's predicted price sensitivity.

	price_before	price_change_percent	total_price_change_percent	months_before	vol_before	avg_price_vol_cluster
Actual values	3.82	0.05	1.29	12	2927	0.90
Permutated values (here for price changes between 1 and 5%)	3.82	0.01	1.24	12	2927	0.87
	3.82	0.02	1.26	12	2927	0.88
	3.82	0.03	1.27	12	2927	0.89
	3.82	0.04	1.28	12	2927	0.89
	3.82	0.05	1.29	12	2927	0.90

Figure 4.1: Price simulation approach

Finally, the *direct prediction* model is based squarely on price sensitivities (i.e., on demand change divided by price change). We test a regression model and also a classification model. In the first case, we predict individuals' price sensitivities and order customers from the highest positive value to the lowest negative value. Second, we predict the probabilities of the focal customer having a low sensitivity and having a high sensitivity. We then use the sum of these two predicted probabilities to order the customers from the highest price sensitivity (i.e., the *largest* difference between the probability prediction for high sensitivity and the prediction for low sensitivity) to the lowest price sensitivity (*smallest* difference between the probability predictions for high and low sensitivity).

To evaluate classification performance, we split the observations in our validation data set into two groups based on the median value of the actual price sensitivities. We then use a minmax normalization to transform the price sensitivity predictions of the approaches described previously into probabilities for the group of highly price-sensitive customers. Next we evaluate the performance of the different classification models using the *area under the curve of the receiver operating characteristic* (AUC-ROC) and the *area under the curve of the precision recall characteristic* (AUC-PRC). The ROC curve is a graphical representation of how the *true positive rate* (TPR), as described in Eq. (4.3), is related to the *false positive rate* (FPR) in Eq. (4.4) for different cut-off values that define the probability threshold for distinguishing between class A and class B (Hanley and McNeil 1982):

$$\text{True positive rate} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \tag{4.3}$$

$$\text{False positive rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{4.4}$$

The PRC curve represents precision (as defined by Eq. (4.5)) in terms of recall (defined in Eq. (4.6))—again for different threshold values (He and Garcia 2009):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}; \tag{4.5}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4.6}$$

The AUC distills an ROC or PRC curve's information into a single value (Krzanowski and Hand 2009). A random classification generates a diagonal ROC curve whose AUC-ROC value is 0.5, whereas the AUC-PRC value of a random classifier is approximately the class ratio (Keilwagen et al. 2014). We use these measures also to compare the churn predictions of models with and without price information.

4.3.3 *Financial impact calculation*

To assess the financial potential of the various prediction approaches, we use a simulated A/B testing approach. We start by distinguishing between customers that encountered small and large price adjustments in the validation data set, using the median of all price change values to split customers into two groups. Next, we split each of those groups based on the price sensitivity predictions (i.e., low vs. high price sensitivities) of the models being compared. Customers whose predicted price sensitivities are above (resp., below) the median are considered to be (resp., not to be) price sensitive. Finally, we evaluate and compare—for each of the four groups—the change in demand that follows a price change. In the event of an accurate classification into low and high price sensitivities, the change in actual demand should differ significantly between those two groups. Thus we compare the change in demand of customer groups with predicted low and high price sensitivities separately for both types (large and small) of price changes. The four customer groups are depicted schematically in Figure 4.2. Our A/B test simulation is applicable because (i) price sensitivities were not considered when making previous pricing decisions and (ii) our validation data set exhibits considerable variance in the actual price changes made (1st quartile, 6%; median, 8%; 3rd quartile, 12%).

Figure 4.2 presents the four customer groups that we distinguish based on their calculated price sensitivity and the actual price change that they receive.

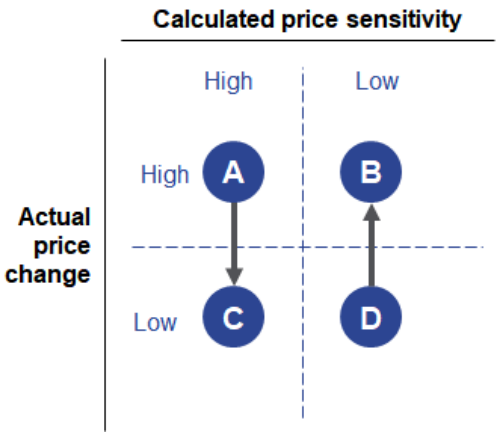

Figure 4.2: Customer groups as a function of actual price change and predicted price sensitivity

We calculate revenue potential by focusing on the two groups for which the actual price change is not aligned with the predicted price sensitivity: groups A and D in Figure 4.2. For group A, we assume that revenue will increase if one lowers the price and thereby increases demand. For group D, we assume that a higher price will generate more revenue—although demand might falter. After the price change, we compare the differences—between groups A and C and between groups B and D—in demand change and check whether those differences are statistically significant. Next, we calculate the revenue potential if the price changes for groups A and D are based on predicted price sensitivities. We use the average difference of the demand change, as well as the average difference of the price change between groups A and C and between groups B and D, to calculate the expected price and demand change and thus, ultimately, the change in revenue.

4.4 CASE STUDY AND DATA

Our research builds on a partnership with a large parcel logistics provider in Germany. This provider serves both B2C and B2B customers, but our analysis focuses on the latter. The market is highly competitive, and several other providers offer similar services. Also, the switching costs for customers are low and so they can easily shift their demand to competitors. Since contracts are negotiated individually with each customer, the prices are also customer specific. Our research partner routinely changes prices for B2B customers; these are normally price increases (due, e.g., to inflation). However, the price increases are customer specific and not all customers receive the same increase at the same

time. For some customers the price might remain the same while others receive moderate price increases and other large price adjustments. The prices charged a customer do not change more than once every six months, and they usually change just once per year. These price changes are determined at least four months in advance and, after being aligned internally, are communicated to customers. Because most price changes become effective in January, customers are notified during the busy time before Christmas—when almost all market capacity is fully utilized. This is one reason why customers refrain from changing their demand volumes until the new prices become effective. Another reason is that negotiations with alternative suppliers take time. Even so, customers generally switch without hesitation when higher prices come into effect. Due to the customer specific pricing and price adjustment approach, the identification of price sensitivities is helpful to identify the right level of price adjustments individually for each customer. It is essential to identify the features that indicate the price sensitivity and adjust the pricing scheme accordingly.

The ambition is to analyze the impact from changes in prices on future demand, individually for each customer and thereby derive the expected price sensitivity per customer. The model learns from the effect of past price adjustments to change in demand. Thereby, one is able to predict the expected demand change and thus price sensitivities as a factor to decide on planned price adjustments and sitinguish between customers with low and the ones with high price sensitivities.

We use the model developed here to study the demand change that follows a price change. However, we are uninformed about the demand split—at the customer level—between our research partner and its rivals. Hence we cannot tell whether a specific customer's demand changes reflect a true demand shift (i.e., to a competing provider) or, instead, a change in the customer's particular circumstances (e.g., variation in the demand of its end customers). Yet the latter case is rather rare, we expect (as does our research partner), so it should not significantly change our model. Note also that our study period is limited to five years (viz., 2015–2019) characterized by positive economic development and no major shocks.

We have access to monthly time-series data on customer-level demand volume and revenue from January 2012 to January 2019. The data set contains 12,000 customers in total. Because we lack information on *all* past price changes, we use a two-step approach (as needed) to identify them. First, we calculate the customer-specific parcel price as revenue divided by the number of parcels. We then perform a break-point analysis to identify the time of structural changes that, we assume, correspond to actual price adjustments (for a detailed descrip-

tion of break-point analysis, see Bai and Perron 2003). We observe that, across all customers, many price changes occur in January (nearly a third of the full year's number) while the remaining price changes are almost evenly distributed across the other eleven months.

Table 4.2: Price features: Overview

Price feature	Description
time_since_last_pc	Number of months since previous price change
price_before	Actual price before the price change
price_change_percent	Price change (in %)
total_price_change_percent	Total price change since January 2012 (in %)
vol_before	Demand volume in the reference period
avg_price_vol_cluster	Price divided by the mean price of customers in the same group (according to their shipping volumes)

We use a variety of feature sets as input data for our model: customer-based features, price features, and time-series–based features; see Table 4.2 and Table 4.3. We do not tabulate all customer-related features because there are more than 100 of them. The customer feature set includes information about, inter alia, recent customer interactions (e.g., complaints, calls), demand for other products and services (e.g., international shipments), contract lengths, and company's location.

Among the time-series features is an *autoregressive integrated moving average* (ARIMA) demand forecast for the target period. The ARIMA method builds on the approach described by Box et al. (2015). Such models are denoted ARIMA(p, d, q) where p denotes the number of time lags, d the "degree of differencing", and q the order of the moving-average model. This method is widely used for the prediction of time series (Zhang 2003). The underlying principle of ARIMA is that time-series values are a linear function of multiple past observations and random errors. This approach incorporates the Box–Jenkins method, an iterative three-step process consisting of model identification, parameter estimation, and evaluation. In the identification step, a time series is transformed to become stationary so that the mean, variance, and autocorrelation are all time invariant. We estimate the (p, d, q) parameter via a nonlinear optimization approach that minimizes errors. In the last step, goodness of fit is evaluated; at this point, either another iteration begins or the finally chosen

Table 4.3: Features of the demand time series

Time-series feature	Description
fluc_1m	Demand in the previous month ($t-1$) as compared with the same period one year earlier
fluc_2m	Demand in the previous 2 months ($t-2$ to $t-1$) as compared with the same periods one year earlier
fluc_3m	Demand in the previous 3 months ($t-3$ to $t-1$) as compared with the same periods one year earlier
volume_recent	Demand in the previous 3 months ($t-3$ to $t-1$)
volume_H_last	Demand in the previous 6 months ($t-6$ to $t-1$)
volume_before	Demand in the reference period ($t-12$ to $t-10$)
M_last_volume_v_av	Demand in the previous month ($t-1$) divided by total demand in the preceding 12 months
Q_last_volume_v_av	Demand in the previous 3 months ($t-3$ to $t-1$) divided by total demand in the preceding 12 months
Q2_last_volume_v_av	Demand in the months ($t-6$ to $t-4$) divided by total demand in the preceding 12 months
H_last_volume_v_av	Demand in the preceding 6 months ($t-6$ to $t-1$) divided by total demand in the preceding 12 months
arima	ARIMA demand forecast for the target period (t to $t+2$)
cv2	Coefficient of variation in the entire demand time series
kur	Kurtosis of the entire demand time series
skew	Skewness of the entire demand time series
cv2_12m	Coefficient of variation in the preceding 12 months
kur_12m	Kurtosis in the preceding 12 months
last_demand1	Total demand in the months $t-3$ to $t-1$
last_demand2	Total demand in the months $t-6$ to $t-4$
last_demand3	Total demand in the months $t-9$ to $t-7$
last_demand4	Total demand in the months $t-12$ to $t-10$

model is used for prediction. Hyndman et al. (2007) propose a stepwise heuris-

tic for automatically choosing the best model. The Hyndman et al. approach is used to generate all the ARIMA forecasts in this paper.

We are able either to collect or to calculate the time-series and price-based features for each month between January 2012 and January 2019. Yet because customer-based features were available only for the time between January 2015 and January 2019, we limit our analysis to that latter period. The data set is split into two parts: the *training* data set includes all price changes between June 2015 and December 2018; and the *validation* data set consists only of the price changes made in January 2019. In this way, we mimic our research partner's actual planning approach, under which past data are used for training purposes and future predictions. As an additional test of our model's stability, we adopt the same approach but with a different split of the data; here data up to December 2017 are used for training purposes and data from January to April 2018 are used for the validation data set.

For the demand prediction model, we use four months for the lead time lt between when the prediction is made and the target period. This choice, too, reflects our research partner's situation. Price changes are planned in advance so that there will be enough time for internal coordination and for communicating the changes to customers. Because we study the demand reaction to price changes, our target period starts in the same month that the price change becomes effective. It follows that the different feature sets reference different times. Both time-series and customer-based features represent information at the time of prediction (t_p)—that is, at the *start* of the lead time. In contrast, the planned price change is known in advance and so price-based features refer to the actual time of the price change ($t_c = t_p + lt$), which corresponds to the *end* of the lead time. Figure 4.3 presents the timeline of a sample price change made in January 2016.

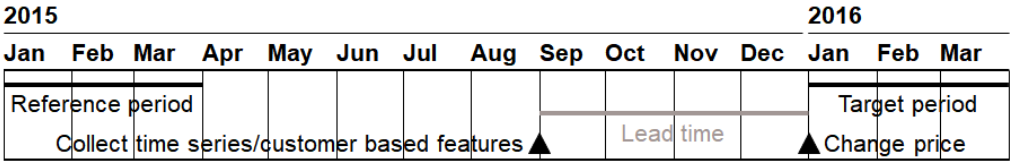

Figure 4.3: Timeline of prediction approach (typical of a price change in January 2016)

We define the responsive variable as a ratio: total demand (D) of the demand volumes (d_i) in the three months of the target period *divided by* total demand in the same months in the prior year (d_{i-12}):

$$D = \frac{\sum_{i=t_p}^{t_p+2} d_i}{\sum_{i=t_p-12}^{t_p+2-12} d_i}. \tag{4.7}$$

Thus, the minimum value of the dependent variable D_p is 0 for a customer that switches all of its demand, but it is greater than 1 when demand increases in comparison with the previous year.

The training and validation data sets are created as follows. For each month (t) in the reference period (June 2015 to January 2019), we select all customers (c_i) for which the price changed in the focal month. Altogether, there are 7,400 observations in our training data set and 2,300 observations in our validation data set.

4.5 RESULTS

In this section, we first assess whether price information increases the forecast accuracy of our tested demand prediction models. Next we identify the price features most predictive of customer-specific price sensitivities. We then present results of the price sensitivity prediction, including the revenue potential when the pricing approach is aligned with calculated price sensitivities.

4.5.1 *Results of the price feature importance assessment*

We distinguish among three different feature sets—customer-based data (cust), demand time-series–based features (ts), and price-related features (price)—and test the random forest demand prediction model's accuracy with different combinations of these sets. For comparison purposes, we also test the accuracy of a linear regression, a naïve prediction (i.e., one that assumes demand is constant, $D = 1$, for all observations), and an ARIMA model (see Section 4.4). The accuracy measures of the random forest model (with different feature sets) are given in Table 4.4 and those of the statistical models are given in Table 4.5.

The model with all three feature sets (cust_price_ts) leads to the best predictive performance (MAE of 0.208 and RMSE of 0.306). The Wilcoxon signed-rank test shows that the MAE and RMSE results for the cust_price_ts model differ significantly from those of all other models at the 95% confidence level. Regardless of which feature sets are used, we find that the linear regression, the

Table 4.4: Accuracy measures used to assess demand prediction by random forest models

cust	price	ts	MAE	RMSE
x	–	–	0.225	0.327
–	x	–	0.243	0.344
–	–	x	0.214	0.315
x	x	–	0.221	0.321
x	–	x	0.212	0.313
–	x	x	0.210	0.309
x	x	x	0.208	0.306

Note: x = included in feature set combination, – = excluded from feature set combination.

Table 4.5: Accuracy measures used to assess demand prediction by statistical models

Model	MAE	RMSE
Linear regression	0.492	1.003
ARIMA	0.278	0.634
Naïve	0.248	0.354

ARIMA forecast, and the naïve model perform significantly worse than do all random forest models. In general, we observe that both MAE as well as RMSE measures are quite high if the result is meant to be used to predict future demand levels. It seems that the features used for the analysis are not sufficient to fully explain the change in demand. There might be other factors that are not observable for the research partner (e.g., prices of competitors) that significantly affect the demand of their customers. In our case, we are only interested to learn about the price sensitivity of customers, measures in their reaction to price changes. Thus, the exact demand size is not important and the results are considered to be sufficient.

Because we use the same data for the churn prediction approach, we also test that model with different data inputs. We find that the model with price features (cust_price_ts model) works significantly better than the one without them (cust_ts model): the AUC-ROC increases from 0.789 to 0.823 and the AUC-PRC from 0.356 to 0.368. The increase in forecast accuracy is even greater for

models that incorporate data-balancing methods; in a model with upsampling, for example, the AUC-ROC increases from 0.779 to 0.825 and the AUC-PRC from 0.303 to 0.361.

We assess the relative importance of predictive features according to the approach described in Section 4.3.1. For our cust_price_ts model, the results—which are scaled from 0 to 100—are reported in Table 4.6.

Table 4.6: Feature importance: Top 20 features of the cust_price_ts model

Feature	Importance
price_change_percent	100
arima	53.40
Q_last_volume_v_av	31.99
fluc_3m	29.88
fluc_2m	29.38
fluc_1m	26.83
Q2last_volume_v_av	26.79
total_price_change_percent	23.46
M_last_volume_v_av	23.44
last_demand2	23.28
revenue_Last_year	22.98
share_product_x	22.34
last_demand4	22.01
cv2	21.94
contractual_quantity	21.14
cv2_12m	20.56
vol_service_x	20.50
price_before	20.46
vol_before	20.31
avg_price_vol_cluster	19.91

The actual price change (price_change_percent) is the most important feature in the random forest model, and it is far more important than the next-listed feature (the ARIMA forecast). This forecasting predominance of the actual price change confirms the strong effect of price changes on subsequent demand. In contrast, the difference in features' relative importance between the ranks of 3

and 20 is quite small. Besides the actual price change, several other price-related features make this list: the total price change since 2012 (ranked 8th), the actual price before the focal price change (ranked 18th), and the price compared with that for customers exhibiting similar demand volumes (ranked 20th). Most of the listed non–price-related features (e.g., Q_last_volume_v_av) are based on the demand time series.

4.5.2 *Classifying customers by their price sensitivity*

We use the AUC-ROC and AUC-PRC measures to compare the classification performance of various models for predicting price sensitivity. The results are presented in Table 4.7. (In Tables 4.7–4.11, "Sens pred (class)" = price sensitivity prediction based on the classification (grouping) of customers and "Sens pred (reg)" = price sensitivity prediction based on regression.)

Table 4.7: Classification performance by type of approach

Approach	AUC-ROC	AUC-PRC
Contract length	0.505	0.513
Demand volume	0.578	0.575
Price simulation	0.562	0.594
Churn prediction	0.652	0.663
Sens pred (class)	0.744	0.728
Sens pred (reg)	0.736	0.717

We can see that the contract length approach (AUC-ROC $= 0.505$) is really no better than random guessing (AUC-ROC $= 0.5$). Also, the demand volume and the price simulation approaches are only slightly better; their AUC-ROC values are (respectively) 0.578 and 0.562. The two direct methods for predicting price sensitivity exhibit the best predictive performance, with AUC-ROC values of 0.744 and 0.736. Hence classification and regression are the preferred methods for distinguishing between customers with high and low price sensitivities. We obtain the same results when using 2018 data (instead of 2019 data) in the validation data set; this outcome establishes that the effectiveness of these direct prediction models is not time dependent.

To identify the features most useful for distinguishing between customers whose price sensitivity is high or low, we perform an ex post analysis and compare—separately for customers that face small and large price changes—

the differences in mean feature values between customer groups with high versus low price sensitivities. Regardless of sensitivity to prices, a customer's profitability is (from the service provider's perspective) one of the most important features. On average, customers with low price sensitivities have higher contribution margins than do customers that are highly price sensitive; hence customers that previously negotiated lower prices are more price sensitive. The same relation holds for the number of additional services (e.g., preferred day delivery, insurance, cash on delivery) offered by our research partner: on average, customers with low price sensitivities order more additional services. There is one difference between the customer groups facing different price change levels. Namely, recent demand trends play a role *only* in outcomes for the group that faces large price changes. This means that customers with a positive (increasing) demand trend are less likely to reduce their demand after a price change and thus are less price sensitive. Although we have identified correlations between feature values and the classification results, these are not causal relationships and are not even the same for each individual customer. We test this claim for the profitability feature and also for the additional services; using each of these features for the classification approach yields AUC-ROC values of 0.49 and 0.52, respectively. Therefore, these correlations are not an effective means for distinguishing between customers of high and low price sensitivity.

4.5.3 *Calculating the financial impact of predictions*

We present results of the price sensitivity analysis in Tables 4.8 and 4.9. For each of the prediction approaches, we compare the mean demand change for the two groups of customers—those classified as having low or high price sensitivities—and then test for whether (or not) the difference between these groups is statistically significant. The tables report the number of observations (N) and the average demand change that follows the price adjustment (Mean) as well as the standard deviation of the demand changes (SD), the difference in the mean demand change between the customer groups with low versus high predicted price sensitivites (Mean difference), the degrees of freedom (df), the t-statistic (t) and the p-value (p), and the lower and upper boundaries of the 95% confidence interval (CI).

Table 4.8: Changes in demand of customers after *small* price changes

									95% CI	
Approach	Sens	N	Mean	SD	Mean difference	t	df	p	Lower	Upper
Contract length	Low	548	1.03	0.36	0.006	0.311	1,101	0.756	−0.032	0.045
	High	582	1.02	0.29						
Demand volume	Low	606	1.06	0.30	0.065	3.236	1,004	0.001	0.025	0.104
	High	524	0.99	0.37						
Price simulation	Low	524	1.07	0.33	0.074	3.782	1,105	0.000	0.036	0.113
	High	606	0.99	0.33						
Churn prediction	Low	828	1.08	0.31	0.201	9.013	500	0.000	0.157	0.245
	High	302	0.88	0.34						
Sens pred (class)	Low	599	1.14	0.33	0.247	13.562	1,124	0.000	0.211	0.283
	High	528	0.90	0.28						
Sens pred (reg)	Low	558	1.15	0.34	0.243	13.210	1,066	0.000	0.207	0.279
	High	572	0.91	0.27						

Note: CI = confidence interval, df = degrees of freedom, SD = standard deviation.

Table 4.9: Changes in demand of customers after *large* price changes

									95% CI	
Approach	Sens	N	Mean	SD	Mean difference	t	df	p	Lower	Upper
Contract length	Low	582	0.97	0.40	0.043	2.043	1,021	0.041	0.002	0.084
	High	548	0.93	0.30						
Demand volume	Low	524	0.99	0.34	0.082	3.969	1,123	0.000	0.042	0.123
	High	606	0.91	0.36						
Price simulation	Low	606	0.98	0.33	0.064	3.016	1,046	0.003	0.022	0.105
	High	524	0.92	0.38						
Churn prediction	Low	302	1.06	0.28	0.150	7.252	690	0.000	0.109	0.190
	High	828	0.91	0.37						
Sens pred (class)	Low	526	1.08	0.33	0.251	12.796	1,094	0.000	0.212	0.289
	High	598	0.83	0.32						
Sens pred (reg)	Low	572	1.07	0.34	0.252	12.867	1,125	0.000	0.213	0.290
	High	558	0.82	0.32						

Note: CI = confidence interval, df = degrees of freedom, SD = standard deviation.

For the group of customers presented with small price changes (6% increase, on average), there is a significant difference in demand change—between customers considered to have high versus low price sensitivity—*except* when the contract length classification method is used ($p = 0.756$). The table's reported values show also that the mean difference is greatest for direct price sensitivity predictions (via classification or regression) and for the churn prediction; in the latter, the demand of customers with low price sensitivities increases by 7% to 15% whereas demand of those with high price sensitivities decreases by more than 9%.

The average decline in demand of customers receiving large price increases (of 13%, on average) is greater, and so demand during the target period is lower, than of customers receiving small price increases. The difference in demand change—between customers with low versus high price sensitivities—is greatest for the direct methods (classification and regression) of predicting price sensitivity (mean difference $= 0.25$), followed by the churn prediction approach (mean difference $= 0.15$). For all prediction approaches, the mean difference is significant at the 95% confidence interval. However, the mean difference for the contract length approach is not significant at the 99% confidence interval.

In short, there is a significant difference in the demand change of customers depending on whether or not they are predicted to be highly price sensitive. This effect is observed irrespective of whether the price change is small or large. The mean difference is greatest for the direct price sensitivity prediction and churn prediction methods. In contrast, there is no significant difference between customer groups whose price sensitivities are based on contract length.

To assess the potential financial benefits from aligning price changes with predicted price sensitivities, we focus on two groups of customers: those that receive large price adjustments and are predicted to be highly price sensitive (group A), and those that receive small price adjustments and are predicted to have low price sensitivity (group D). We compare the former group with customers that are also presumed to be highly price sensitive but that receive small price adjustments and compare the latter group with those that are also classified as not being price sensitive but receive large price adjustments. For both comparisons, we again check the difference of the demand change that follows the respective price change and test whether that difference is significant. We present the results for customers with high and low predicted price sensitivities in (respectively) Table 4.10 and Table 4.11. For each prediction approach, we distinguish (in the "Price" column) between customers that receive large versus small price changes.

Table 4.10: Changes in demand of customers with *high* predicted price sensitivities

									95% CI	
Approach	Price	N	Mean	SD	Mean difference	t	df	p	Lower	Upper
Contract length	Low	582	1.02	0.29	0.094	5.314	1,127	0.000	0.059	0.129
	High	548	0.93	0.30						
Demand volume	Low	524	1.06	0.30	0.062	3.677	1,101	0.000	0.037	0.122
	High	606	0.99	0.34						
Price simulation	Low	606	0.98	0.33	0.068	3.590	1,050	0.000	0.034	0.118
	High	524	0.91	0.38						
Churn prediction	Low	302	0.88	0.34	−0.031	−1.328	575	0.185	−0.077	0.015
	High	828	0.91	0.37						
Sens pred (class)	Low	599	0.90	0.28	0.064	3.561	1,124	0.000	0.029	0.099
	High	526	0.83	0.32						
Sens pred (reg)	Low	572	0.91	0.27	0.084	4.743	1,097	0.000	0.049	0.118
	High	558	0.82	0.32						

Note: CI = confidence interval, df = degrees of freedom, SD = standard deviation.

Table 4.11: Changes in demand of customers with *low* predicted price sensitivities

									95% CI	
Approach	Price	N	Mean	SD	Mean difference	t	df	p	Lower	Upper
Contract length	Low	548	1.03	0.36	0.057	2.516	1,105	0.012	0.013	0.102
	High	582	0.97	0.40						
Demand volume	Low	606	1.06	0.30	0.062	3.286	1,052	0.001	0.025	0.099
	High	524	0.99	0.34						
Price simulation	Low	524	1.07	0.33	0.087	4.425	1,103	0.000	0.048	0.125
	High	606	0.98	0.33						
Churn prediction	Low	828	1.08	0.31	0.021	1.047	588	0.295	−0.018	0.059
	High	302	1.06	0.28						
Sens pred (class)	Low	526	1.14	0.33	0.061	3.063	1,103	0.002	0.022	0.099
	High	599	1.08	0.33						
Sens pred (reg)	Low	558	1.15	0.34	0.075	3.716	1,127	0.000	0.036	0.115
	High	572	1.07	0.34						

Note: CI = confidence interval, df = degrees of freedom, SD = standard deviation.

The results for customers that are predicted to have high price sensitivities (Table 4.10) establish that the demand change difference between customers receiving small versus large price adjustments is significant for all prediction methods—except for the churn prediction approach. The mean difference is greater than 0.06 for all other prediction methods, and it exceeds 0.08 for the contract length approach and the sensitivity regression prediction.

For customers predicted to have low price sensitivities (Table 4.11), the mean difference in the demand change between groups is insignificant only when the churn prediction approach is used. With all other prediction methods, the difference in demand change between customers that received small versus large price changes is greater than 0.06. This difference is greatest for the price simulation approach (0.087), followed by the regression-based direct sensitivity prediction (0.075).

Combining the results of assessments according to price change levels and to predicted price sensitivity levels, we find the greatest between-group differences in demand change when using the prediction approach based on price sensitivity regression. Among all the tested approaches, this one most accurately distinguishes between customers with low and high price sensitivities. The other tested models—excepting the contract length and churn prediction approaches—also yield significant (albeit smaller) differences between the customer groups. The churn prediction approach reveals a correlation between price sensitivity customer classes but no difference between customers subject to small versus large price changes. It seems that customers with extremely low or extremely high predicted churn probabilities are unconcerned about the price change level. The latter group may already have decided to use other providers (i.e., independently of the price change), and the former group is relatively insensitive to price because they demand the same volumes regardless of whether the price adjustment is small or large. Hence firms in this group can be financially exploited by a supplier that institutes a large price increase only for customers with low predicted churn probabilities.

We calculate the potential financial gain—from adopting the prediction approach of price sensitivity regression—when all customers with high (resp. low) predicted price sensitivities are given small (resp. large) price adjustments. Thus we combine the average difference between the price changes (from 6% to 13%) and the average difference between the demand changes (from 1.07 to 1.15 for customers with high price sensitivites and from 0.82 to 0.91 for those with low price sensitivities) to calculate the potential new revenue for both customer groups: the group of customers with low predicted price sensitivites and previously small price adjustments and customers with high predicted price

sensitivites and previously large price adjustments. In the first case, we find no change in revenue. In the second case, however, there is a 2% increase that would amount to some €1.5 million in additional annual revenue (calculated by comparing the expected revenue based on the presumed price change and the expected demand change due to the predicted price sensitivities to the actual revenue based on the actual price adjustment and respective demand change).

The potential revenue gain can be increased still further if we split our customers into quartiles (for both price change and predicted price sensitivity) and then focus on the first and the fourth quartiles. So under this approach we consider only customers that are in quartiles with the largest (17% on average) and smallest (5% on average) price adjustments and are also in quartiles with the highest and lowest predicted price sensitivities. We again calculate the potential revenue gain and identify a revenue increase of 4% for customers with previously small price adjustments and low price sensitivities and an increase of 18% for customers with previously high price adjustments and high price sensitivities. In this case, the annual revenue potential totals more than €4 million.

Finally, we test the financial impact of basing price change levels not on the first and last quartile of customers but use the average price increases of all four quartiles as potential options. We find that these additional options do not increase the revenue potential, from which it follows that one need only consider the two extreme cases of customers in the first and last quartile of price change levels.

4.6 MANAGERIAL IMPLICATIONS

The results of our analysis show the effect of price changes on customer demand as well as the large variance—between customers with low and high price sensitivities—in these demand changes. The prediction method we propose for distinguishing between customers with low and high price sensitivities is easy to implement and so could be used immediately by our research partner. Even though price experiments were not considered in the past, this prediction approach enables them because it significantly reduces the risk of losing customers. Results of the classification performance measure confirm the model's high level of accuracy. Our prediction approach thus allows for effectively selecting between small and large price adjustments for different customer groups and hence for assessing the demand effects of various price change levels. Our research partner anticipates undertaking an A/B test to

validate our results and to generate more observations that can be used for further training of (and for making additional improvements in) the model. The more diverse are the training set's price changes, the more accurate will be the model's predictions.

Price adjustments can be fine-tuned by testing price change levels below and above the currently considered levels for customers with, respectively, the lowest and highest predicted price sensitvities. For that purpose, the lower and upper 5% or 10% of customers could be targeted with price changes that are more extreme than the current average. Such adjustments could become critical given the increasing cost pressure in the parcel shipping sector, a result of higher costs for "last mile" delivery (Thiele and Dieke 2018). These increased costs can strain a firm's current prices, which may need to be raised so that earnings do not decline. This consideration magnifies the importance of setting optimal price levels and of thereby remaining competitive with rivals—and *not* losing customers because of price increases that are too extreme.

4.7 CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Our research documents that price changes affect customer demand in a competitive B2B service environment. Because price adjustments can lead to partial defection, it is essential for the firm to identify which customers are price sensitive—and which are not—and then to align its pricing strategy with those predictions. We find that price features increase the accuracy of predictions regarding both demand and churn. After testing several methods of predicting price sensitivity that do not rely on price experiments, we conclude that the most effective approaches are those that use classification or regression to distinguish between customers with low and high price sensitivities. The classification performance of these two direct models is significantly better than that exhibited by other prediction methods. This result is time independent, which we prove by using different splits of the time-series data.

The prediction approach proposed here is applicable whenever sufficient historical data—with different levels of price adjustments—are available. So even if price experiments are an option, our data-driven approach is a good starting point from which to distinguish between customers with low and high sensitivity to prices; that information can then serve as input for an A/B test.

We also find that, as compared with highly price-sensitive customers, those with low price sensitivites have (on average) the following characteristics: they order more additional services and have higher contribution margins; when

given large price changes, they also have a more positive demand trend. In lieu of price experiments, we use a simulated A/B test with different price changes to illustrate the effectiveness of our method and to calculate its financial potential. This approach works well because of the high variance in the validation data set's actual price change levels. Thus one can apply this method *before* initiating any price experiments, which run a nonnegligible risk of losing customers. Considerable financial gain is possible when prices are aligned with customers' price sensitivity. In our case, the expected annual revenue increase is €4 million—even though this estimate is based only on customers that received a price adjustment in January 2019. The actual potential is much higher, since two thirds of all price adjustments occur in the rest of the year.

The insights gained from this research paper suggest that scholars could profit from investigating optimal price adjustment levels for price-sensitive and price-insensitive customers. Here we have split the customers into two groups based on the median of the actual price change levels; we then used the mean of the actual price changes given to the two groups as the two possible price change levels that customers could face. Thus we have *not* tested whether revenue gain could be increased by setting some other (lower or higher) price level.

At this point we can suggest two alternative methods for testing other price change levels: (i) an analytical approach that requires only a large enough validation data set; and (ii) a method that builds on price experiments. First, one can increase the number of customers in the validation data set and then split those customers into more than four groups (e.g., into deciles) based on the actual price changes they were given; then the optimal price change can be identified at a more granular level. A disadvantage of this approach is that one can test only those price change levels that have previously been used. Second, one way to overcome that limitation is by conducting A/B tests. The more of such tests that are carried out and the more diverse are the price changes in them, the better a model learns individual demand reactions and so the more its predictive accuracy improves. Of course, this approach risks—indeed, it practically guarantees—losing customers whose willingness to pay is exceeded by the changed price.

5

CONCLUSION AND OUTLOOK

5.1 CONCLUSION

The application of *data analytics* in the context of *supply chain management* and in the *logistics industry* is very diverse and covers different stages of the value chain as well as different industries. Although the potential of analytics capabilities is well known, many companies still struggle to identify appropriate use cases and to estimate the financial potential from using this technology. In academia, the topic became much more popluar in recent years, yet many research gaps remain. Also, many researchers focus on theoretical contributions, neglecting the implementation in practice.

This dissertation aims to contribute to a better understanding of data analytics with regards to specific needs such as *interpretability, calculation efficiency, data availability* and *impact analysis*. To this end, the three underlying research projects cover typical challenges in the field of data analytics. All three projects have been conducted in cooperation with *industry research partners* who provided empirical data for the analysis. The main findings from the projects are threefold:

First, the tested *machine learning-based* application methods outperform traditional statistical methods in most cases. However, the best performing method differs for each data set and thus it is challenging to identify universally valid recommendations which prediction method to use in which application area or which industry. To overcome this challenge, *meta learning methods* can be used that either learn to combine several different prediction methods or to select a specific method for a specific data set. Thereby, one can either further increase the predictive performance in case of learned combinations or significantly decrease the calculation time in case of the method selection.

Second, machine learning methods are known to be *black-box* models. But employees who use the output of these models, often want to understand the reason for a specific prediction to trust the data input they receive. Therefore, *interpretability* is a key factor to consider in many application areas of prediction models in the industry. Due to the fact that interpretable models often come with lower accuracies compared to more complex black-box models, there is a trade-off between interpretability and predictive performance. However, there

are approaches to make *black-box* models interpretable and thus allow insights how specific predictions materialized.

Third, the value of existing data is often not clear and the risks associated with new data acquisition can be high. In cases such as churn mitigation actions or pricing decisions, randomized experiments can cause harm, because one might loose customers. It is important to consider such risks in the planning phase of analytics use cases and to decide whether to take the risk or to find alternative data sources. Past data observations can eventually contain sufficient information to train models without the need for randomized experiments.

5.2 CONTRIBUTIONS TO THEORY AND PRACTICE

We[1] contribute to the existing literature and the current discussion around data analytics in forecasting and planning by analyzing three main challenges in the domain: The need for *computationally efficient predictions*, the request for *interpretability* and the requirement to run algorithms with existing, historical data *without the need to run risky experiments*. All our findings are based on empirical research in collaboration with *research partners from the industry*. Thus, we also consider the practical requirements, implications and advantages through the proposed models. Following the structure of this dissertation, the contributions are summarized along the three case studies.

In Chapter 2, we apply meta learning methods in the field of *intermittent demand prediction* with the ambition to improve warehouse operations. We compare a variety of different prediction methods including data pre-processing through time series decomposition and data clustering. The results show that forecast combinations increase the predictive performance but come at the cost of long calculation times, because many models need to be trained in parallel. As an alternative, we apply a method selection scheme that performs as good as the combinatorial methods but makes predictions much faster, because only the selected prediction method needs to be trained. Thus, we combine efficiency with predictive performance. The results of a warehouse simulation show that the proposed approach leads to yearly savings of 3% of the warehouse operation costs. The approach is easily implementable and does not require any change in the warehouse setup.

[1]The term "we" refers to the authors of the respective chapters as denoted at the beginning of each chapter. For the conclusion, this refers to the authors of Faber and Spinler (2019a,b,c).

In Chapter 3, we predict *partial defection* of customers in a business-to-business market. With our analysis, we broaden the scope of previous research to cover partial defection in business-to-business relations that are characterized by continuous service delivery and individually negotiated contracts. The competition in the market is high and customers can easily shift demand to competitors. We compare a set of different prediction models including machine learning and probability based methods. The ambition is to combine high predictive performance, maximization of profits and interpretability to understand the case-specific reasons for customers to partially defect. We find that directly interpretable models have significantly lower predictive accuracy compared to more complex, black-box models. Therefore, we use a data permutation approach to make the better performing methods interpretable. In addition, we use sensitivity analysis to measure the impact from different partial defection threshold values and different prediction lead times. Financial measures guide the choice of a combination of threshold level and lead time that maximizes profit, after retention actions, while accommodating such company-specific business requirements as the capacity to undertake retention actions. The results show that the research partner can increase its profit (after retention actions) by more than €500,000 in the short term.

In Chapter 4, we predict *price sensitivities* of customers to select the right price change level for each of them. Unlike previous research, we do not carry out any price experiments but fully rely on existing data from past price changes and the following adjustments to demand. This is, because in the case of price experiments the risk of losing customers is too high. With the existing data, we simulate a random experiment to identify which customers are little and which ones are highly price sensitive. We train a model to predict the expected price sensitivity for each customer. To validate the findings, we compare the actual demand change of the customers in focus. We find that an approach that directly predicts the customer specific price sensitivity works best whereas we do not find any correlation between the contract length or the demand volume and the actual price sensitivity. The results indicate a potential revenue increase of €4 million per annum.

5.3 AVENUES FOR FUTURE RESEARCH

This dissertation covers different applications of data analytics in forecasting and planning. Although, the research reveals interesting and valuable insights,

avenues for future research exist. Following these calls for further research may lead to new insights.

First, Chapter 2 covers intermittent demand prediction and warehouse operation optimization. The existing rolling one-period forecast can be extended to a multiple-period forecast. Thereby, the costs of stock transfers over multiple periods can be taken into consideration to optimize the warehouse operations. Other than that, the multiple-period forecast can be used to plan order volumes or to optimize the warehouse layout. Furthermore, additional information can be added to further increase the predictive performance. In addition to the used time series data, price information, weather data or behavioral data from the website can be leveraged.

Second, Chapter 3 contains a partial defection prediction approach that combines high predictive accuracy with interpretability and profit optimization. In a next step, judgemental information from the sales agents regarding the last customer contacts can be added to the feature space to further improve the predictive performance. In addition, information regarding the actual defection reason can help to tailor different retention actions. Also, different prediction models can be trained to make forecasts for different defection causes. A better understanding of different retention actions and the respective costs and success rates might further contribute to improve the model. Also, an assessment of the retention success rates for different prediction lead times would allow to choose the best suitable lead time.

Third, Chapter 4 is about price discrimination to distinguish between little and highly price sensitive customers. With this regard, the proposed approach can be used as an input for price experiments to validate the findings and to generate additional data that can be used to further improve the model. In addition, it would be helpful to not only classify customers into two groups based on their expected price sensitivities but to better understand which price levels to choose for which customer group. A larger data set may allow to use more classes of different price changes. Thereby, a more diverse choice of potential price change levels would exist.

In conclusion, the research areas of *method selection, model interpretability* and *data driven analysis* without risky experiments should all be further explored as they deal with major challenges in the field of data analytics. The further rise of analytics makes it more important to use efficient prediction methods and to understand the outcome. Also, the use of existing data with minimal additional data collection is essential, because the latter is time consuming and sometimes difficult to perform. Apart from the specific topics covered in the dissertation, there are numerous other fields for further research in the field of data-driven

forecasting and planning. With regards to supply chain management and demand predictions, other application areas cover procurement or production. In the logistics industry, routing and capacity planning as well as predictive risk management are potential application areas.

A

APPENDIX TO CHAPTER 2

A.1 HYPERPARAMETERS FOR MACHINE LEARNING METHODS

The hyperparameters we tested for machine learning methods are listed in Table A.1. The choice of these hyperparameters was based on a literature review that revealed the parameter values most commonly used in intermittent demand prediction. We use a fivefold cross-validation approach with grid search to identify the hyperparameters that work best.

Table A.1: Hyperparameters for machine learning methods

Method	Parameters
SVM linear	Cost: 1
SVM radial	Cost: 0.25, 0.5, 1
	Sigma: automatic sigma estimation (*sigest*)
RF	Randomly selected predictors: $2, p/2, p$ (p = number of features)
FFNN	Weight decay: 0.0001, 0.001, 0.1
	Neurons: 1, 3, 5, 10
BRNN	Neurons: 1, 2, 3
GBM	Maximum tree depth: 1, 2, 3
	Boosting iterations: 50, 100, 150
	Shrinkage: 0.1
	Minimum terminal node size: 10

A.2 TIME-SERIES FEATURES FOR CLUSTERING AND METHOD SELECTION META LEARNING

The time-series features used for clustering and for our method selection approach are given in Table A.2. The "Decomp." column indicates whether (or not) the feature is also calculated for the trend- and seasonality-decomposed time series.

Table A.2: Features for time-series clustering and meta learning using method selection

Feature	Description	Decomp.
CV	Coefficient of variation	Yes
ADI	Average inter-demand interval	No
Mean	Mean demand	Yes
Mean_52w	Mean demand of last 52 weeks	No
Nonzero	Number of nonzero-demand periods	No
Skewness	Symmetry measure	Yes
Kurtosis	Peak measure	Yes
Breakpoints	Number of breakpoints in the time series	Yes
Teraesvirta	Neural network test for nonlinearity	Yes
Hurst	Exponent for self-similarity (long-range dependence)	Yes
Serial-corr	Measurement of the serial correlation of the time series	Yes
Trend	Measurement of the trend	No
Seasonality	Measurement of the seasonality	No

Coefficient of variation and *average inter-demand interval* measure the variation in demand height (CV) and the regularity of a demand (ADI: average interval between two periods of nonzero demand).

Mean and *Nonzero* are general descriptive statistics that indicate the average value and the number of zero demand values in a time series.

Skewness describes the symmetry of a data set. Negative (resp. positive) values indicate skewness to the left (resp. right). The normal distribution's skewness is 0.

Kurtosis characterizes the tail extremity of a distribution as compared with the normal distribution. High values of kurtosis reflect the existence of outliers.

Breakpoints are indicators of unexpected shifts in a time series. We apply the *strucchange* package in R that uses dynamic programming to find breakpoints. The Bayesian information criterion (BIC) is used to find an optimal model with a minimal residual sum of squares.

The Teraesvirta neural network test for nonlinearity is used to test the nonlinearity of a time series.

The Hurst exponent provides information about the long-term self-similarity of a time series. When this parameter equals 0.5, the time series is considered to be a geometric random walk; a smaller (resp. larger) exponent corresponds to a mean-reverting (resp. trending) series.

Serial correlation (a.k.a. autocorrelation) reflects the stationarity of a time series. The Box–Pierce statistic is a well-known measure that combines the autocorrelation for different time lags.

Trend and *seasonality* are indicators of the magnitude of trend and seasonality patterns in a time series. First, the focal time series is decomposed using the seasonal and trend decomposition using Loess (STL) approach. Then the standard deviation of the non-decomposed time series is divided by the standard deviation of (respectively) the de-trended or the de-seasonalized time series.

In Table A.3 we present, for all three data sets, the feature ranking used during method selection. We report the ranking for the best-performing method selection approach—that is, the ranking prediction. The table contains no rank for the trend, seasonality, or decomposed features for the Royal Air Force (RAF) data because there are so few observations in that data set that reflect seasonality or trend patterns.

Table A.3: Ranking of features used in the ranking method selection approach

Feature	Research partner data	RAF data	Simulation data	Average rank
trend	1		4	2.50
seasonality	2		3	2.50
skew_decomp	5		2	3.50
mean_12m	3	8	5	5.33
cv2	7	3	7	5.67
stdev_12m	4	4	10	6.00
stdev	9	1	14	8.00
mean	13	2	16	10.33
stdev_decomp	6		15	10.50
hurst	12	15	8	11.67

A.3 WAREHOUSE OPERATION COSTS ALGORITHM

Our simulation seeks to capture the circumstances of the online retailer, so we start by using its current cost factors for picking and stock transfers as well as its current storage space restriction for the shelf-picking area. Then we relax those assumptions and calculate the costs for different scenarios with different cost factors and storage space allocations. In this way we aim to generalize our findings and ensure that the proposed forecast approach is sufficiently robust to work well for other warehouse setups.

Thus our first simulation is based on the following costs, which are supplied by the online retailer. The picking costs are €0.1 per item in the shelf-picking area and €0.4 for items stored in the high-bay warehouse, where the higher cost for picks from the high-bay warehouse are due to the time needed for repackaging. It costs €0.6 per product for a stock transfer from the high-bay warehouse to the picking area and €1.5 for a stock transfer from the picking area to the high-bay warehouse; the latter's higher costs stem from the costs for new cardboard boxes in which the items are stored. We use the same cost factors for all products because the products (mostly books or boxes with children's toys) are similar to each other and so picking times are comparable across items. The capacity of the shelf-picking area is 10% of the retailer's overall storage capacity. In subsequent simulations we change the shelf-picking area's proportion (from 10% to 5%, 20%, and 30% of overall capacity) and also change the ratio of stock transfer costs (from a ratio of 15:1 to 8:1, 4:1, and 1:1).

The algorithm we use to calculate warehouse operation costs is reproduced in Figure A.1. The indicator variables $\mathbb{1}_{storage\_shelf}$ and $\mathbb{1}_{storage\_wh}$ are set to 1 if the focal stock keeping unit (SKU) was stored *either* in the shelf-picking area ($\mathbb{1}_{storage\_shelf}$) or the high-bay warehouse ($\mathbb{1}_{storage\_wh}$) in the preceding period; otherwise, those indicators are set to 0.

for $m \leftarrow 1, M$(number of prediction methods) **do**

  for $n \leftarrow 1, \mathrm{Ntext}(\mathrm{numberofperiods})$ **do**

    for $s \leftarrow 1, S$(number of SKUs) **do** ▷ For each SKU, calculate costs for both areas

      $Cost\_wh_{m,n,s} \leftarrow forecast_{m,n,s} * pick\_wh + transfer_{shelf\_wh} * \mathbb{1}_{storage\_shelf,m,n-1,s}$

      $Cost\_shelf_{m,n,s} \leftarrow forecast_{m,n,s} * pick\_shelf + transfer_{wh\_shelf} * \mathbb{1}_{storage\_wh,m,n-1,s}$

      $Cost\_difference_{m,n,s} \leftarrow Cost\_wh_{m,n,s} - Cost\_shelf_{m,n,s}$

    end for

    Sort s according to $Cost\_difference_{m,n}$ in decreasing order ▷ Assign SKUs with highest cost difference to shelf-picking area; assign other SKUs to high-bay warehouse

    for $s \leftarrow 1, capacity_{shelf}$ **do**

      $\mathbb{1}_{storage\_shelf,m,n,s} \leftarrow 1$

      $\mathbb{1}_{storage\_wh,m,n,s} \leftarrow 0$

    end for

    for $s \leftarrow capacity_{shelf} + 1, S$ **do**

      $\mathbb{1}_{storage\_shelf,m,n,s} \leftarrow 0$

      $\mathbb{1}_{storage\_wh,m,n,s} \leftarrow 1$

    end for

    for $s \leftarrow 1, S$ **do** ▷ Calculate picking and replenishment (transfer) costs for each SKU

      $Cost\_pick_{m,n,s} \leftarrow forecast_{m,n,s} * pick\_wh * \mathbb{1}_{storage\_wh,m,n,s} + forecast_{m,n,s} * pick\_shelf * \mathbb{1}_{storage\_shelf,m,n,s}$

      $Cost\_transfer_{m,n,s} \leftarrow transfer_{shelf\_wh} * \mathbb{1}_{storage\_shelf,m,n-1,s} + transfer_{wh\_shelf} * \mathbb{1}_{storage\_wh,m,n-1,s}$

    end for

  end for ▷ Calculate total costs for selected method across all SKUs and all periods

  $Total\_cost_m \leftarrow \sum_{n=1}^{N} \sum_{s=1}^{S} Cost\_pick_{m,n,s} + Cost\_transfer_{m,n,s}$

end for

Figure A.1: Algorithm for calculating warehouse operation costs

B

APPENDIX TO CHAPTER 3

B.1 PREDICTIVE PERFORMANCE OF CHURN PREDICTION MODELS WITH FEATURE SELECTION

Table B.1: Predictive performance of all models: Boruta feature selection (column rankings in parentheses)

	Balancing	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
LogR	orig	0.723 (23)	0.288 (21)	2.879 (18)	562,362 (10)	678,027 (20)
	weights	0.738 (12)	0.292 (17)	2.879 (18)	491,978 (26)	672,753 (21)
	down	0.729 (21)	0.279 (24)	2.820 (23)	502,286 (24)	652,333 (28)
	up	0.736 (15)	0.291 (18)	2.857 (20)	489,008 (28)	669,580 (22)
	smote	0.738 (13)	0.299 (15)	2.937 (14)	517,047 (23)	667,416 (23)
RF	orig	0.745 (10)	0.346 (4)	3.244 (3)	557,514 (13)	697,883 (7)
	weights	0.736 (14)	0.346 (2)	3.273 (1)	546,516 (18)	692,148 (15)
	down	0.748 (9)	0.305 (14)	2.901 (17)	549,734 (15)	695,321 (10)
	up	0.733 (18)	0.308 (13)	3.032 (12)	537,977 (20)	694,611 (14)
	smote	0.756 (5)	0.326 (8)	3.091 (7)	565,918 (8)	702,695 (4)
NN	orig	0.733 (16)	0.290 (20)	2.798 (24)	490,881 (27)	638,148 (29)
	weights	0.758 (3)	0.310 (12)	3.018 (13)	565,937 (7)	695,197 (11)
	down	0.721 (24)	0.285 (23)	2.915 (16)	532,108 (21)	691,637 (16)
	up	0.697 (26)	0.275 (26)	2.798 (24)	530,465 (22)	684,908 (18)
	smote	0.731 (19)	0.290 (19)	2.828 (22)	538,894 (19)	695,433 (9)
GBM	orig	0.758 (2)	0.339 (5)	3.061 (10)	546,674 (17)	695,649 (8)
	weights	0.757 (4)	0.346 (3)	3.142 (4)	590,807 (2)	712,026 (2)
	down	0.760 (1)	0.335 (6)	3.135 (5)	589,256 (3)	706,218 (3)
	up	0.756 (6)	0.355 (1)	3.251 (2)	596,692 (1)	712,352 (1)

Continued on next page

Table B.1: Predictive performance of all models: Boruta feature selection (column rankings in parentheses) (continued)

	Balancing	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
	smote	0.730 (20)	0.330 (7)	3.083 (8)	580,672 (5)	702,400 (5)
C5.0	orig	0.714 (25)	0.285 (22)	2.784 (26)	574,384 (6)	664,876 (25)
DT	weights	0.693 (27)	0.230 (28)	2.382 (28)	338,121 (30)	615,806 (30)
	down	0.740 (11)	0.312 (9)	3.127 (6)	550,951 (14)	694,918 (12)
	up	0.683 (28)	0.267 (27)	2.638 (27)	549,534 (16)	663,024 (26)
	smote	0.733 (17)	0.297 (16)	2.930 (15)	582,451 (4)	683,942 (19)
SVM	orig	0.619 (30)	0.210 (30)	2.323 (30)	500,389 (25)	665,561 (24)
	weights	0.625 (29)	0.214 (29)	2.360 (29)	486,172 (29)	658,925 (27)
	down	0.728 (22)	0.277 (25)	2.857 (20)	558,088 (12)	694,770 (13)
	up	0.749 (8)	0.312 (11)	3.069 (9)	558,403 (11)	699,494 (6)
	smote	0.749 (7)	0.312 (10)	3.040 (11)	565,845 (9)	687,303 (17)

Table B.2: Predictive performance of all models: RFE feature selection (column rankings in parentheses)

	Balancing	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
LogR	orig	0.739 (21)	0.307 (21)	3.120 (14)	544,668 (19)	691,881 (17)
	weights	0.756 (8)	0.309 (19)	2.974 (19)	509,031 (26)	679,111 (21)
	down	0.746 (19)	0.299 (23)	2.945 (22)	479,696 (29)	667,973 (26)
	up	0.755 (9)	0.308 (20)	2.981 (18)	511,240 (25)	679,702 (20)
	smote	0.752 (13)	0.318 (14)	3.113 (15)	508,390 (27)	666,395 (27)
RF	orig	0.749 (17)	0.353 (1)	3.266 (2)	577,603 (4)	700,875 (7)
	weights	0.751 (16)	0.351 (2)	3.178 (8)	575,637 (5)	700,440 (10)
	down	0.753 (11)	0.314 (16)	2.952 (20)	545,078 (18)	694,629 (15)
	up	0.744 (20)	0.315 (15)	3.083 (16)	582,687 (2)	704,069 (3)
	smote	0.739 (22)	0.297 (25)	2.923 (23)	562,638 (12)	698,437 (12)

Continued on next page

Table B.2: Predictive performance of all models: RFE feature selection (column rankings in parentheses) (continued)

	Balancing	AUC-ROC	AUC-PRC	TDL	TDP	Profit index
NN	orig	0.758 (4)	0.310 (18)	2.945 (21)	540,348 (21)	675,873 (24)
	weights	0.764 (2)	0.325 (9)	3.244 (3)	563,558 (11)	700,658 (8)
	down	0.757 (6)	0.325 (10)	3.135 (12)	558,511 (13)	701,279 (5)
	up	0.759 (3)	0.326 (8)	3.171 (9)	542,268 (20)	689,481 (18)
	smote	0.757 (5)	0.324 (11)	3.193 (7)	527,995 (23)	680,551 (19)
GBM	orig	0.753 (12)	0.337 (6)	3.164 (11)	550,387 (16)	693,296 (16)
	weights	0.748 (18)	0.343 (5)	3.171 (9)	566,757 (9)	705,505 (2)
	down	0.757 (7)	0.345 (3)	3.325 (1)	554,617 (15)	703,706 (4)
	up	0.752 (15)	0.344 (4)	3.244 (3)	594,221 (1)	707,359 (1)
	smote	0.730 (24)	0.314 (17)	3.025 (17)	567,016 (8)	700,916 (6)
C5.0	orig	0.711 (26)	0.298 (24)	2.908 (25)	575,520 (6)	672,681 (25)
DT	weights	0.617 (30)	0.167 (30)	0.614 (30)	263,916 (30)	549,453 (30)
	down	0.733 (23)	0.319 (13)	2.923 (23)	529,948 (22)	694,990 (14)
	up	0.677 (27)	0.254 (27)	2.594 (29)	545,469 (17)	675,952 (23)
	smote	0.721 (25)	0.286 (26)	2.813 (26)	568,900 (7)	676,496 (22)
SVM	orig	0.640 (28)	0.243 (29)	2.674 (27)	498,667 (28)	662,873 (29)
	weights	0.637 (29)	0.243 (28)	2.660 (28)	512,140 (24)	665,485 (28)
	down	0.755 (10)	0.307 (22)	3.127 (13)	577,609 (3)	700,544 (9)
	up	0.765 (1)	0.330 (7)	3.208 (5)	554,873 (14)	699,301 (11)
	smote	0.752 (14)	0.323 (12)	3.200 (6)	565,308 (10)	696,553 (13)

BIBLIOGRAPHY

Abbasimehr, H., Setak, M., Tarokh, M.J., 2014. A comparative assessment of the performance of ensemble learning in customer churn prediction. Int. Arab J. Inf. Technol. 11, 599–606.

Accorsi, R., Manzini, R., Maranesi, F., 2014. A decision-support system for the design and management of warehousing systems. Computers in Industry 65, 175–186.

Adenso-Díaz, B., Lozano, S., Palacio, A., 2017. Effects of dynamic pricing of perishable products on revenue and waste. Applied Mathematical Modelling 45, 148–164.

Ahmed, N.K., Atiya, A.F., Gayar, N.E., El-Shishiny, H., 2010. An empirical comparison of machine learning models for time series forecasting. Econometric Reviews 29, 594–621.

Ahn, J.H., Han, S.P., Lee, Y.S., 2006. Customer churn analysis: Churn determinants and mediation effects of partial defection in the korean mobile telecommunications service industry. Telecommunications Policy 30, 552–568.

Akerkar, R., 2013. Big data computing. Crc Press.

Alpaydin, E., 2009. Introduction to machine learning. MIT press.

Andrawis, R.R., Atiya, A.F., El-Shishiny, H., 2011. Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition. International Journal of Forecasting 27, 672–688.

Araman, V.F., Caldentey, R., 2009. Dynamic pricing for nonperishable products with demand learning. Operations Research 57, 1169–1188.

Arora, D., Malik, P., 2015. Analytics: Key to go from generating big data to deriving business value, in: 2015 IEEE first international conference on big data computing service and applications, IEEE. pp. 446–452.

Athanasopoulos, G., Hyndman, R.J., Kourentzes, N., Petropoulos, F., 2017. Forecasting with temporal hierarchies. European Journal of Operational Research 262, 60–74.

Au, T., Ma, G., Li, S., 2003a. Applying and evaluating models to predict customer attrition using data mining techniques. Journal of Comparative International Management 6.

Au, W.H., Chan, K.C., Yao, X., 2003b. A novel evolutionary data mining algorithm with applications to churn prediction. IEEE transactions on evolutionary computation 7, 532–545.

Bahnsen, A.C., Aouada, D., Ottersten, B., 2015. A novel cost-sensitive framework for customer churn predictive modeling. Decision Analytics 2, 5.

Bai, J., Perron, P., 2003. Computation and analysis of multiple structural change models. Journal of Applied Econometrics 18, 1–22.

Bajari, P., Nekipelov, D., Ryan, S.P., Yang, M., 2015. Machine learning methods for demand estimation. American Economic Review 105, 481–85.

Ban, G.Y., Keskin, N.B., 2019. Personalized dynamic pricing with machine learning: High dimensional features and heterogeneous elasticity. Available at SSRN 2972985 .

Bao, Y., Wang, W., Zhang, J., 2004. Forecasting intermittent demand by SVMs regression, in: Systems, Man and Cybernetics, 2004 IEEE International Conference on, IEEE. pp. 461–466.

Van den Berg, J.P., Sharp, G.P., Gademann, A.N., Pochet, Y., 1998. Forward-reserve allocation in a warehouse with unit-load replenishments. European Journal of Operational Research 111, 98–113.

Besbes, O., Zeevi, A., 2009. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. Operations Research 57, 1407–1420.

Bhattacharya, C., 1998. When customers are members: Customer retention in paid membership contexts. Journal of the Academy of Marketing Science 26, 31–44.

Bitran, G., Caldentey, R., 2003. An overview of pricing models for revenue management. Manufacturing & Service Operations Management 5, 203–229.

den Boer, A.V., 2015. Dynamic pricing and learning: Historical origins, current research, and new directions. Surveys in Operations Research and Management Science 20, 1–18.

Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2015. Time series analysis: Forecasting and control. John Wiley & Sons.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Breiman, L., Friedman, J., Stone, C.J., Olshen, R., 1984. Classification and Regression Trees. CRC Press.

Brown, B., Chui, M., Manyika, J., 2011. Are you ready for the era of big data. McKinsey Quarterly 4, 24–35.

Brynjolfsson, E., Hu, Y.J., Smith, M.D., 2009. A longer tail?: Estimating the shape of amazon's sales distribution curve in 2008, in: Workshop on Information Systems and Economics (WISE).

Buckinx, W., Van den Poel, D., 2005. Customer base analysis: Partial defection of behaviourally loyal clients in a non-contractual FMCG retail setting. European Journal of Operational Research 164, 252–268.

Charrad, M., Ghazzali, N., Boiteau, V., Niknafs, A., 2014. NbClust: An R package for determining the relevant number of clusters in a data set. Journal of Statistical Software 61, 1–36.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16, 321–357.

Chellappa, R., Konsynski, B., Sambamurthy, V., Shivendu, S., 2007. An empirical study of the myths and facts of digitization in the music industry, in: Presentation 2007 Workshop Information Systems Economics (WISE), Montreal.

Chen, C.P., Zhang, C.Y., 2014. Data-intensive applications, challenges, techniques and technologies: A survey on big data. Information Sciences 275, 314–347.

Chen, I.F., Lu, C.J., Chen, I.F., Lu, C.J., 2016. Sales forecasting by combining clustering and machine-learning techniques for computer retailing. Neural Computing and Applications , 1–15.

Chen, K., Hu, Y.H., Hsieh, Y.C., 2015a. Predicting customer churn from valuable B2B customers in the logistics industry: A case study. Information Systems and e-Business Management 13, 475–494.

Chen, X., Owen, Z., Pixton, C., Simchi-Levi, D., 2015b. A statistical learning approach to personalization in revenue management. Available at SSRN 2579462 .

Chen, Z.Y., Fan, Z.P., Sun, M., 2012. A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. European Journal of Operational Research 223, 461–472.

Cheung, W.C., Simchi-Levi, D., Wang, H., 2017. Dynamic pricing and demand learning with limited price experimentation. Operations Research 65, 1722–1731.

Chodak, G., 2016. The nuisance of slow moving products in electronic commerce. Professionals Center for Business Research 3, 11–16.

Chui, M., Manyika, J., Miremadi, M., Henke, N., Chung, R., Nel, P., Malhotra, S., 2018. Notes from the AI frontier: Insights from hundreds of use cases. McKinsey Global Institute .

Cleveland, R.B., Cleveland, W.S., Terpenning, I., 1990. STL: A seasonal-trend decomposition procedure based on Loess. Journal of Official Statistics 6, 3.

Cohen, M., Lobel, I., Paes Leme, R., 2016. Feature-based dynamic pricing. Available at SSRN 2737045 .

Collopy, F., Armstrong, J.S., 1992. Rule-based forecasting: Development and validation of an expert systems approach to combining time series extrapolations. Management Science 38, 1394–1414.

Columbus, Louis, 2019. How to improve supply chains with machine learning: 10 proven ways. URL: https://www.forbes.com/sites/louiscolumbus/2019/04/28/how-to-improve-supply-chains-with-machine-learning-10-proven-ways/.

Coussement, K., Van den Poel, D., 2008. Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. Expert Systems with Applications 34, 313–327.

Croston, J.D., 1972. Forecasting and stock control for intermittent demands. Operational Research Quarterly , 289–303.

Davenport, T.H., 2013. Analytics 3.0. Harvard Business Review 91, 64–+.

Davenport, T.H., Ronanki, R., 2018. Artificial intelligence for the real world. Harvard Business Review 96, 108–116.

De Bock, K.W., Van den Poel, D., 2012. Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models. Expert Systems with Applications 39, 6816–6826.

De Caigny, A., Coussement, K., De Bock, K.W., 2018. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. European Journal of Operational Research .

De Gooijer, J.G., Hyndman, R.J., 2006. 25 years of time series forecasting. International journal of forecasting 22, 443–473.

De Livera, A.M., Hyndman, R.J., Snyder, R.D., 2011. Forecasting time series with complex seasonal patterns using exponential smoothing. Journal of the American Statistical Association 106, 1513–1527.

Dekhne, A., Hastings, G., Murnane, J., Neuhaus, F., 2019. Automation in logistics: Big opportunity, bigger uncertainty. URL: https://www.mckinsey.com/industries/travel-transport-and-logistics/our-insights/automation-in-logistics-big-opportunity-bigger-uncertainty.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30.

Devaine, M., Gaillard, P., Goude, Y., Stoltz, G., 2013. Forecasting electricity consumption by aggregating specialized experts. Machine Learning 90, 231–260.

Dingli, A., Marmara, V., Fournier, N.S., 2017. Comparison of deep learning algorithms to predict customer churn within a local retail industry. International journal of machine learning and computing 7, 128–132.

Dominique-Ferreira, S., Vasconcelos, H., Proença, J.F., 2016. Determinants of customer price sensitivity: An empirical analysis. Journal of Services Marketing 30, 327–340.

Dubé, J.P., Misra, S., 2017. Scalable price targeting. Technical Report. National Bureau of Economic Research.

Dudek, G., 2015. Short-term load forecasting using random forests, in: Filev, D., Jabłkowski, J., Kacprzyk, J., Krawczak, M., Popchev, I., Rutkowski, L., Sgurev, V., Sotirova, E., Szynkarczyk, P., Zadrozny, S. (Eds.), Intelligent Systems' 2014, Springer International Publishing. pp. 821–828.

Dwyer, F.R., Tanner, J.F., 2002. Business marketing: Connecting strategy, relationships, and learning. McGraw-Hill New York.

Eaves, A.H., Kingsman, B.G., 2004. Forecasting for the ordering and stock-holding of spare parts. Journal of the Operational Research Society 55, 431–437.

Elberse, A., Oberholzer-Gee, F., 2006. Superstars and underdogs: An examination of the long tail phenomenon in video sales. Division of Research, Harvard Business School.

Elmaghraby, W., Keskinocak, P., 2003. Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. Management Science 49, 1287–1309.

Esser, K., Kurte, J., 2020. Kep-studie 2020-analyse des marktes in deutschland. Bundesverband Paket und Expresslogistik eV: Berlin, Germany .

Faber, A., Spinler, S., 2019a. An empirical assessment of method selection for intermittent demand prediction. Unpublished working paper .

Faber, A., Spinler, S., 2019b. Interpretable prediction of partial defection: A case study in the B2B parcel logistics industry. Unpublished working paper .

Faber, A., Spinler, S., 2019c. Machine learning–based prediction of price sensitivity: A case study in the B2B parcel logistics industry. Unpublished working paper .

Fader, P.S., Hardie, B.G., 2007. How to project customer retention. Journal of Interactive Marketing 21, 76–90.

Fader, P.S., Hardie, B.G., 2009. Probability models for customer-base analysis. Journal of Interactive Marketing 23, 61–69.

Fathian, M., Hoseinpoor, Y., Minaei-Bidgoli, B., 2016. Offering a hybrid approach of data mining to predict the customer churn based on bagging and boosting methods. Kybernetes 45, 732–743.

Ferreira, K.J., Lee, B.H.A., Simchi-Levi, D., 2015. Analytics for an online retailer: Demand forecasting and price optimization. Manufacturing & Service Operations Management 18, 69–88.

Ferreira, K.J., Simchi-Levi, D., Wang, H., 2018. Online network revenue management using thompson sampling. Operations Research 66, 1586–1602.

Fildes, R., Goodwin, P., Lawrence, M., Nikolopoulos, K., 2009. Effective forecasting and judgmental adjustments: An empirical evaluation and strategies for improvement in supply-chain planning. International Journal of Forecasting 25, 3–23.

Fisher, M., Gallino, S., Li, J., 2017. Competition-based dynamic pricing in online retailing: A methodology validated with field experiments. Management Science 64, 2496–2514.

Frazelle, E., Hackman, S., Passy, U., Platzman, L., 1994. The forward reserve problem. optimization in industry, 2.

Freitas, A.A., 2014. Comprehensible classification models: A position paper. ACM SIGKDD explorations newsletter 15, 1–10.

Freund, Y., Schapire, R.E., et al., 1996. Experiments with a new boosting algorithm, in: Icml, Citeseer. pp. 148–156.

Friedman, J., Hastie, T., Tibshirani, R., et al., 2000. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). The Annals of Statistics 28, 337–407.

Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. Annals of Statistics , 1189–1232.

Frow, P., Payne, A., 2009. Customer relationship management: A strategic perspective. Journal of Business Market Management 3, 7–27.

Gaillard, P., Goude, Y., 2016. opera: Online Prediction by Expert Aggregation. URL: `https://CRAN.R-project.org/package=opera`. r package version 1.0.

Gamberini, R., Lolli, F., Rimini, B., Sgarbossa, F., 2010. Forecasting of sporadic demand patterns with seasonality and trend components: An empirical comparison between Holt-Winters and (S)ARIMA methods. Mathematical Problems in Engineering 2010.

García, D.L., Nebot, À., Vellido, A., 2017. Intelligent data analysis approaches to churn as a business problem: a survey. Knowledge and Information Systems 51, 719–774.

Ge, Y., He, S., Xiong, J., Brown, D.E., 2017. Customer churn analysis for a software-as-a-service company, in: Systems and Information Engineering Design Symposium (SIEDS), 2017, IEEE. pp. 106–111.

Gesing, B., Peterson, S.J., Michelsen, D., 2018. Artificial intelligence in logistics: A collaborative report by DHL and IBM on implications and use cases for the logistics industry. DHL Customer Solutions & Innovation .

Glady, N., Baesens, B., Croux, C., 2009. Modeling churn using customer lifetime value. European Journal of Operational Research 197, 402–411.

Gregory, B., 2018. Predicting customer churn: Extreme gradient boosting with temporal data. arXiv preprint arXiv:1802.03396 .

Gunasekaran, A., Papadopoulos, T., Dubey, R., Wamba, S.F., Childe, S.J., Hazen, B., Akter, S., 2017. Big data and predictive analytics for supply chain and organizational performance. Journal of Business Research 70, 308–317.

Gupta, R., Pathak, C., 2014. A machine learning framework for predicting purchase by online customers based on dynamic pricing. Procedia Computer Science 36, 599–605.

Gustafsson, A., Johnson, M.D., Roos, I., 2005. The effects of customer satisfaction, relationship commitment dimensions, and triggers on customer retention. Journal of Marketing 69, 210–218.

Gutierrez, R.S., Solis, A.O., Mukhopadhyay, S., 2008. Lumpy demand forecasting using neural networks. International Journal of Production Economics 111, 409–420.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182.

Hackman, S.T., Rosenblatt, M.J., Olin, J.M., 1990. Allocating items to an automated storage and retrieval system. IIE transactions 22, 7–14.

Hadden, J., Tiwari, A., Roy, R., Ruta, D., 2006. Churn prediction: Does technology matter. International Journal of Intelligent Technology 1, 104–110.

Hanley, J.A., McNeil, B.J., 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143, 29–36.

Hansen, J., McDonald, J., Nelson, R., 2006. Some evidence on forecasting time-series with support vector machines. Journal of the Operational Research Society 57, 1053–1063.

Hartigan, J.A., Wong, M.A., 1979. Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics) 28, 100–108.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The elements of statistical learning: Data mining, inference, and prediction. Springer New York.

He, H., Garcia, E.A., 2009. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21, 1263–1284.

Herrera, M., Torgo, L., Izquierdo, J., Pérez-García, R., 2010. Predictive models for forecasting hourly urban water demand. Journal of Hydrology 387, 141–150.

Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X., 2013. Applied logistic regression. John Wiley & Sons.

Hsu, C.W., Chang, C.C., Lin, C.J., et al., 2003. A practical guide to support vector classification .

Huang, Y., Kechadi, T., 2013. An effective hybrid learning system for telecommunication churn prediction. Expert Systems with Applications 40, 5635–5647.

Hung, S.Y., Yen, D.C., Wang, H.Y., 2006. Applying data mining to telecom churn management. Expert Systems with Applications 31, 515–524.

Hyndman, R.J., Athanasopoulos, G., 2014. Forecasting: Principles and practice. OTexts.

Hyndman, R.J., Khandakar, Y., et al., 2007. Automatic time series for forecasting: The forecast package for R. 6/07, Monash University, Department of Econometrics and Business Statistics.

Hyndman, R.J., Koehler, A.B., Snyder, R.D., Grose, S., 2002. A state space framework for automatic forecasting using exponential smoothing methods. International Journal of Forecasting 18, 439–454.

Hyndman, R.J., et al., 2006. Another look at forecast-accuracy metrics for intermittent demand. Foresight: The International Journal of Applied Forecasting 4, 43–46.

Jahromi, A.T., Stakhovych, S., Ewing, M., 2014. Managing B2B customer churn, retention and profitability. Industrial Marketing Management 43, 1258–1268.

Kane, M., Price, N., Scotch, M., Rabinowitz, P., 2014. Comparison of ARIMA and random forest time series models for prediction of avian influenza H5N1 outbreaks. BMC Bioinformatics 15.

Keilwagen, J., Grosse, I., Grau, J., 2014. Area under precision-recall curves for weighted and unweighted data. PLoS One 9, e92209.

Keskin, N.B., Zeevi, A., 2014. Dynamic pricing with an unknown demand model: Asymptotically optimal semi-myopic policies. Operations Research 62, 1142–1167.

Keskin, N.B., Zeevi, A., 2016. Chasing demand: Learning and earning in a changing environment. Mathematics of Operations Research 42, 277–307.

Kourentzes, N., 2013. Intermittent demand forecasts with neural networks. International Journal of Production Economics 143, 198–206.

Kourentzes, N., Barrow, D.K., Crone, S.F., 2014a. Neural network ensemble operators for time series forecasting. Expert Systems with Applications 41, 4235–4244.

Kourentzes, N., Petropoulos, F., 2016. Forecasting with r, in: International Symposium on Forecasting, p. 19th.

Kourentzes, N., Petropoulos, F., Trapero, J.R., 2014b. Improving forecasting by estimating time series structural components across multiple frequencies. International Journal of Forecasting 30, 291–302.

Krzanowski, W.J., Hand, D.J., 2009. ROC curves for continuous data. CRC Press.

Kuhn, M., 2012. Variable selection using the caret package URL: `https://www.idg.pl/mirrors/CRAN/web/packages/caret/vignettes/caretSelection.pdf`.

Kuhn, M., 2015. Caret: classification and regression training. ascl , ascl–1505.

Kumar, D.A., Ravi, V., 2008. Predicting credit card customer churn in banks using data mining. International Journal of Data Analysis Techniques and Strategies 1, 4–28.

Kursa, M.B., Rudnicki, W.R., et al., 2010. Feature selection with the Boruta package. J Stat Softw 36, 1–13.

Laney, D., 2001. 3D data management: Controlling data volume, velocity and variety. META Group Research Note 6, 1.

Lantz, B., 2013. Machine learning with R. Packt publishing ltd.

LaPlaca, P.J., Katrichis, J.M., 2009. Relative presence of business-to-business research in the marketing literature. Journal of Business-to-Business Marketing 16, 1–22.

Larivière, B., Van den Poel, D., 2005. Predicting customer retention and profitability by using random forests and regression forests techniques. Expert Systems with Applications 29, 472–484.

Lemke, C., Gabrys, B., 2010. Meta-learning for time series forecasting and forecast combination. Neurocomputing 73, 2006–2016.

Lemmens, A., Croux, C., 2006. Bagging and boosting classification trees to predict churn. Journal of Marketing Research 43, 276–286.

Lemmens, A., Gupta, S., 2017. Managing churn to maximize profits. Available at SSRN 2964906 .

Lessmann, S., Voß, S., 2009. A reference model for customer-centric data mining with support vector machines. European Journal of Operational Research 199, 520–530.

Ling, C.X., Li, C., 1998. Data mining for direct marketing: Problems and solutions, in: Kdd, pp. 73–79.

Liu, H., Motoda, H., Setiono, R., Zhao, Z., 2010. Feature selection: An ever evolving frontier in data mining, in: Feature Selection in Data Mining, pp. 4–13.

Lolli, F., Gamberini, R., Regattieri, A., Balugani, E., Gatos, T., Gucci, S., 2017. Single-hidden layer neural networks for forecasting intermittent demand. International Journal of Production Economics 183, 116–128.

Lu, N., Lin, H., Lu, J., Zhang, G., 2014. A customer churn prediction model in telecom industry using boosting. IEEE Transactions on Industrial Informatics 10, 1659–1665.

Makridakis, S., Wheelwright, S.C., Hyndman, R.J., 2008. Forecasting methods and applications. John Wiley & Sons.

Mashayekhi, M., Gras, R., 2015. Rule extraction from random forest: The RF+ HC methods, in: Canadian Conference on Artificial Intelligence, Springer. pp. 223–237.

Mattison, R., 2001. Telecom churn management: The golden opportunity. APDG Publ.

Miao, S., Chen, X., Chao, X., Liu, J., Zhang, Y., 2019. Context-based dynamic pricing with online clustering. arXiv preprint arXiv:1902.06199 .

Moon, M.A., Mentzer, J.T., Smith, C.D., 2003. Conducting a sales forecasting audit. International Journal of Forecasting 19, 5–25.

Morton, E., 2017. More products lead to more growth in online retailing. URL: https://www.digitalcommerce360.com/2017/08/29/products-lead-growth-online-retailing/.

Mukherjee, S., Osuna, E., Girosi, F., 1997. Nonlinear prediction of chaotic time series using support vector machines, in: Neural Networks for Signal Processing VII. Proceedings of the 1997 IEEE Signal Processing Society Workshop, IEEE. pp. 511–520.

Mukhopadhyay, S., Solis, A.O., Gutierrez, R.S., 2012. The accuracy of non-traditional versus traditional methods of forecasting lumpy demand. Journal of Forecasting 31, 721–735.

Neslin, S.A., Gupta, S., Kamakura, W., Lu, J., Mason, C.H., 2006. Defection detection: Measuring and understanding the predictive accuracy of customer churn models. Journal of Marketing Research 43, 204–211.

Nguyen, T., Li, Z., Spiegler, V., Ieromonachou, P., Lin, Y., 2018. Big data analytics in supply chain management: A state-of-the-art literature review. Computers & Operations Research 98, 254–264.

Nie, G., Rowe, W., Zhang, L., Tian, Y., Shi, Y., 2011. Credit card churn forecasting by logistic regression and decision tree. Expert Systems with Applications 38, 15273–15285.

Nikolopoulos, K., 2020. We need to talk about intermittent demand forecasting. European Journal of Operational Research .

Nikolopoulos, K., Syntetos, A.A., Boylan, J.E., Petropoulos, F., Assimakopoulos, V., 2011. An aggregate–disaggregate intermittent demand approach (ADIDA) to forecasting: An empirical proposition and analysis. Journal of the Operational Research Society 62, 544–554.

Petropoulos, F., Kourentzes, N., 2015a. Forecast combinations for intermittent demand. Journal of the Operational Research Society 66, 914–924.

Petropoulos, F., Kourentzes, N., 2015b. Forecast combinations for intermittent demand. Journal of the Operational Research Society 66, 914–924.

Petropoulos, F., Kourentzes, N., Nikolopoulos, K., 2016. Another look at estimators for intermittent demand. International Journal of Production Economics 181, 154–161.

Petropoulos, F., Nikolopoulos, K., Spithourakis, G.P., Assimakopoulos, V., 2013. Empirical heuristics for improving intermittent demand forecasting. Industrial Management & Data Systems 113, 683–696.

Phillips, R.L., 2005. Pricing and revenue optimization. Stanford University Press.

Van den Poel, D., Lariviere, B., 2004. Customer attrition analysis for financial services using proportional hazard models. European Journal of Operational Research 157, 196–217.

Qiang, S., Bayati, M., 2016. Dynamic pricing with demand covariates. Available at SSRN 2765257 .

Qu, H., Ryzhov, I.O., Fu, M.C., Bergerson, E., Kurka, M., 2016. Learning demand curves in B2B pricing: A new framework and case study. Submitted for publication .

Quinlan, J.R., 1986. Induction of decision trees. Machine Learning 1, 81–106.

Rauyruen, P., Miller, K.E., 2007. Relationship quality as a predictor of B2B customer loyalty. Journal of Business Research 60, 21–31.

Reichheld, F.F., 1996. Learning from customer defections. Harvard business review 74, 56–67.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016. Why should I trust you?: Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM. pp. 1135–1144.

Ridgeway, G., 2007. Generalized boosted models: A guide to the gbm package. Update 1, 2007.

Rumelhart, D.E., Hinton, G.E., Williams, R.J., et al., 1988. Learning representations by back-propagating errors. Cognitive Modeling 5, 1.

Runge, J., Gao, P., Garcin, F., Faltings, B., 2014. Churn prediction for high-value players in casual social games, in: Computational Intelligence and Games (CIG), 2014 IEEE Conference on, IEEE. pp. 1–8.

Schlosser, R., Boissier, M., 2018. Dynamic pricing under competition on online marketplaces: A data-driven approach, in: Proceedings of the 24th ACM

SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM. pp. 705–714.

Schoenherr, T., Speier-Pero, C., 2015. Data science, predictive analytics, and big data in supply chain management: Current state and future potential. Journal of Business Logistics 36, 120–132.

Scholkopf, B., Smola, A.J., 2001. Learning with kernels: Support vector machines, regularization, optimization, and beyond. MIT press.

Schuster, M., Paliwal, K.K., 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45, 2673–2681.

Shaaban, E., Helmy, Y., Khder, A., Nasr, M., 2012. A proposed churn prediction model .

Shiller, B.R., et al., 2013. First degree price discrimination using big data. Brandeis Univ., Department of Economics.

Sivarajah, U., Kamal, M.M., Irani, Z., Weerakkody, V., 2017. Critical analysis of big data challenges and analytical methods. Journal of Business Research 70, 263–286.

Stock, R.M., 2005. Can customer satisfaction decrease price sensitivity in business-to-business markets? Journal of Business-to-Business Marketing 12, 59–87.

Strack, G., Pochet, Y., 2010. An integrated model for warehouse and inventory planning. European Journal of Operational Research 204, 35–50.

Sugar, C.A., James, G.M., 2003. Finding the number of clusters in a dataset: An information-theoretic approach. Journal of the American Statistical Association 98, 750–763.

Syntetos, A.A., Babai, M.Z., Dallery, Y., Teunter, R., 2009. Periodic control of intermittent demand items: Theory and empirical analysis. Journal of the Operational Research Society 60, 611–618.

Syntetos, A.A., Babai, M.Z., Gardner Jr, E.S., 2015. Forecasting intermittent inventory demands: simple parametric methods vs. bootstrapping. Journal of Business Research 68, 1746–1752.

Syntetos, A.A., Boylan, J.E., 2005. The accuracy of intermittent demand estimates. International Journal of forecasting 21, 303–314.

Taieb, S.B., Hyndman, R.J., 2014. A gradient boosting approach to the kaggle load forecasting competition. International Journal of Forecasting 30, 382–394.

Tamaddoni, A., Stakhovych, S., Ewing, M., 2016. Comparing churn prediction techniques and assessing their performance: A contingent perspective. Journal of Service Research 19, 123–141.

Tanford, S., Raab, C., Kim, Y.S., 2012. Determinants of customer loyalty and purchasing behavior for full-service and limited-service hotels. International Journal of Hospitality Management 31, 319–328.

Teunter, R.H., Duncan, L., 2009a. Forecasting intermittent demand: A comparative study. Journal of the Operational Research Society 60, 321–329.

Teunter, R.H., Duncan, L., 2009b. Forecasting intermittent demand: a comparative study. Journal of the Operational Research Society 60, 321–329.

Teunter, R.H., Syntetos, A.A., Babai, M.Z., 2011. Intermittent demand: Linking forecasting to inventory obsolescence. European Journal of Operational Research 214, 606–615.

Thiele, S., Dieke, A.K., 2018. The impact of competition on consumer prices for cross-border parcels, in: The Contribution of the Postal and Delivery Sector. Springer, pp. 257–269.

Thomassey, S., Fiordaliso, A., 2006. A hybrid sales forecasting system based on clustering and decision trees. Decision Support Systems 42, 408–421.

Thorndike, R.L., 1953. Who belongs in the family? Psychometrika 18, 267–276.

Timmermann, A., 2006. Forecast combinations. Handbook of Economic Forecasting 1, 135–196.

Tiwari, S., Wee, H.M., Daryanto, Y., 2018. Big data analytics in supply chain management between 2010 and 2016: Insights to industries. Computers & Industrial Engineering 115, 319–330.

Tsai, C.F., Chen, M.Y., 2010. Variable selection by association rules for customer churn prediction of multimedia on demand. Expert Systems with Applications 37, 2006–2015.

Vafeiadis, T., Diamantaras, K.I., Sarigiannidis, G., Chatzisavvas, K.C., 2015. A comparison of machine learning techniques for customer churn prediction. Simulation Modelling Practice and Theory 55, 1–9.

Vassakis, K., Petrakis, E., Kopanakis, I., 2018. Big data analytics: Applications, prospects and challenges, in: Mobile Big Data. Springer, pp. 3–20.

Verbeke, W., Dejaeger, K., Martens, D., Hur, J., Baesens, B., 2012. New insights into churn prediction in the telecommunication sector: A profit driven

data mining approach. European Journal of Operational Research 218, 211–229.

Verbeke, W., Martens, D., Mues, C., Baesens, B., 2011. Building comprehensible customer churn prediction models with advanced rule induction techniques. Expert Systems with Applications 38, 2354–2364.

Verbraken, T., Verbeke, W., Baesens, B., 2013. A novel profit maximizing metric for measuring classification performance of customer churn prediction models. IEEE transactions on knowledge and data engineering 25, 961–973.

Vives, A., Jacob, M., Aguiló, E., 2018. Online hotel demand model and own-price elasticities: An empirical application in a mature resort destination. Tourism Economics .

Waller, M.A., Fawcett, S.E., 2013. Data science, predictive analytics, and big data: A revolution that will transform supply chain design and management. Journal of Business Logistics 34, 77–84.

Wallström, P., Segerstedt, A., 2010. Evaluation of forecasting error measurements and techniques for intermittent demand. International Journal of Production Economics 128, 625–636.

Wang, G., Gunasekaran, A., Ngai, E.W., Papadopoulos, T., 2016. Big data analytics in logistics and supply chain management: Certain investigations for research and applications. International Journal of Production Economics 176, 98–110.

Wang, X., Smith-Miles, K., Hyndman, R., 2009. Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series. Neurocomputing 72, 2581–2594.

Wei, C.P., Chiu, I.T., 2002. Turning telecommunications call details to churn prediction: A data mining approach. Expert Systems with Applications 23, 103–112.

Weinberg, C.R., Gladen, B.C., 1986. The beta-geometric distribution applied to comparative fecundability studies. Biometrics , 547–560.

Wiersema, F., 2013. The B2B agenda: The current state of B2B marketing and a look ahead. Industrial Marketing Management 42, 470–488.

Willemain, T.R., Smart, C.N., Schwarz, H.F., 2004. A new approach to forecasting intermittent demand for service parts inventories. International Journal of forecasting 20, 375–387.

Willemain, T.R., Smart, C.N., Shockor, J.H., DeSautels, P.A., 1994. Forecasting intermittent demand in manufacturing: A comparative evaluation of

croston's method. International Journal of Forecasting 10, 529–538.

Witten, I.H., Frank, E., Hall, M.A., Pal, C.J., 2016. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Xia, G.e., Jin, W.d., 2008. Model of customer churn prediction on support vector machine. Systems Engineering-Theory & Practice 28, 71–77.

Xie, Y., Li, X., Ngai, E., Ying, W., 2009. Customer churn prediction using improved balanced random forests. Expert Systems with Applications 36, 5445–5449.

Yi, X., Liu, F., Liu, J., Jin, H., 2014. Building a network highway for big data: Architecture and challenges. IEEE Network 28, 5–13.

Zhang, G., Patuwo, B.E., Hu, M.Y., 1998. Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting 14, 35–62.

Zhang, G.P., 2003. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50, 159–175.

Zhang, G.P., Qi, M., 2005. Neural network forecasting for seasonal and trend time series. European journal of operational research 160, 501–514.

Zhang, J.Z., Netzer, O., Ansari, A., 2014. Dynamic targeted pricing in B2B relationships. Marketing Science 33, 317–337.

Zhang, X., Zhu, J., Xu, S., Wan, Y., 2012. Predicting customer churn through interpersonal influence. Knowledge-Based Systems 28, 97–104.

Zhao, Y., Li, B., Li, X., Liu, W., Ren, S., 2005. Customer churn prediction using improved one-class support vector machine, in: International Conference on Advanced Data Mining and Applications, Springer. pp. 300–306.

Zhou, L., Pan, S., Wang, J., Vasilakos, A.V., 2017. Machine learning on big data: Opportunities and challenges. Neurocomputing 237, 350–361.

Zicari, R.V., 2014. Big data: Challenges and opportunities. Big Data Computing 1, 103–128.