To order or not to order

Predicting customer grocery shopping behaviour using multi-label classification techniques

R.B. Verbruggen

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

To order or not to order

Predicting customer grocery shopping behaviour using multi-label classification techniques

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

R.B. Verbruggen

April 17, 2020

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Picnic Technologies BV. Their cooperation is hereby gratefully acknowledged.

Abstract

Research and Objective: In the recent years the online grocery sector experienced an enormous uplift and evolved to a highly competitive business sector. Within this demanding environment, the need for strategic information has become extremely important, as it greatly enhances decision-making processes and the optimisation of the supply chain. In this research, a novel approach is proposed that is aimed at predicting customers' daily purchase probabilities, with the goal to improve short-term forecasting accuracy. Besides the well-acknowledged importance of forecasting practices and customer relationship management, this research is motivated by three main observations in online grocery retail; short interpurchase times, consistent shopping patterns and loyal customers.

Methodology: The approach involves the application of binary classification methods to analyse and predict online shopping behaviour. Within this context, two non-parametric learning algorithms, namely stochastic gradient boosting and random forest, are compared to traditional logistic regression. Both stochastic gradient boosting and logistic regression are extended using classifier chains (CC) to handle multiple outputs. Subsequently, the obtained purchase probabilities are aggregated and compared to the predictions of a univariate Seasonal Autoregressive Integrated Moving Average Exogenous (SARIMAX) time series model.

Results: The boosted tree CC model was able to achieve an improvement of 1.77% in mean-absolute-percentage error (MAPE) and 20.95% in mean-squared logarithm of the accuracy ratio (MSLAR) compared to the predictions of the random forest and an improvement of 1.15% in MAPE and 16.81% in MSLAR compared to the SARIMAX time series model. The model acquired consistent results for customer groups of different sizes, with prediction errors that exhibited the lowest bias as well as variance of all models. The analysis of the explanatory variables indicate that behavioural attributes and variables, that concern interpurchase times in particular, were most significant of the target variables. Eventually, the application of calibration methods led to a decrease in forecasting performance rather than improving it.

Conclusion: This research proposes a novel approach for short-term customer demand prediction within the online grocery retail market, which can provide an alternative to conventional time series forecasting techniques. The obtained results are satisfactory and of value for management and decision makers.

Table of Contents

1 Introduction 1

1-1 Motivation . . . 1

1-2 Problem Definition . . . 4

1-3 Outline . . . 5

2 Literature 7

2-1 Time Series Forecasting . . . 7

2-1-1 Time Series Analysis and Modeling . . . 8

2-1-2 Machine Learning . . . 8

2-2 Customer Behaviour Prediction . . . 9

2-2-1 Binary Classification . . . 9

2-2-2 Explanatory Analysis . . . 14

2-3 Literature Gap . . . 16

3 Data 17

3-1 Explanatory Variables . . . 17

3-1-1 Customer Demographics . . . 18

3-1-2 Customer Purchase behaviour . . . 19

3-1-3 Event Data . . . 21

3-1-4 Capacity Constraints . . . 22

3-1-5 Meta Data . . . 22

3-2 Pre-Processing . . . 22

3-2-1 Cleansing . . . 23

3-2-2 Transformation . . . 24

4 Methodology 25

4-1 Learning Algorithms . . . 25

4-1-1 Seasonal Autoregressive Integrated Moving Average Exogenous (SARIMAX) model . . . 26

4-1-2 Logistic Regression . . . 28

4-1-3 Decision Tree Ensemble Methods . . . 28

4-2 Multi-label Classification . . . 34

4-2-1 Classifier Chains . . . 35

4-2-2 Adapted Algorithms . . . 36

4-3 Feature Exploration and Selection . . . 37

4-3-1 Variable Statistics . . . 37

4-3-2 Variable Importance . . . 38

4-4 Hyperparameter Tuning . . . 40

4-4-1 SARIMAX . . . 41

4-4-2 Logistic Regression . . . 41

4-4-3 Random Forest . . . 42

4-4-4 Stochastic Gradient Boosting . . . 43

4-5 Probability Calibration . . . 44

4-5-1 Platt Scaling . . . 45

4-5-2 Isotonic Regression . . . 45

4-6 Model Performance Evaluation . . . 46

4-6-1 Separability and Classification . . . 46

4-6-2 Calibration Performance . . . 48

4-6-3 Forecasting Accuracy . . . 49

4-6-4 Training, Validation and Testing . . . 50

5 Results 53

5-1 Feature Importance . . . 53

5-2 Calibration . . . 56

5-3 Final Model Performance Comparison . . . 59

5-4 Classification . . . 61

6 Discussion 65

6-1 Principal Findings . . . 65

6-1-1 Model Performance . . . 65

6-1-2 Explanatory Variables . . . 67

6-2 Business Implications . . . 67

6-3 Limitations . . . 68

6-4 Recommendations and Future Research . . . 69

7 Conclusion 73

A Correlation Matrix **75**

B Model Configuration **77**

B-1 Final Model Configurations 78

B-2 Hyperparameter: Number of Trees (Random Forest) 79

C Model Performance **81**

C-1 Forecasting 82

C-2 Prediction Errors 85

C-2-1 Comparison of the prediction errors (3250 customers) 85

C-2-2 Comparison of the prediction errors (6500 customers) 86

C-2-3 Comparison of the prediction errors (13000 customers) 87

C-3 Classification 89

C-3-1 Precision and Recall plots (3250 customers) 89

C-3-2 Precision and Recall plots (6500 customers) 92

C-3-3 Precision and Recall plots (13000 customers) 95

Bibliography **99**

List of Figures

2-1 A simple linear support vector machine (left) [132, Figure 1 (a)]. 12

2-2 Typical feed-forward neural network composed of three layers [128, Figure 1]. . . 13

3-1 Schematic representation of the period of analysis. 18

4-1 Schematic Overview of a partitioned two-dimensional feature space (left) and corresponding decision tree (right) [55, Figure 9.2]. 29

4-2 Schematic Overview of a Random Forest Model. 31

4-3 Example transformation under BR (left) and CC (right) for $(\mathbf{y}, \mathbf{x})$ with binary attribute space [108, Figure 1]. 36

4-4 Feature Selection Funnel Approach. 37

4-5 k-fold nested cross-validation for time series. 51

5-1 Reliability curves. 57

5-2 Precision and Recall curves of the stochastic gradient boosting CC model (3250 customers). 62

A-1 Heatmap correlation matrix. 76

B-1 Brier score per horizon for different number of trees in the random forest model. 79

C-1 Histogram plots of the prediction errors. 85

C-1 Histogram plots of the prediction errors. 86

C-2 Histogram plots of the prediction errors. 86

C-2 Histogram plots of the prediction errors. 87

C-3 Histogram plots of the prediction errors. 87

C-3 Histogram plots of the prediction errors. 88

C-4 Precision and Recall curves of the logistic regression CC model. 89

C-5 Precision and Recall curves of the random forest model. 90
C-6 Precision and Recall curves of the stochastic gradient boosting CC model. 91
C-7 Precision and Recall curves of the logistic regression CC model. 92
C-8 Precision and Recall curves of the random forest model. 93
C-9 Precision and Recall curves of the stochastic gradient boosting CC model. 94
C-10 Precision and Recall curves of the logistic regression CC model. 95
C-11 Precision and Recall curves of the random forest model. 96
C-12 Precision and Recall curves of the stochastic gradient boosting CC model. 97

List of Tables

4-1 Hyperparamter search space. 44

5-1 Permutation importance of the 20 most important variables. 54
5-2 Summary of feature impact on performance prior and post variable permutation. 55
5-3 Performance comparison prior and post feature selection. 56
5-4 Performance comparison prior and post calibration. 58
5-5 Model performance evaluation - 3250 customers. 60
5-6 Model computation time comparison. 61

B-1 Final model configurations. 78

C-1 Model performance evaluation - 3250 customers. 82
C-2 Model performance evaluation - 6500 customers. 83
C-3 Model performance evaluation - 13000 customers. 84

Chapter 1

Introduction

This chapter provides an introduction to the research conducted during this master thesis in the field of electronic commerce, commonly referred to as e-commerce. The research concerns the analysis, identification, and prediction of customer behaviour in the online grocery retail sector. A motivation of research within this area is given in section 1-1. Subsequently, a problem definition along with some research questions are stated in section 1-2. Finally, the organization of this thesis is outlined in section 1-3.

1-1 Motivation

Over the past years, e-commerce became increasingly popular as a result of today's knowledge-based economy and information society. E-commerce includes a variety of services, with Business-to-Customer trade being the most common. Driven by technological advances, that mostly concern modern communication systems and the World Wide Web, e-commerce has enabled customers to make purchases independent of time and location [96]. Retailers expand their businesses by building online stores, with the result of physical stores being 'displaced' rapidly. The transition from visiting brick-and-mortar stores to online shopping can be easily recognized by analysing the growth in terms of turnover. In 2019, for example brick-and-mortar turnover experienced a growth of 3.8% in the United States, whereas e-commerce could achieve a growth of almost 14.9% [54]. A few years back, online retailing included to most extent consumer goods that are related to books, music, electronics and fashion. However, more recently the online food and grocery market has experienced an enormous uplift as well. In 2018, 25% of all online orders in Europe were made for food and groceries [51].

Within the demanding and rapid developing online grocery retail sector, the need for strategic information has become extremely important. Not only accurate forecasts are needed for the short-term operations and mid-term alignment, but also retail managers need to have insight in the type of customers they have to supply as a supportive foundation for long-term planning. Demand management and future planning have great impact on overall organizational performance and is one of the key success factors for profitable businesses since it greatly helps improving decision-making processes and optimising the supply chain [52]. From strategic to operational areas, such as finance, marketing, recruitment, personnel planning, logistic planning, inventory management and maintenance, many decisions are supported by sales (or demand) forecasts [15]. Poor forecasting straight away affects revenue due to unnecessary high personnel and substantial write-off costs along with customer service issues, which subsequently harm the competitive position of businesses [2, 9]. Balancing customers requirements with the capabilities of the supply chain by having the right management in place, makes it possible to match supply with demand in a proactive manner [43].

In the past decades some major efforts have been done in the field of research, modeling and forecasting seasonal and trend time series data. A time series is a sequence of observations taken sequentially in time [19]. The variety of algorithms that have been developed since then reaches from longstanding best practices to cutting-edge methodologies [47]. Besides commonly used modeling methods, including simple historical averages and exponential smoothing, machine learning (ML) and artificial intelligence (AI) methodologies have established themselves as serious contenders in the area of forecasting [35]. The fast and ever evolving computing technologies have provided companies with the ability to collect, store and analyse data of unimaginable size. For each customer, thousands, or even millions of data objects that enable the analysis of the complete purchasing history are stored. By then, the methodology extended from analyzing sales data only to more advanced approaches, which involved the development and application of a wide range of data mining methodologies (e.g., pattern extraction from data, model fitting to data etc.), with the goal to identify and predict customer demand [14]. However, these techniques are not only used for forecasting purposes but also for the analysis of online customer behaviour, which dates back to the early beginning of e-commerce [11]. Some of the many applications involve the classification of customers into categories [94], item recommendation systems [112], customer churn prediction [42, 86, 141] and purchase probability estimation (propensity models) [84, 136]. Despite their different applications, all models, are ultimately using customer data, historical purchase and clickstream data (e.g., data about interactions between customer and website, application, etc.) to predict future customer behaviour. The rapid growth of e-commerce has changed the whole way of shopping and with it the traditional relationship between retailer and customer. Retailers face challenges like increasing competition and volatile relationships due to an anonymous shopping experience, resulting in less loyal customers. Therefore, most applications in the field of e-commerce, and many others (e.g., financial services etc.), have the goal to improve customer service and build longstanding relationships in order to strengthen the competitive position of the business and increase its revenue in the long run [97, 137, 141].

In general, demand forecasting and online customer purchase behaviour prediction are two separate methodologies. Whereas forecasting usually involves the modeling of time series data to predict future values of the series, customer behaviour prediction aims at the analyses, classification and prediction of individual customers, which suggests the application of other techniques, such as classification algorithms. The application of time series modeling and regression techniques in order to predict future values of a time series is quite evident, however, this thesis aims at using customer behaviour prediction methods to predict future demand of an online grocer. This approach includes the identification of online customer behaviour in order to obtain individual purchasing probabilities. Unlike previous applications, this involves not only the prediction of the probability that a customer will make a next purchase, but also when. Besides the increasing importance of classifying customer behaviour and a large number of possible prediction models and data sources, this research is motivated by three main observations in online grocery retail; short interpurchase times, consistent shopping patterns and loyal customers. The observed (purchase) penetration rate of single customers is significantly higher than in 'traditional' e-commerce, which leads to overall shorter interpurchase times. This is mainly the result of greater demand of groceries, which is usually lower for utensils or services that are available via other online services. Moreover, the observed shopping pattern of individual customers is more consistent and less volatile in comparison to other services due to weekly shopping routines. Lastly, the relationship between customer and business is less volatile as well. This is most likely to be explained by the relatively low density of online suppliers in the field of grocery, and people getting used to the service, application and the assortment of a single grocer. It is more convenient to stick with the same store rather than constantly switching between multiple stores.

The observations above motivate the possibility to identify weekly shopping routines of customers of an online grocery store and individual trends due to quantitative and qualitative input data that is available for each customer. It is expected that analysing the online shopping behaviour of individual customers is an effective approach to predict future demand. As Boone et. al stated: "The better a firm understands its customers' buying behaviours, the more accurate its demand forecasts will be, which in turn helps it to plan and execute supply chain operations more efficiently" [18, page 1].

Besides the established large supermarket chains that provide an online grocery shopping experience, there are a number of innovative companies trying to compete in the online grocery market. One of them is Picnic, a data-driven online only grocery store that also executes the whole logistics for home delivery using their own developed distribution model. Meanwhile it accounts for about 12% market share in this category since they started in 2015 [127]. The company chooses maximum growth over making profit [26], which made it possible to grow with an impressive 100% in 2018. In Picnic, accurate forecasting is needed in order to balance supply with demand, to ensure smooth operations along with waste and cost minimization. Long-term forecasts are used to plan openings of new facilities such as distribution and fulfilment centres, and the purchase of new delivery vans. Mid-term forecasts are used to set targets for recruiting new employees and marketing campaigns. Finally, the short-term forecast is used for capacity planning, trunking planning, personnel planning and inbound management purposes.

Currently, Picnic faces some challenges in the field of forecasting. All forecasting in retail depends on a degree of aggregation, whether it is on product units, location or time instances, depending on the purpose of the forecasting activity. Due to Picnics' unique supply chain, the lowest aggregation of interest is currently based on the delivery areas. Since Picnic is still growing at rapid pace, opening facilities and extending their service to new delivery areas, historical data is limited in such areas. However, most forecasting techniques require a certain amount of input data in order to provide (reasonable) results. At the moment, there is no more convenient way than applying a default purchase rate to predict future demand within the first couple of weeks/months of a new delivery area. Another challenge are capacity constraints, which straightaway affect customer behaviour. Constraining regular demand implicitly forces customer behaviour to change, which subsequently increases uncertainty within forecasting.

Given that customer demand, besides capacity constraints, is the only driver in Picnic sales, one can think of the potential benefits of an accurate customer demand forecasting tool. This approach introduces new challenges (e.g., increasing computational complexity), but also possibilities which have the potential to improve forecasting accuracy, inventory management, generate additional insights in the customer base and allow for both customized service and the analysis of predictions on any aggregation level. The belief exists that this approach can identify customer behaviour during irregular situations (such as constraint capacity), which in turn can improve predictability in such situations. Since the model is customer based, it can be implemented fairly quickly in, for example, new delivery areas. The model can be trained on a representative group of customers and right away applied to new customers with little purchasing history. Eventually, the possibility to include variables, such as discounts and free gifts that have an effect on customer demand [34], directly in the model forecast in a convenient way, makes it a very attractive approach.

1-2 Problem Definition

The focus of this research lies in predicting the daily purchasing probabilities of customers of an online grocery store, with the final goal to improve aggregated demand forecasting. This involves the implementation of learning algorithms for the analysis and identification of customer shopping behaviour based on historical data in order to predict their future actions in the context of purchases. The problem at hand is a classification problem of binary nature since the target variable has two categories: *purchase* and *no purchase*. The desired output of the model are individual customer purchase probabilities, which are then aggregated in order to obtain the demand forecast. The performance of the aggregated forecast is compared to a top-line time series model. For the sake of this project, it is decided to evaluate the predictions on individual customer level as well as on a total level. Furthermore, as the daily operation currently involves many challenges and is an area where still a lot of value can be accomplished, this thesis is aimed at developing a model that makes predictions on the short-term demand (e.g., a horizon of 7 days).

Earlier, L. Raasveld [107] conducted some research on *Predicting Invites Conversion*, in order to predict first order conversion of Picnic customers. This research will continue on her work and will consider customers that used the service at least once. Fortunately, this substantially reduces the amount of input data, which in turn speeds up the whole evaluation procedure. Deployment is out of scope for this research and can only be regarded as an implication if one of the models shows promising results.

There is one main research question that results from the points discussed above. Three sub questions provide guidance on how to answer it.

Research Question:

> Can customers' daily purchasing probability estimates provide an alternative to conventional time series modeling techniques?

Sub Questions:

1. How can the daily purchase probability of a customer of an online grocery store be obtained?
2. Which machine learning model is best suited to solve the prediction problem?
3. How do different explanatory variables influence model performance?

1-3 Outline

The remainder of this report is organized as follows. First, a literature review on customer behaviour prediction as well as forecasting in general is given in chapter 2, followed by chapter 3 about data exploration and pre-processing. Thereafter, in chapter 4, a methodology is proposed which includes the modeling procedure as well as the evaluation process of the models considered in this project. In chapter 5, the performance of the finalized models are compared for three different cases. Additionally, some intermediate results are given that are obtained during the modeling procedure. Chapter 6 discusses these findings by relating them back to the research questions stated above and contextualizing the results within the literature found in chapter 2. Furthermore, a review concerning limitations of this project is included and some recommendations for future research are posed. Finally, a conclusion of this research is given in chapter 7.

Chapter 2

Literature

This chapter briefly introduces the current state-of-the-art of time series forecasting, while providing a more detailed view on research that has been done on similar binary classification problems as the one at hand. Besides reviewing the different learning algorithms that have been applied within this field, a revision on the type of input data that is used for prediction is given. Finally, a gap analysis of the reviewed literature is conducted.

2-1 Time Series Forecasting

In the past decades some major efforts have been done in the field of research, modeling and forecasting seasonal and trend time series data. The variety of algorithms that have been developed since then, reaches from longstanding best practices to cutting-edge methodologies [47]. Besides traditional modeling methods, including simple historical averages and exponential smoothing, machine learning (ML) and artificial intelligence (AI) methodologies have established themselves as serious contenders in the area of forecasting [35]. While each have their strengths and weaknesses, at their core, every method is ultimately using historical data to try to predict future demand [47]. The complexity, assumptions, and types of data inputs used and how they are weighted, depend on the given model type, but the basic ingredients are similar across the board. There are two types of approaches to statistical analysis, univariate and multivariate. Univariate involves the analysis of a single variable while multivariate analysis examines two or more variables. Most multivariate analysis involves a single dependent variable and multiple independent variables. Whether a given model is better than the other mainly depends on the time series data itself, the purpose of the model, the domain of the forecasting problem and the quality of input data. Easy implementation, interpretation and understanding are all valuable assets in industry that often come in trade-off with accuracy. Chu and Zhang argue that "one of the major limitations of the traditional methods is that they are essentially linear methods" and "in order to use them, users must specify the model form without the necessary genuine knowledge about the complex relationship in the data" [35, page 2018]. However, if the linear less complex models can capture the underlying characteristics of the data evenly well, they should be preferred over more complicated models as they have the important practical advantage of easy interpretation and implementation. If the linear models fail to perform well in both in-sample fitting and out-of-sample forecasting, more complex nonlinear models should be considered. Machine learning models do a great job in discovering nonlinear and complex relationships in the data, not needing to manually select the exact model type, make assumptions about external factors or pre-defining some sort of heuristic or logic. Instead of explicitly weighting variables or variable interactions, many of these algorithms come with built in variable importance. However, this comes at the price of requiring a lot of input data and a fair amount of investment in setup and maintenance, the output is often less understandable and models can be prone to overfitting.

2-1-1 Time Series Analysis and Modeling

Time series analysis is the oldest and most widely discussed methodology for predicting future values of a time series [1, 23, 65, 98, 140, 109]. It is practiced in many ways where each model represents different stochastic processes of the series. In general, a time series can be decomposed into four constituent parts: level, trend, seasonality and noise. Where *level* embodies the baseline value of the series (as if it were a straight line), *trend* describes the optional and often linear increasing or decreasing behaviour of the series over time, *seasonality* describes the optional recurrent patterns over time and *noise* expresses the optional (but often present) variability in the observations that cannot be modeled by the model.

Three broad classes are considered when modeling variations in time series, that being the autoregressive (AR) models, the integrated (I) models, and the moving average (MA) models. All three classes depend linearly on previous data points [59]. Models like the famous univariate ARIMA model, introduced by Box and Jenkins [19], combine these classes into one model. To handle seasonality in time series, several methods have been developed. They range from one of the oldest techniques of seasonal decomposition (like the X-11 method and its variants [45, 53]) over to new methods, such as model-based approaches (like the TRAMO-SEATS [60]), nonparametric methods (like the STL [37]) and sinusoidal models [119]. Later studies accounted for the ability to include exogeneous regressors which resulted in various types of adaptations to the models [40, 76, 134].

2-1-2 Machine Learning

In the past years, further research extended the concept to other models, such as Artificial Neural Networks (ANNs), Decision Trees (DTs), Ensemble Methods (EMs), Support Vector Machines (SVMs), and others, that are mutually called machine learning models [6, 55]. Theoretical understanding and the amount of variations of the models developed have increased to an impressive extent. Unlike linear statistical techniques, these models are part of a more flexible class, not explicitly requiring to specify the functional relationship between input variables and output variable(s). Besides the numerous studies found in literature comparing various versions of ANNs with traditional approaches, a lot of effort has been done related to the research of more advanced machine learning models to model time series data. Especially in the case of modeling nonlinearities in the data, various machine learning approaches have shown their true potential compared to more conventional approaches that were not able to adequately capture the evolution of the series [32, 68, 116, 145]. However, not in every situation all machine learning models work evenly well, as their performance highly depend on the nature of the forecasting problem. Ahmed et al. [3] employed an empirical comparison of eight machine learning models and found significant differences between the models in terms of performance.

2-2 Customer Behaviour Prediction

In an era of strong customer relationship management (CRM) emphasis, it is becoming more evident that profitable businesses need to focus on building long-term relationships instead of being customer-driven only. Web shop visitors leave more traces than ever before, where any action is recorded and stored for analysis. The retrieved knowledge, in turn can improve customer satisfaction, by making the shopping process more engaging, personalized and efficient. In the long run, this can strengthen the competitive position resulting from higher conversion rates and increased turnover [14]. Therefore, many studies in this field are related to the research and development of customer behaviour mining models. Such models are aimed at predicting future behaviour of customers based on explanatory variables, which to most extent involve past purchasing behaviour and clickstream data. This section will mainly focus on propensity and partial-defection models, as these come closest to the problem at hand. The following sections review the application of various machine learning methods in this field of research, and summarizes the type of explanatory variables that were used for prediction.

2-2-1 Binary Classification

Classification is probably the most common among data mining applications and is aimed at predicting a categorical target variable from a set of input variables (also referred to as independent variables, features or explanatory variables). This target value can be represented either by multiple categories or, like the task at hand, a binary variable. A learning algorithm aims to learn a generalized relationship between input and target variables from a labeled data set in order to predict future target variables. This is called supervised learning [101]. Several machine learning algorithms exist that have different approaches to solve the task.

The most common type of machine learning algorithms for a binary classification task are vector-based methods. Belonging to this class are Decision Trees (DTs), Ensemble Tree methods such as Random Forests (RFs) and Gradient Boosted Trees (GBTs), Support Vector Machines (SVMs), Logistic Regression (LR), and Feed-forward Neural Networks (FNNs). These models belong to the category of eager learning methods. Given a set of training data, eager learners construct the model before receiving new (e.g., test) data to predict. The opposites are lazy learners, such as the K-nearest Neighbor (KNN) algorithm, where training data is simply stored and test data is awaited for classification [66]. Therefore, lazy learners take less training time but more time in predicting, while the opposite is the case within eager learning. All of the algorithms mentioned above, along with some applications, are briefly explained in the following sections.

Logistic Regression

Logistic regression (LR), also called logit regression, despite its name, is a linear model for classification rather than regression. Logistic regression, which first application was introduced by Berkson [13], is a statistical model that uses a logistic function to model the binary dependent variable. The log-odds (logarithm of the odds) is modeled as a linear combination of the independent variable(s), which can be both binary or continuous. The logistic function transforms the log-odds to the corresponding probability that can vary between 0 and 1 using the sigmoid function. The unit of measurement for the log-odds scale is called a logit (logistic unit). The generalized logistic model is defined as stated in Equation 4-4.

$$h_\theta(X) = \frac{1}{1+e^{-\theta^T X}} = P(Y=1|X;\theta), \tag{2-1}$$

where Y is the dependent variable, X the design matrix of independent variables and θ represents the regression coefficients that are estimated based on the mapping between input variables and the output variable [78]. The parameters of each input vector are estimated through maximum-likelihood estimation (Equation 4-1).

The probabilistic output of the sigmoid function makes the LR one of the most popular machine learning algorithms for binary classification. The parametric model has favorable characteristics such as computational efficiency, easy implementation and interpretability. On the downside, however, its simplistic modeling assumptions may lead to underfitting for complex nonlinear data sets. Since it uses linear combinations of variables it is not adept at modeling nonlinear complex interactions between variables [50]. Moreover, LR is sensitive to model misspecification as well as to noise, which suggests the removal of outliers before training.

In the context of an online store, Van den Pool and Buckinx [136] evaluate a broad range of attributes using logit modeling to predict whether or not a purchase is made during the next visit to the website. In the same year, Van den Pool and Buckinx [141] conducted some research on predicting partial defection by behaviourally loyal clients at an Fast-Moving Consumer Goods (FMCG) retailer and compared a logit model to Neural Networks and Random Forests. Their results show no significant differences in terms of performance between all three models. Despite its popularity and successful implementations, LR has been outperformed by other machine learning methods in the majority of studies, concerning various applications [80, 84, 86, 102].

Decision Tree

Due to its ease of use and interpretability, the DT algorithm has evolved to a popular concept among researchers and analysts [84], and comes closest to meeting the requirements for serving as an *off-the-shelf* procedure for data mining [55]. DTs use decision rules (or split conditions) on a set of independent variables in order to partition a heterogeneous population of observations into smaller, more homogeneous subgroups. The goal is to obtain the most homogeneous subgroups possible [55]. One major advantage are the simple classification rules, which greatly enhance interpretability of single DTs [66]. Comprehensibility decreases as the models grow larger and more unbalanced, however, remain easier to interpret than other, more black box models [110]. Next to that, they usually exhibit short learning and prediction times. They are robust to outliers and account for interaction effects between variables. However, DTs have one major flaw - that is - their high variance. They are prone to overfitting regarding noisy data. A small change in the data can result in a very different series of splits, hence in a complete different tree with different outcome [55].

Random Forest
Bagging (Bootstrap Aggregation) is a popular ensemble technique and very effective in reducing the variance of a single DT. By training DTs on several random subsets of the data and averaging the predictions will lead to a more robust result than a single decision tree [55]. The RF algorithm is an extension over bagging and is aimed at reducing the correlation between the sampled trees [55]. In addition to taking random subsets of the data, Random Forest also takes a random selection of features rather than using all features to grow the trees.

Gradient Tree Boosting
Another ensemble method that experienced great attention in both literature and industry is *Boosting*. According to Friedman et al. [55] it is one of the most powerful learning ideas that has been introduced in the last couple of decades. In boosting, a collection of weak predictors (DTs) are learned sequentially and trained iteratively on residuals. In other words, consecutive trees are fitted on random samples and at every step, the goal is to solve for net error from the prior tree. In case an input is not correctly classified by a hypothesis, its weight is increased so that the updated hypothesis is more likely to classify it correctly.

Note: The DT algorithm and the ensemble methods are explained in more detail in subsection 4-1-3.

Throughout the literature, DTs show great results in various applications that are aimed at predicting customer behaviour. While the results of Van den Pool and Buckinx [141] showed no significant differences in terms of performance among alternative classification techniques, other examples of applications for this purpose, including DTs [102] and SVMs [42], suggest differently. While logistic regression performs slightly better than DTs, it is outperformed by SVMs, whereas random forests outperform both kinds of models. Similarly, Larivière and Van den Poels' [84] findings demonstrate that random forest techniques provide a better fit for the estimation and validation sample compared to ordinary linear regression and logistic regression models in the context of a financial services company. Croux and Lemmens [86] investigated the contribution of bagging and boosting to classification trees in the case of predicting customer churn and found superiority in terms of performance compared to a binary logit model. In the same context, Xie et. al [143] obtained superior results of an improved random forest model over other methods, such as artificial neural networks (ANNs), DTs and SVMs. In the more recent study of Martinez et al. [91] a gradient tree boosting algorithm was implemented in order to predict future purchases in the non-contractual setting. The results indicate its superiority over logistic Lasso and extreme learning machines.

Support Vector Machine

The goal of the SVM algorithm is to find a hyperplane (as indicated in Figure 2-2) in an N dimensional space that partitions a heterogeneous population of observations into two, more homogeneous groups in order to distinctly classify them [41]. The dimension of the hyperplane depends on the number of independent variables. For example, if the number of input features is 3, then the hyperplane becomes a two-dimensional plane. By finding the optimal hyperplane out of all possibilities that maximizes the margin (distance) between data points of both classes, some reinforcement is achieved concerning the confidence of future predictions. Support vectors are the data points that are closest to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, the margin of the classifier is maximized.

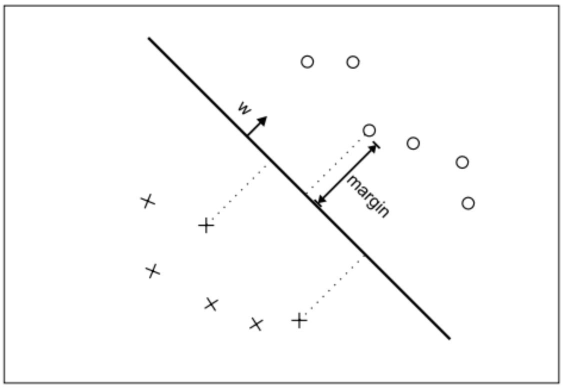


Figure 2-1: A simple linear support vector machine (left) [132, Figure 1 (a)].

In case the feature space is not linearly separable, a kernel-function is used to map data on a higher dimensional feature space where the data becomes linear separable. By mapping inputs to a high-dimensional feature space, it becomes possible for SVMs to not only model linear relationships but to also conduct non-linear classification [41].

SVMs are highly preferred by many as they produce significant accuracy with less computation power. Next to that, they are well suited for high-dimensional input. These advantages come in trade-off with long training times and being less interpretable. Furthermore, careful hyperparameter tuning is required, which can be difficult and time-consuming [66]. Another flaw of SVMs is that they do not directly output probabilities, which would require additional mapping of the output variable to probabilities.

Most implementations of SVM in customer behaviour prediction concern churn prediction. For example, Coussement and Van den Poel [42] implemented SVMs for predicting customer churn in subscription services while comparing two parameter-selection techniques. Both selection techniques led to slightly higher accuracy than achieved by a logit model, however, were outperformed by a random forest model. Also in the context of customer churn prediction, Xia and Jin [142] build a SVM and compared it with ANNs, DTs, LR, and a naive bayesian classifier. The results show superior performance of the SVM over the other models.

Feed-forward Neural Network

The FFN is the most basic class of the artificial neural networks, where information only flows one-directional from input layer to output layer. A neural network consists of neurons that are ordered into layers [128]. The first layer is called input layer and the last layer is called output layer. The layer(s) in between are called hidden layers. Each neuron in a particular layer is connected with all neurons in the following layer, where each connection is characterized with an individual weight coefficient. The coefficients are determined during the learning phase, for example via back-propagation. Back-propagation is the most common used learning method and uses the steepest-descent minimisation method. With enough hidden units, an FNN can approximate any function. Output is the probability of each class, which add up to one. From a statistical point of view, FNNs perform a nonlinear regression.

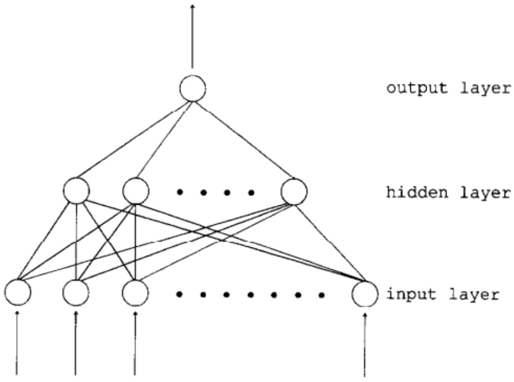


Figure 2-2: Typical feed-forward neural network composed of three layers [128, Figure 1].

FFN tend to provide good results as they generalize well to unseen data while being fairly robust to noisy data [66]. On the downside, they have little comprehensibility, model fitting takes long and hyperparameter tuning is less straightforward.

Suchacka and Templewski [125] succesfully implemented a FNN model to predict purchases in sessions and obtained good results. However, they provided no performance comparison with other models. Mozer et al. [97] compared a FNN with a DT algorithm in the context of subscriber dissatisfaction prediction in the wireless telecommunications industry. The results indicate that the FNN outperformed the DT. They also applied boosting to both models, which slightly enhanced performance. In the context of consumer choice prediction, Clemens et al. [58] implemented two ANN models and achieved superior results than with traditional logistic models.

K-nearest Neighbor

KNN is a lazy learning algorithm and has been used in statistical estimation and pattern recognition already in the 1970's as a non-parametric technique [49]. The KNN algorithm stores all available data in an N-dimensional feature space and awaits new observations for classification. An observation is classified by a majority vote of its neighbors, with the observation being assigned to the class most common amongst its K nearest neighbors measured by a similarity measure or distance metric (e.g., Euclidean distance etc.)[66]. Choosing the optimal K can reduce the overall noise, which therefore can lead to better results, and is done via inspection of the data or cross-validation.

Key advantage of the KNN algorithm is the simple and easy implementation, since there is no need for tuning several parameters or making additional assumptions. Furthermore, KNN are fairly robust to both noisy data and irrelevant input features, and have very fast training speed. The latter, however, comes in trade-off with very long prediction times. Another disadvantage is the computational performance that significantly decreases as the number of examples or independent variables increase. Similarly, comprehensibility decreases with high-dimensional input. Like SVMs, KNN models do not output probabilities directly.

In the field of e-commerce, the KNN algorithm has been mainly applied in the context of recommender systems. Products are recommended to a visitor of a web shop based on similar preferences of prior visitors (nearest neighbors). Whereas, limited research can be found concerning customer behaviour prediction. Suchacka et al. [124] implemented a KNN model that aimed at classifying customers of a web shop into buying or browsing sessions. The best results were obtained using 11 neighbors and indicate great performance, however no performance comparison with other models was conducted.

2-2-2 Explanatory Analysis

It is evident that selecting an appropriate model is an important step within predictive modeling, however, it is not the only one. Choosing the right input data to train the models is evenly, or maybe even more important. On the one hand that involves the quality of the data and on the other hand the type of data, whereas the latter is closely related to the purpose of the model. Since learning algorithms aim to learn a generalized relationship between input and target variables, it is desired to find input variables that are highly predictive of the target variable(s). Hence, in a binary problem, input variables are desired that maximize the separability of the samples into the two distinct classes. Some of the studies that are discussed in the previous sections, have evaluated the effect of different data types on prediction performance. The general consensus is that dynamic session data is more effective in predicting future behaviour than static customer data, whereas a combination of both leads to the best results in most of the cases [28, 105, 111, 136, 141]. The next paragraphs review the type of explanatory variables that is used for various applications concerning the classification and prediction of customer behaviour.

A widely researched methodology is market or customer segmentation for classifying customers into distinct groups. Its objective is to provide better understanding about the overall composition of customers, including their characteristics and purchase behaviour. The first step and crucial part of segmentation is to identify attributes that can measure dissimilarities (known as a distance measure) in order to distinguish customers from each other [27, 69]. Followed by creating analogous segments based on these attributes which is known as the methodology of clustering. Dividing a market into segments can be done by various approaches, however at their core, all methods are based either on descriptive or behavioural attributes [29, 66, 82, 139]. Descriptive attributes (such as age, sex, size or location) are commonly used since this type of variables is easy to quantify. Whether segmentation based on descriptive attributes is successful, depends to large extent on the relevancy of the attributes for defining the segments and the availability of supporting data of those attributes [5]. In customer relationship management, behavioural data is often seen as very effective since it carries a lot of predictive value [111]. Segmentation based on behavioural attributes not only requires historical data that describes the behaviour of customers, but also a method for extracting and identifying it within the data [10, 77]. There are many external factors, such as seasonality, economy, competitors' actions, and social perception, that influence customer behaviour. However, selecting appropriate factors can be challenging [115].

Many customer behaviour studies investigated the application of customer demographic variables to analyse and predict customer behaviour [103, 122]. Whereas, behavioural variables, such as recency, frequency, and monetary (RFM), are found to have high value in discriminating customer contributions to a business [93]. The variables, respectively, measure the recency of customer purchasing behaviour, the frequency of purchasing, and the average monetary expenditure on purchasing. In literature it is widely observed that the use of the RFM attributes for customer behavioural analysis can effectively identify customer values and segment markets [33, 81]. Buckinx and Van den Poel [141] use several classification techniques to build partial defection models in grocery retail and examine behavioural antecedents, demographics and perceptions as input variables.

With the ever evolving technologies in data systems another source of information for predicting purchase behaviour has become available, *clickstream* data (event data). Bucklin et al. describe clickstream as the path taken by a user through one or more websites [24]. The detailed stream of information allows retailers to follow and understand the decision-making process of their customers in the online environment. Several studies support the findings of event variables having a positive effect in the context of predicting online purchase behaviour. Padmanabhan et al. [103] predicted the probability that the remainder of a visit results in a purchase and if that user would make a purchase in any future session. Additionally, they demonstrate that including user-centric clickstream data outperforms models that are built on site-centric data only. Moe and Fader [94] conducted some research on developing a model for evolving visiting behaviour based on Internet event data. By analysing the conversion of store visits into purchases based on historical visiting data, for each customer predictions can be made concerning the probability he or she will make a purchase during the next visit. They found evidence that supports the notion that people who visit a retail site more frequently have a greater propensity to buy. In their empirical study, Van den Pool and Buckinx [136] examine a wide variety of variables, including customer demographics, historical purchase behaviour and (detailed) event data. Also included are features such as the number of products viewed and whether the search engine is used, which both are shown to have a significant effect on the purchase probability. The results show that predictors from all categories are retained in the subset of variables with most predictive power. According to Bucklin and Sismeiro [120], even the sequence of different actions taken by a visitor has some predictive value about the probability of placing an order. Since most online stores provide a virtual shopping cart to assist customers to collect items of their interest, such useful information can be exploited and may be a strong predictor for online-purchasing behaviour. Close and Kukar-Kinney [38], as well as Tang et al. [130] found supportive results, indicating that online behaviour regarding shopping cart usage significantly increases the probability of a visit resulting in an actual order. The results of prior research mostly rely on computer-based features. However, in the recent years mobile commerce is emerging at fast pace and with it an even richer data source becomes available, which has the potential to result in new possible features. Recent research by Cardoso et al. [28] regarding customer lifetime value prediction based on customer demographics, returns history, purchase history and web/app session logs, reveals that the latter two among the variables were most successful in predicting customer behaviour.

2-3 Literature Gap

Even though the general idea of the above-mentioned studies and the problem at hand are similar, namely the prediction of online customer behaviour, there are some major differences which will be researched within this thesis study. Whereas most studies are conducted in the field of conventional e-commerce, this research will focus on the online grocery retail market. Instead of predicting, for example, the resulting purchase probability of the remainder of a session based on clickstream data, or the likelihood that a customer will churn based on their past behaviour, this study aims at the estimation of customers' daily purchasing probabilities. Within this application, the effect of different types of explanatory variables on model performance will be investigated. The objective of all the reviewed studies involves a single binary target variable, whereas the approach proposed in this thesis requires the extension to a model that can handle multiple outputs, namely one for each horizon. Finally, and most importantly, the obtained purchasing probabilities will be evaluated in the context of aggregated forecasting performance for a group of customers.

Chapter 3

Data

For the analysis employed in this project, Picnic provided access to their data warehouse. Picnic stores all sorts of dynamic data, such as transactional data, and static data in tabular format in a relational database. Transactional data is a set of sequential timestamped events that represent interactions between customers and companies, where most represent purchases. The data in Picnic's data warehouse ranges from detailed customer and purchase data, application and operational systems data, over to demographic and geographic data. This enables analysis of the complete purchasing history of all customers and any additional data that is available and may be valuable for this research. At midnight all data of the previous day is gathered, structured, polished and finally stored in the database, allowing access to fresh data every day. For the sake of this project, Picnic's database is used as only data source for the analysis performed in this project. Therefore, no real time data is fed into the models.

In section 3-1 the potential explanatory variables are discussed that are considered as an input for the models. Thereafter, in section 3-2, some necessary pre-processing steps are explained that are performed before the actual modeling procedure.

3-1 Explanatory Variables

It is investigated whether the different methodologies found in literature, involving various types of explanatory variables, can be of use in order to form the basis for a propensity model in the context of an online grocer. Besides analysing customer characteristics and purchase behaviour to distinguish customer groups and estimate next-buy probabilities, this methodology also aims for clear identification of customer (weekly) purchase patterns. The latter is of great importance, since the goal is to obtain accurate predictions for each customer on a day level. As in literature suggested, this project will focus on the analysis of historical customer purchase behaviour data, event data and descriptive customer attributes, to explore their potential in identifying future purchasing behaviour. For modeling purposes, all information that is known from the customers' first purchase on, up to the cutoff (time of performing the prediction), is included in a set of independent variables as time in-/variant features (see Figure 3-1). Note that this only includes data that is gathered till the end of the day prior to the cutoff date, as following data is not known at that time.

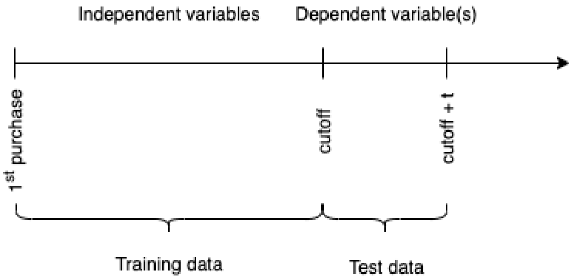


Figure 3-1: Schematic representation of the period of analysis.

Therefore, four main sets are considered for retrieving data that possibly contains explanatory power. The first set incorporates descriptive information about individual customers including the identification (id) number, demographics and a summary of the customers purchase history. In the second table detailed information about every purchase is stored, such as the purchase id, customer id, the date of order creation and date of delivery, the time window of the delivery, the size and value, the satisfaction rating and many more. The third set captures all kinds of events concerning application usage aggregated to sessions, hence any interaction between customer and application. It includes information about the type of event (views, searches, product added etc.), session start and end times and to which customer it belongs. The final set includes all kind of meta data, such as time dependencies, holiday dates and meteorology data. The following sections provide explanation of the various explanatory variables that are used for the analysis.

3-1-1 Customer Demographics

The customer demographics considered in this project include the household composition, their geographical location and some statistical information regarding the neighbourhood of a customer. These features are used for heterogeneity purposes, which are supposed to help identifying different behaviour among customer groups. Subsequently, this is expected to enhance predictive performance for new customers where little purchase history is available, by matching their profile with customers of similar type.

At first, the household composition is included as it is known from the data that families show a different behaviour compared to non-families. Especially during (school) vacations families are more likely to go on holiday and therefore the probability of them placing an order during that time decreases. Furthermore, whether a customer owns a pet can tell various things about his or her personality. Results by James et al. [73] suggest significant differences among those who own only dogs, only cats, dogs and cats, and non-owners. Furthermore, pet owners are expected to go less (or for a shorter period of time) on vacation, since they need to take care of the pet. Hence, the household composition is represented by four variables; the number of adults, the number of children, the number of dogs and the number of cats. To distinguish business users from regular customers, an additional binary variable is introduced. Moreover, the geographical location is included by the latitudinal and longitudinal coordinates in order to account for local behavioural trends and customer habits. Finally, attributes concerning some statistics about the neighbourhood that a customer lives in are also incorporated. These are expected to contain some information about the type of customer. They embody the number of households, the average number of cars per household, the average income per inhabitant, and the number of supermarkets within a range of 1 kilometer.

3-1-2 Customer Purchase behaviour

Customer purchasing behaviour is represented by multiple predictors, that embody recency and frequency of purchases, amount of money spent, inter-purchase times, length of the relationship between customer and company, promotional behaviour and customer satisfaction, as encouraged by many researchers (subsection 2-2-2).

Recency, Frequency and Monetary

At first, the RFM variables are included. The recency variable is a temporal measure that implies how recently a customer has made a purchase. Among the RFM variables, this is considered as the most powerful predictor of customer future behaviour [93]. In the case of partial churn prediction, a lower value corresponds to a higher probability that the customer will remain using the service [136]. In this study, recency indicates the number of days between the previous purchase and the time of prediction. It is expected that it carries strong predictive value regarding the likelihood of a next purchase. Frequency is represented by the number of purchases pursued within a certain time window and indicates the strength of the customer relationship with the company. A high frequency of purchases at a particular company naturally conforms with a high loyalty towards it [133]. Therefore, frequency is expected to be a predictor of future behaviour, especially in terms of churn prediction [17]. Frequency is included in several ways here; (1) the number of total purchases, (2) the number of purchases last week, 2, 3, 4, 5 and 6 weeks ago, and (3) the number of purchases per weekday within the last 8 weeks. The latter two are a measure of recent frequency, which are more representative for the customer's current state of interest. Moreover, feature (3) is also an indicator for the preferred weekday(s) of the customer. Monetary is the amount spent by the customer and is seen as the least powerful variable of the three, although it is still noticed as being valuable in cooperation with the others [111]. The variable is covered by the total and average amount spent.

Interpurchase Time

There is also evidence that the temporal relation between purchases can provide additional predictive value in cases where logical orders of purchases exist [117]. From the data analysed in this study it is found that in the majority of cases customers tend to order in patterns. Therefore, it is assumed that the period between purchases has high predictive value on when the next purchase will be, especially in conjunction with the recency variable *number of days* *since previous purchase*. It is expected that the median of the observed interpurchase times is more appropriate than the average in this context, as one is interested in the usual period between purchases. However, irregularities like missing a week out (or more) due to vacations or other reasons would result in a misleading value. Therefore the variable is represented by the median of the observed interpurchase times. In order to deal with varying periods (e.g., a customer makes a purchase every 3 and 4 days), its standard deviation is incorporated as a feature as well.

Since for the majority of the customers the shopping behaviour conforms to a noticeable pattern, the interpurchase time can be viewed as cyclical. In order for a machine learning algorithm to recognize that a variable is of cyclical nature, its discrete form needs to be transformed into a continuous form. Using sine and cosine trigonometric functions the interpurchase time can be transformed into two continuous periodic variables that indicate whether the prediction date is close to the expected purchase date, while taking time dependencies into account. For each customer, the sine and cosine transformations of the interpurchase time at time t are given by

$$\begin{aligned} \alpha_{(t)} &= \sin\left(2\pi\frac{d}{T}\right) \quad \& \\ \beta_{(t)} &= \cos\left(2\pi\frac{d}{T}\right), \end{aligned} \tag{3-1}$$

where d is equal to the number of days since the previous purchase and T is the interpurchase time.

Length of Customer Relationship

The length of customer relationship is defined as the number of days since invitation till the day of prediction. It is used to distinguish new customers from more mature ones.

Promotional behaviour

Many studies investigating the effect of loyalty programs and short-term promotions on customer retention found encouraging results [87]. Since Picnic also uses similar tools, it is interesting to see whether such variables have predictive value on future customer behaviour. Variables such as average and total discount, total discount in the last 5 weeks, number of promotions in the most recent order and the number of gifts in the most recent order, are all included in this study.

Customer Satisfaction

It is evident that customer satisfaction plays a big role in building good relationships with customers and that it has positive effect on customer retention. Bad experiences result in less trust towards the company and can finally lead to customer churn [63]. Therefore, it is assumed to be a predictor for future purchase behaviour. The satisfaction level is represented by the average rating and the rating of the most recent order on a scale 1 to 10, where 1 is worst and 10 is best. Additionally, the total number of incidents concerning a delivery (e.g.

missing products, freshness issues etc.), and number of incidents in the last order are included as variables. Since one of Picnic trademarks is on time delivery within a small time window, two more variables are considered regarding the on time of a delivery. The first one covers the lateness in minutes of the most recent order, whereas the second one covers the average lateness in minutes up to prediction date.

Placed Purchases

Since customers are able to place orders up to 14 days ahead, a (small) fraction of orders is already known before time of prediction. In order to incorporate them directly into the model rather than first excluding them and adding them again to the prediction afterwards, the number of days till the known date of delivery is included as a feature.

3-1-3 Event Data

In comparison to past research, which is mostly web-based, this study considers mobile application data as only source for event related features. While certain features that are shown to be important for predicting customer purchasing behaviour (such as the total number of products viewed [136]) are not available, some others still can be implemented. First, it is desired to incorporate the frequency of visits, like suggested by Moe and Fader [94]. This is done by counting the total number of unique sessions per customer since the previous purchase up to the moment of cutoff. Moreover, recency is also included as feature for the event data, since this is shown to have a significant effect on online purchase probabilities [136]. In this study, the variable represents the number of days between the date of the last visit and the cutoff date. Another promising feature noted in those studies, is the time that a customer has spent during the sessions. Therefore, the variable representing the total number of seconds spent in the app since the previous purchase, is added to the features as well. Similarly, the total number of clicks, that also seem to have significant predictive value [136], are covered by an adjusted feature that counts the total number of events since the previous purchase. As suggested by Close and Kukar-Kinney [38] and Tang et al. [130], the behaviour concerning the usage of the virtual shopping cart can tell a lot about future purchasing behaviour. Therefore, this study incorporates the number of products in the basket as a feature. Finally, since recency tends to be such a great predictor, all of the variables above are additionally adjusted to variables that summarise the behaviour of just the day prior to the date of prediction. This describes the most recent behaviour of the customer that is known at the moment of cutoff.

Since Picnic makes use of designated time slots per delivery area, these are select-able items in the application. A customer can choose to select it whenever he or she likes but prior to checkout. Most of the customers do it after they filled the basket with groceries and are ready to place the order. However, there is a fraction of customers that "reserves" a time slot up to a couple of days before checkout, as they probably already know when they want to make the (next) purchase. Therefore, the number of days till the most recent selected time slot is assumed to be a great predictor of their future purchase behaviour.

3-1-4 Capacity Constraints

A big challenge in forecasting within Picnic are capacity constraints. Regularly, the capacity of the supply chain does not meet the actual demand of certain delivery areas or regions, with the result that the store closes before regular closing time for that specific area or region. Within Picnic this is defined as a "slot closing", where slot refers to a time slot that is available dependent on the region a customer lives in. A slot closing can lead to "overflow" of orders to the following day(s), or may even result in customers placing no order at all that at first had the intention to. Either way, it forces the behaviour of customers to change, which can have an one-off effect or an effect that remains for longer period of time. For example, a customer that orders weekly has decided to place an order for the day after, then could adapt his or her future behaviour to that new weekday or fall back to the initial day of preference. Similarly, a customer could decide to simply not order anymore at all, or just not this time only and order the following week again. There are several possible outcomes to this situation.

The days with slot closings are incorporated for each weekday, for each customer dependent on their location. Using this, in combination with the daytime that a customer usually orders, it is attempted to identify whether a customer was likely to be affected by the early closing or not. It is expected to provide more information for future behaviour. For example, it could tell the model not to give too much weight on the features concerning customer periodicity in case a customer was forced to change behaviour due to a slot closing. Hence, the first variable expresses the average create time of purchases of a customer, by measuring the difference between the time of purchasing and regular closing time of the store and taking the average. The other variables embody the minutes that a slot closed before regular closing time per weekday.

3-1-5 Meta Data

In order to cope with weekly and yearly seasonality some temporal measures are included in the feature set as well. The first variable that is included represents the weekday at cutoff. Since customers usually live in weekly routines due to work and other returning activities, they often have their preferred weekdays for delivery of their groceries. With similar reason the calendar week is incorporated as well, to account for yearly seasonal effects. Both variables are transformed into their two periodic variants using Equation 3-1, with t the weekday and calendar week, and T the number of days in a week and number of weeks in a year, respectively. Together with two encoded variables that indicate whether the date of prediction is a (summer) holiday, it is attempted to deal with seasonal trends and changing behaviour.

Furthermore, some meteorological information such as precipitation probability, temperature and cloud cover on cutoff day are included to investigate whether this affects people's online shopping behaviour.

3-2 Pre-Processing

The first and one of the most important steps is to ensure the quality of the data that is used for analysis [62]. The quality of data is represented by five key characteristics: validity, accuracy, completeness, consistency and uniformity. The data should conform to defined business rules and constraints, and should represent values that are close to reality. If a large proportion of observations have missing values, it will have a negative effect on the statistical power. Same holds for inconsistent data sets, e.g. when two or more values in the data set contradict each other. Finally, the data should contain the same unit of measure across the board in order to avoid misleading results.

Therefore, pre-processing is often necessary before modeling. Pre-processing involves data cleansing and transformation. Data cleansing is the methodology of missing value imputation, outlier detection, and replacement. In many applications, transformation of the data is necessary as the applied techniques rely on statistical assumptions, like unit variance, zero-mean and (standard) normal distribution.

3-2-1 Cleansing

At first, internal Picnic customers along with customers that did not place an order yet, are excluded from the data set. Next, the set is scanned for missing values and outliers. Fortunately Picnic does a great job in keeping the data that they gather clean and as complete as possible. Therefore, very little cleansing have to be performed. The variables concerning the household size (e.g. number of adults, children, cats and dogs), however, do contain some missing values as well as data that seems to be wrongly reported by some customers. Missing values and outliers are imputed by the value -1 to indicate that the customer did not or likely not correctly fill in the questions at sign up. For the number of adults a threshold of 20 is chosen; hence for any non-business user a value larger than that is considered as an outlier. For the other variables a threshold of 10 is considered. This technique is chosen over conventional techniques (e.g. mean imputation or imputation based on feature similarities), as these techniques are less suitable in this case. Since the main purpose of these attributes is to distinguish customer groups, mean imputation would naturally work in a contradicting way. Although, imputation based on feature similarities would be an appropriate approach, the other features are found not strong enough for proper identification. Therefore the imputed values would result in biased parameter estimates due to low prediction accuracy. Furthermore, some earlier studies suggest to introduce a (binary) variable that indicates whether some information (e.g. age, phone number, mailing address, etc.) was provided by the customer or not, as this may indicate certain level of trust towards the company [136].

In univariate time series modeling, missing value imputation is required in order to maintain the temporal relation between observations. Since there are some days for which Picnic does not deliver any groceries (e.g., Christmas, new years day etc.), there are some dates without recorded purchases in the data base. Leaving the values at zero would bias the outcome of the model, which suggests the imputation of the missing values. This can be done in several ways (e.g., by mean, median, mode, etc. imputation, linear interpolation , etc.), however there is no *good* way to deal with them [106]. For the sake of this research it is chosen to predict the missing values using the model itself and use the predicted values for model training that follows.

3-2-2 Transformation

Since in many applications the range of raw data varies widely, the objective function of many machine learning algorithms would fail in obtaining proper results without scaling the data first. For example, many classifier models calculate the distance between two points using some distance metric (e.g., Euclidean distance). If one of the input vectors contains values with a much broader range than the other vectors, then the resulting distance will be governed by that particular feature. Hence, in such cases it is favorable to re-scale the data to unit variance so that each feature contributes approximately proportionately to the final distance calculated by the objective function. Besides metric-based methods, scaling is also useful in gradient-based applications as the gradient descent algorithm converges much faster when features are scaled to unit-variance [72].

When handling data that includes variables with multiple dimensions, feature standardization should be applied before modeling in order for some machine learning algorithms to properly work. By subtracting the mean in the numerator and dividing by the variables standard deviation, it transforms each feature to have zero-mean and unit variance. This method is widely used for data transformation in many machine learning applications (e.g., support vector machines, logistic regression and artificial neural networks) as these assume approximately standard normally distributed data (e.g., Gaussian with zero-mean and unit variance) [61]. The most common method is the z-score transformation to obtain a variable z' with zero mean and unit variance [79]:

$$z' = \frac{x - \overline{x}}{\sigma} \tag{3-2}$$

where x is the variable to be transformed, $\overline{x}$ the mean of that feature vector, and σ is its standard deviation.

A notable exception are decision tree-based estimators that are robust to arbitrary scaling of the data since they rely on rules instead of distances. Any monotonic transformation of variables would not affect the outcome as the relative order of a variable is maintained post scaling [55].

In conventional time series modeling, data transformation is required to convert non-stationary time series into stationary time series. A stationary time series has no seasonality and no trend, and complies with statistical measures like mean, variance and autocorrelation [65]. A non-stationary time series can be made stationary by differencing [70].

Chapter 4

Methodology

As mentioned in the introduction, the interest of this research lies in predicting the daily purchasing probabilities of each customer, with the final goal to improve aggregated demand forecasting of an online grocer. In this section, an approach is proposed which implements three different classification algorithms that map a set of independent variables to a set of dependent variables in order to make predictions for the future. The dependent variables are of binary nature that for each customer for each horizon reflect whether a purchase is made (1) or not (0). The output of the models are individual customer purchase probabilities. For a group of customers, the obtained probabilities are then aggregated to the total level and compared to a top-line time series model. Within this research, the top-line model is defined as a model that is fit to the aggregated time series data of the total number of customers within a group. For the sake of this research, it is decided to evaluate the predictions on individual customer level as well as on a total level, within the next week (e.g. a horizon of 7 days). This enables Picnic to use the results for short-term planning purposes.

At first, an explanation of the proposed learning algorithms along with their mathematical formulations are given in section 4-1. Followed by the extension methods that adapt the single binary classification models in order to handle multiple labels in section 4-2. Thereafter, the processes of feature selection as well as hyperparamter tuning are discussed in section 4-3 and section 4-4, respectively. Then, the final modelling step is explained, namely the methodology of probability calibration in section 4-5. This chapter comes to an end with topics concerning model evaluation in section 4-6 and subsection 4-6-4.

4-1 Learning Algorithms

One of the most well-known and widely applied techniques for probability estimation in a binary choice model is logistic regression [103, 136]. This straightforward parametric model is known for its easy implementation and interpretability. On the down side, however, its simplistic modeling assumptions may lead to underfitting for rich and complex data sets. The logit algorithm will be used as a benchmark model for the classification algorithms. Literature suggests that there are several other non-parametric approaches which can be used for solving binary classification problems. In the following sections, two tree-based ensemble methods are discussed that have proven to be superior over logit regression and other machine learning methods in the context of various applications in different domains [86, 84, 91, 136, 143]. A detailed explanation of these models, namely the methodology of Random Forest and Stochastic Gradient Tree Boosting, will be given along with some of their advantages and disadvantages. The other methods discussed in subsection 2-2-1 are excluded from further analysis, as they either did not show enough evidence of being fit for the job (KNN), do not directly output probabilities (SVM and KNN), require careful hyperparameter tuning (SVM and FNN), have low interpretability (SVM and FNN) or exhibit long training times (SVM and FNN). The latter is an issue in the context of this research only, as modeling and evaluation of the models take a considerable amount of time.

At first, however, a general description of the top-line model is given in subsection 4-1-1. Since e-grocery retail is a 'relatively' new concept, not much literature is available about customer demand forecasting in this area. Therefore, the time series at hand is compared with time series of other problems in the domain of forecasting. Some similarities are recognized between the series of (short-term) load demand and the demand of an online grocer. In particular, the daily load pattern show some analogies with the weekly demand pattern at hand. Accordingly, the model described in subsection 4-1-1 is selected as benchmark model in the context of forecasting. The model is widely applied to load forecasting problems due to its accuracy and mathematical soundness [39, 64]. It is fairly easy to implement and provides good results in case the series does not exhibit any extreme nonlinear, volatile behaviour, which is not the case. The seasonal adaptation to the model enhances its ability in modeling any seasonal and recurrent pattern [95, 134]. Additionally, the model has the advantage of being highly interpretable while not requiring any tedious feature engineering.

4-1-1 Seasonal Autoregressive Integrated Moving Average Exogenous (SARIMAX) model

The ARIMA model is among the most applied statistical methods for *stationary*, univariate time series problems. The application of the model is straightforward and they usually provide satisfying results. As the name already suggests, the ARIMA model consists of three parts in order to model time series data for forecasting (e.g., predicting future points in the series) [65].

- The *auto-regressive* (AR) part accounts for a pattern of growth/decline in the data.
- The *integrated* (I) part accounts for the rate of change of the growth/decline in the data.
- The *moving average* (MA) part accounts for the noise between consecutive time points.

A time series is stationary if its statistical properties are all constant over time. In other words, it has no trend and no seasonality, meaning its variations around its mean have a constant amplitude and its short-term random time patterns resemble in a statistical sense. The latter condition means that its autocorrelations remain constant over time, or equivalently, that its power spectrum remains constant over time. A random variable can be viewed as a combination of signal and noise, where the signal could be a pattern of fast or slow mean reversion, sinusoidal oscillation or rapid alternation in sign, as well as have a seasonal component. If a time series is non-stationary, it can be made stationary by differencing, and if necessary in conjunction with nonlinear transformations such as logging or deflating. An ARIMA model can be viewed as a *filter* that tries to separate the signal from the noise, and the signal is then extrapolated into the future to obtain the predictions [70].

The equation of the ARIMA model for a stationary time series is a linear regression-type equation, in which the time series can be modeled as a combination of past values Y_t and past errors ϵ_t, also known as *lags*. Lags of the stationarized series in the forecasting equation are called auto-regressive terms, lags of the forecast errors are called moving average terms, and a time series which needs to be differenced to be made stationary is said to be an integrated version of a stationary series.

Let $Y = [y_1, ..., y_n]^T$ be a series of observations, then the differenced ARIMA(p, d, q) model at time instance t is expressed as

$$y'_t = \zeta_1 y_{t-1} + \zeta_2 y_{t-2} + \ldots + \zeta_p y_{t-p} + \epsilon_t - \phi_1 \epsilon_{t-1} - \phi_2 \epsilon_{t-2} - \ldots - \phi_q \epsilon_{t-q},$$

where ζ_t and ϕ_t are the parameters of interest and p and q the orders of auto-regressive and moving average polynomials, respectively. The model parameters ζ_t and ϕ_t are estimated using *maximum likelihood estimation* (MLE). The goal of MLE is to find the value $\hat{\theta}$ that maximizes the likelihood function $L(\theta)$ over the parameter space Θ (Equation 4-1).

$$\hat{\theta} = \underset{\theta \in \Theta}{\arg\max} \hat{L}(\theta; Y), \tag{4-1}$$

with $\theta = [\zeta_1, ..., \zeta_p, \phi_1, ..., \phi_q]$. The likelihood function is equal to the joint probability distribution of observations in the series. Typically, the log-likelihood (Equation 4-2) is maximized which approximates the exact solution while being computationally more efficient.

$$\ln \hat{L}(\theta; Y) = -\frac{n}{2} \ln(2\pi) - \frac{1}{2} \ln \det(\Gamma_n) - \frac{1}{2} Y^T \Gamma_n^{-1} Y, \tag{4-2}$$

with Γ_n the autocovariance matrix.

ARIMA expects data that is either not seasonal or has the seasonal component removed (e.g., seasonally adjusted via methods such as seasonal differencing). In order to support time series data with a seasonal component, the ARIMA model can be extended to a seasonal model [70]. Such a model can be represented as ARIMA $(p, d, q) \times (P, D, Q)_m$ (SARIMA), where P, D and Q represent the coefficients for the seasonal part of the time series and m denotes the number of periods within each season. "The seasonal part of the model consists of terms that are very similar to the non-seasonal components of the model, but they involve backshifts of the seasonal period" [70, page 242]. However, the time series exhibits both weekly and yearly seasonality, which the SARIMA model is not able to account for. Therefore, to incorporate the double seasonality, additional Fourier terms are added to the SARIMA model in terms of exogenous regressors (SARIMAX).

$$y'_t = \sum_{k=1}^{K}[\sin(\frac{2\phi kt}{p}) + \cos(\frac{2\phi kt}{p})] + N_t, \tag{4-3}$$

with N_t the SARIMA process, p the period and K the corresponding number of Fourier terms.

4-1-2 Logistic Regression

Logistic regression (or logit regression), despite its name, is a linear model for classification rather than regression. Logistic regression, which first application was introduced by Berkson [13], is a statistical model that uses a logistic function to model the binary dependent variable. The log-odds (logarithm of the odds) is modeled as a linear combination of the independent variable(s), which can be both binary or continuous. The logistic function transforms the log-odds to the corresponding probability that can vary between 0 and 1 using the sigmoid function. The unit of measurement for the log-odds scale is called a logit (logistic unit). The generalized logistic model is defined as stated in Equation 4-4.

$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}} = P(Y = 1|X; \theta), \tag{4-4}$$

where Y is the dependent variable, X the design matrix of independent variables and θ represents the regression coefficients that are estimated based on the mapping between input variables and the output variable [78]. Similar as in the ARIMA case, the parameters are estimated using MLE (Equation 4-1).

Since $Y \in \{0, 1\}$, $P(y|X; \theta) = h_\theta(X)^y(1 - h_\theta(X))^{(1-y)}$. Then, assuming that all the observations in the sample are independently Bernoulli distributed, the likelihood function is given by

$$\begin{aligned} L(\theta|x) &= P(Y|X; \theta) && (4\text{-}5)\\ &= \prod_{i=1} P(y_i|x_i; \theta) && (4\text{-}6)\\ &= \prod_{i=1} h_\theta(x_i)^{y_i}(1 - h_\theta(x_i))^{(1-y_i)} && (4\text{-}7) \end{aligned}$$

In this case, a logit model is implemented that uses the *SAGA* algorithm for optimisation to find the optimum of the likelihood function. *SAGA* is inspired by both *SAG* [113] and *SVRG* [75] and is an incremental gradient algorithm with fast convergence rates, which therefore makes it a suitable algorithm for large data sets. For more information and the theory behind the algorithm, this research refers to the work of Defazio et al. [48].

4-1-3 Decision Tree Ensemble Methods

Classification and Regression Trees (CART) were first introduced by Breiman et al. [20]. They are based on the Decision Tree (DT) algorithm and form the basis for the ensemble methods Random Forest and Stochastic Gradient Tree Boosting.

Due to its ease of use and interpretability, the DT has evolved to a popular concept among researchers and analysts [84], and comes closest to meeting the requirements for serving as an *off-the-shelf* procedure for data mining [55]. As the name already insists, DTs use decision rules in order to obtain the dependent variable from the set of independent variables. Given training vectors $x_i \in \mathbb{R}^n, i = 1, ..., n$ and a label vector $y \in \mathbb{R}$, a decision tree recursively partitions the space into a set of rectangles (or their higher dimensional equivalent) such that the samples with the same labels are grouped together [55]. For example, at its starting point, the algorithm divides the feature space $\mathcal{X}$ into two regions by splitting feature X_j at some threshold t (splitting point) [55].

$$R_1(j,t) = \{\mathcal{X}|X_j \leq t\} \quad \text{and} \quad R_2(j,t) = \{\mathcal{X}|X_j > t\}. \tag{4-8}$$

This procedure is then repeated for one or both of the resulting regions and continues S times, until some stopping criterion is satisfied. This will end up in $S + 1 = L$ regions as visualized in Figure 4-1. A tree consists of multiple *nodes* that are connected with each other. The initial node is called *root* and the terminal nodes are called *leaves.* The paths from the root to the individual leaves are known as *branches.*

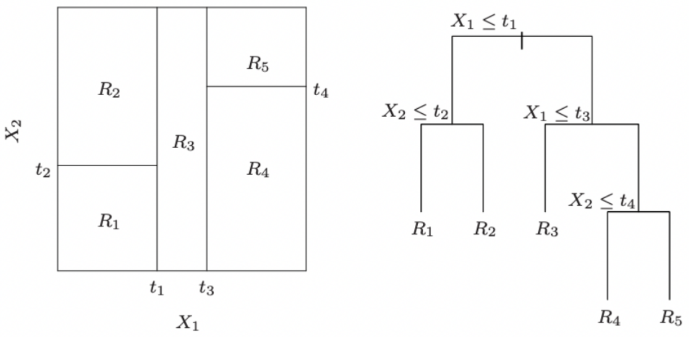


Figure 4-1: Schematic Overview of a partitioned two-dimensional feature space (left) and corresponding decision tree (right) [55, Figure 9.2].

The optimal splitting point t_m for variable X_j in node m is found by minimizing the impurity in both the resulting nodes (child nodes). Impurity simply measures the class distribution within a node. An equal distribution corresponds to maximum impurity, whereas minimum impurity occurs when a node contains all samples of a single class. Hence, the objective of the decision tree is to find the leaves with lowest achievable impurity, indicating that the classes are well separated. Let the data in node m be represented by Q, then for each candidate split $\theta = (X_j, t_m)$ (hypothesis) the data is partitioned into two subsets:

$$Q_{left}(\theta) = (x, y)|x_j <= t_m \tag{4-9}$$

$$Q_{right}(\theta) = Q \setminus Q_{left}(\theta). \tag{4-10}$$

The weighted impurity (loss function) of the child nodes is computed by

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta)), \tag{4-11}$$

using some impurity function $H()$. Then, the optimal parameter is selected by minimizing the weighted impurity:

$$\theta^* = \arg\min_{\theta} G(Q, \theta). \tag{4-12}$$

In classification, the two most commonly used measures of impurity are Entropy and Gini (index). The difference between the two in terms of performance is rather small for tree-based models [129]. However, since it is computationally faster, the Gini index is selected as impurity measure in this research. The Gini index is defined as

$$H(X_m) = \sum_{k=1}^{K} p_{mk}(1 - p_{mk}), \tag{4-13}$$

where X_m is the training data in node m with N_m observations and p_{mk} the proportion of class k observations, given by

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} \mathbb{I}(y_i = k), \tag{4-14}$$

with $\mathbb{I}$ an indicator function, which is 1 when the arguments evaluate to true and 0 otherwise [55].

In order to estimate the class probabilities of a new sample, one simply has to follow the branch from the root until the designated leaf is reached. The class probability is then given by the proportion of that class in the leaf. Finally, the sample is classified (assigned to a class) using *majority voting.*

Decision trees, however, have one major flaw - that is - their high variance. A small change in the data can result in a very different series of splits, hence a complete different tree with different outcome [55]. This is due to their hierarchical structure, as the effect of an error in the top split is propagated down to all of the following splits. To overcome this problem, there are some adaptations to the algorithm while mostly remaining the advantageous characteristics of the decision tree. Ensemble methods combine the results of several 'weak' learners into a more powerful estimator. The idea behind this concept is that, despite the instability of decision trees they are unbiased predictors [55]. Hence, on average decision trees provide 'correct' estimations.

Bagging (Bootstrap Aggregation) is a popular ensemble technique and very effective in reducing the variance of a decision tree. By training a decision tree on several random subsets of the data and averaging the predictions will lead to a more robust result than a single decision tree [55]. The *Random Forest* algorithm is an extension over bagging and is aimed at reducing the correlation between the sampled trees [55]. In addition to taking random subsets of the data, Random Forest also takes a random selection of features rather than using all features to grow the trees.

Another ensemble technique that experienced great attention in both literature and industry is *Boosting.* According to Friedman et al. [55] it is one of the most powerful learning ideas that has been introduced in the last couple of decades. In boosting, a collection of weak predictors are learned sequentially and trained iteratively on residuals. In other words, consecutive trees are fitted on random samples and at every step, the goal is to solve for net error from the prior tree. In case an input is not correctly classified by a hypothesis, its weight is increased so that the updated hypothesis is more likely to classify it correctly.

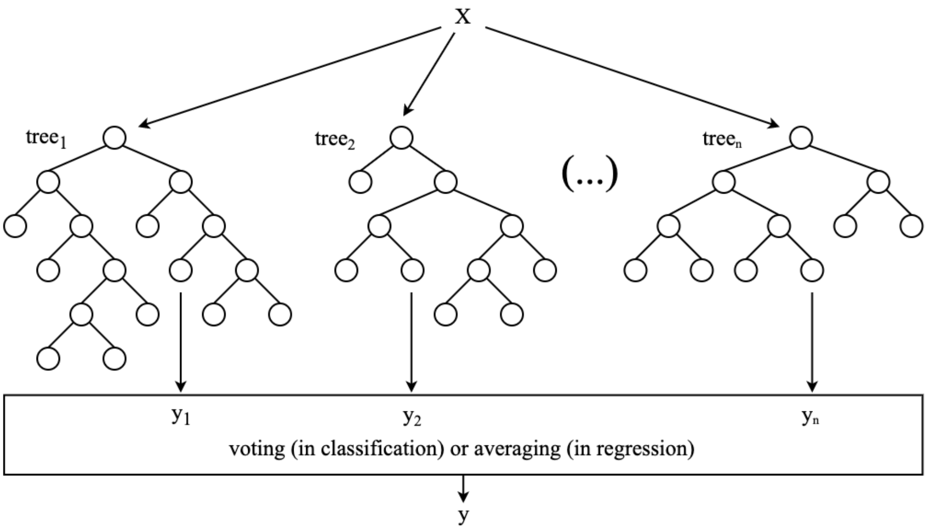


Figure 4-2: Schematic Overview of a Random Forest Model.

Since tree-based ensemble models are built upon the concept of decision trees, they remain some of their favorable properties. They are robust to monotonic transformations, which therefore makes feature scaling redundant (subsection 3-2-2). Furthermore, the models are fairly resistant to outliers and robust to the inclusion of irrelevant features [55]. Additionally, they are able to model interaction effects between variables without explicitly including them in the model. Although single decision trees are more easy to interpret, some variable importance measures for ensemble methods can yet provide some comprehension about the predictive power of each variable. Therefore, these models are more transparent than most other machine learning models, like for example *Neural Networks*.

Yet, compared to logistic regression, ensemble tree models lack interpretability of the relationship between the independent variables and dependent variabele. However, this issue is disregarded since the main focus of this study concerns the prediction performance. Like many other machine learning models, tree-based methods are in need for careful tuning of the hyperparameters to prevent overfitting (or underfitting). This is especially important in the case of gradient boosting, whereas this is less an issue for random forests [55].

Random Forest

The random forest algorithm draws D bootstrap samples with replacement from the data and grows a tree for each sample. The idea is to further improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much [55]. This is done by selecting a new set of features $n_{sel} \leq n$ of the input variables at random at each split as candidates for splitting. The algorithm works as described in Algorithm 1: Random Forest [55].

The class probability estimates of the entire random forest are then simply obtained by averaging all the individual trees. Hence, let $\hat{p}_d$ be the probability estimate for the positive class of a subject of the d^{th} tree, then the class probability estimate of the random forest is

Algorithm 1: Random Forest

```
1. For $d = 1$ to $D$:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size $N$ from the training data.
   (b) Grow a random-forest tree $T_b$ to the bootstrapped data, by recursively
       repeating the following steps for each terminal node of the tree, until the
       stopping criterion is reached.
         i. Select $n_{sel}$ variables at random from the $n$ variables.
        ii. Pick the best variable/split-point at $m$.
       iii. Split the node into two daughter nodes.
   (c) Output the ensemble of trees $\{T_d\}_1^D$.
```

Let $\hat{C}_d(\mathbf{x})$ be the class prediction of the d^{th} decision tree in the forest with $\mathbf{x} \in \mathbb{R}^n$ the vector of input variables, then the prediction for a new $\mathbf{x}$ is given by

$$\hat{C}_{rf}(\mathbf{x}) = \textit{majority vote} \left\{ \hat{C}_d(\mathbf{x}) \right\}_1^D . \tag{4-15}$$

given by

$$\hat{p}_{rf}^D(\mathbf{x}) = \frac{1}{D} \sum_{d=1}^{D} \hat{p}_d(\mathbf{x}). \tag{4-16}$$

Random forests are shown to lead to both improved classification and probability estimates as compared to single decision trees [55]. They are easy to implement and computationally fast. Furthermore, random forests are suitable for probability estimation as long as some tree-building rules are met. For example, the presence of some impurity within the tree must be guaranteed [90].

Stochastic Gradient Tree Boosting

In the methodology of gradient boosting (GB), additive regression models are constructed by fitting simple parameterized functions (base learners) sequentially to 'pseudo'-residuals using a least-squares approach at each step. The residuals are obtained by minimizing the gradient of the loss function, with respect to the model values at each training data point evaluated at the current iteration [57]. Since the models are grown in an adaptive manner, gradient boosting minimizes bias. This is in contradiction to random forests, where only variance is reduced. The implementation of the algorithm follows the procedure described by Friedman [57]. In function approximation, the interest lies in finding some function $f^*(\mathbf{x})$ that maps a vector of independent variables $\mathbf{x} \in \mathbb{R}^n$ to $y \in \mathbb{R}$ while minimizing the expected value of some arbitrary loss function $\Psi(y, f(\mathbf{x}))$ over the joint distribution of all $(y, \mathbf{x})$-values.

$$f^*(\mathbf{x}) = \arg\min_{f(\mathbf{x})} E_{y,\mathbf{x}} \Psi(y, f(\mathbf{x})). \tag{4-17}$$

In boosting, $f^*(\mathbf{x})$ is approximated by using an 'additive' expansion of the form

$$f(\mathbf{x}) = \sum_{b=0}^{B} \beta_b h(\mathbf{x}; \mathbf{a}_b), \tag{4-18}$$

where $h(\mathbf{x}; \mathbf{a})$ (base learner) are in general basic functions of $\mathbf{x}$ with parameters $\mathbf{a} = \{a_1, a_2, ...\}$. The expansion coefficients $\{\beta_b\}_0^B$ and parameters $\{\mathbf{a}_b\}_0^B$ are mutually fit to the training data using a forward stage-wise approach. Starting with an initial guess $f_0(\mathbf{x})$, then for $b = 1, 2, ..., B$

$$(\beta_b, \mathbf{a}_b) = \arg\min_{\beta, \mathbf{a}} \sum_{i=1}^{N} \Psi(y_i, f_{b-1}(\mathbf{x}_i) + \beta h(\mathbf{x}_i; \mathbf{a})) \tag{4-19}$$

and

$$f_b(\mathbf{x}) = f_{b-1}(\mathbf{x}) + \beta_b h(\mathbf{x}; \mathbf{a}_b). \tag{4-20}$$

To replace the potentially difficult function optimisation problem in Equation 4-19, the solution can be approximated using a two-step procedure. First, the base learner $h(\mathbf{x}; \mathbf{a})$ is initialized by fitting least-squares

$$\mathbf{a}_b = \arg\min_{\mathbf{a}, \rho} \sum_{i=1}^{N} [\tilde{y}_{ib} - \rho h(\mathbf{x}_i; \mathbf{a})]^2 \tag{4-21}$$

to the current 'pseudo'-residuals

$$\tilde{y}_{ib} = -\left[\frac{\delta\Psi(y_i, f(\mathbf{x_i}))}{\delta f(\mathbf{x}_i)}\right]_{f(\mathbf{x}) = f_{b-1}(\mathbf{x})}. \tag{4-22}$$

Then, having $h(\mathbf{x}; \mathbf{a}_b)$, the optimal value of the coefficient β_b can be calculated by a single parameter optimisation

$$\beta_b = \arg\min_{\beta} \sum_{i=1}^{N} \Psi(y_i, f_{b-1}(\mathbf{x}_i) + \beta h(\mathbf{x}_i; \mathbf{a}_b)), \tag{4-23}$$

based on the general loss criterion Ψ.

In the case of gradient *tree* boosting, the base learner is an L-terminal node regression tree that at each iteration partitions the $\mathbf{x}$-space into L-disjoint regions $\{R_{lb}\}_{i=1}^{L}$. In each region the tree predicts a separate constant value

$$h(\mathbf{x}; \{R_{lb}\}_1^L) = \sum_{i=1}^{L} \overline{y}_{lb} \mathbb{I}(\mathbf{x} \in R_{lb}), \tag{4-24}$$

where $\overline{y}_{lb}$ is the mean of the 'pseudo'-residuals (Equation 4-22) in each region R_{lb} and $\mathbb{I}$ the indicator function. Now the parameters are equal to the splitting variables and corresponding split points of the b^{th} tree at terminal node l that define the corresponding regions $\{R_{lb}\}_1^L$ of the partition. Now, Equation 4-23 can be solved separately within each region, and since the tree in Equation 4-24 predicts a constant value $\overline{y}_{lb}$, its solution can be reduced to a simple estimate based on the criterion Ψ

$$\gamma_{lb} = \arg\min_{\gamma} \sum_{\mathbf{x}_i \in R_{lb}} \Psi(y_i, f_{b-1}(\mathbf{x}_i) + \gamma), \tag{4-25}$$

with $\gamma_{lb} = \beta_b \overline{y}_{lb}$. Finally, the update of the current approximation $f_{b-1}(\mathbf{x})$ in each corresponding region is given by

$$f_b(\mathbf{x}) = f_{b-1}(\mathbf{x}) + \nu \gamma_{lb} \mathbb{I}(\mathbf{x} \in R_{lb}), \tag{4-26}$$

where $0 < \nu \leq 1$ is the *shrinkage* parameter that controls the learning rate of the algorithm. Hence, every update is scaled by the value of this parameter [55].

For a binary classification problem, the *deviance* is an appropriate choice for the loss function, which is equal to the binomial log-likelihood:

$$\Psi(y, f(\mathbf{x})) = log\left(1 + e^{-2yf(\mathbf{x})}\right), \tag{4-27}$$

with

$$f(\mathbf{x}) = \frac{1}{2} log \left[\frac{P(y=1|\mathbf{x})}{1 - P(y=1|\mathbf{x})}\right]. \tag{4-28}$$

According to Friedman [57], the concept of bagging can also be implemented in the boosting algorithm to incorporate randomness as an integral part of the procedure. Stochastic gradient (tree) boosting replaces the base learner with the corresponding bagged base learner while at each iteration substituting the ordinary residuals with *out-of-bag* residuals. Specifically, at each step a subsample of training data is drawn at random (without replacement) from the full training data set, which then is used to train the base learner and compute the model update. This implementation has shown to lead to both reduced computation time and improved prediction performance [57].

Given the entire training data sample $\{\pi(i)\}_1^N$ and $\{\pi(i)\}_1^N$ a random permutation of the integers $\{1, ..., N\}$, then a random subsample of size $\tilde{N} < N$ is given by $\{y_{\pi(i)}, \mathbf{x}_{\pi(i)}\}_1^{\tilde{N}}$. As proposed by Friedman [57], the resulting algorithm can be described as in Algorithm 2: Stochastic Gradient Tree Boosting.

Finally, by rewriting the log-odds from Equation 4-28 and imputing the final approximation $f_B(\mathbf{x})$, one can obtain the probability estimates [55], given by

$$\begin{aligned} \hat{P}(y=1|\mathbf{x}) &= \frac{1}{1 + e^{-2yf_B(\mathbf{x})}}, \\ \hat{P}(y=0|\mathbf{x}) &= \frac{1}{1 + e^{2yf_B(\mathbf{x})}}. \end{aligned} \tag{4-29}$$

4-2 Multi-label Classification

Due to multiple horizons the problem becomes a multi-output (or multi-label) classification problem. Let $\mathcal{L} = \{1, ..., H\}$ be the output domain of all possible labels, then the output is represented by an H-vector $\mathbf{y} = [y_1, ..., y_H]$, where $y_1 = 1$ if and only if label j is associated with instance $\mathbf{x}$, and 0 otherwise. Hence, for each horizon there is a label with two classes (0 and 1) that the model must predict. Most traditional learning algorithms, however, are developed for single-label classification problems. Therefore a lot of approaches in literature

Algorithm 2: Stochastic Gradient Tree Boosting

```
1. Let $f_0(\mathbf{x}) = \arg\min_{\gamma} \sum_{i=1}^{N} \Psi(y_i, \gamma)$
2. For $b = 1$ to $B$:
    (a) $\{\pi(i)\}_1^N = \text{random\_perm } \{i\}_1^N$
    (b) $\tilde{y}_{\pi(i)b} = -\left[\frac{\delta\Psi(y_{\pi(i)}, f(\mathbf{x}_{\pi(i)}))}{\delta f(\mathbf{x}_{\pi(i)})}\right]_{f(\mathbf{x})=f_{b-1}(\mathbf{x})}, i = 1, ..., \tilde{N}$
    (c) $\{R_{lb}\}_1^L = L - \text{terminal node tree } (\{\tilde{y}_{\pi(i)b}, \mathbf{x}_{\pi(i)}\}_1^{\tilde{N}})$
    (d) $\gamma_{lb} = \arg\min_{\gamma} \sum_{\mathbf{x}_{\pi(i)} \in R_{lb}} \Psi(y_{\pi(i)}, f_{b-1}(\mathbf{x}_{\pi(i)}) + \gamma)$
    (e) $f_b(\mathbf{x}) = f_{b-1}(\mathbf{x}) + \nu\gamma_{lb}\mathbb{I}(\mathbf{x} \in R_{lb})$
3. Output $\hat{f}(\mathbf{x}) = f_B(\mathbf{x})$
```

propose to transform the multi-label problem into multiple single-label problems, hence building H separate models that each independently predict one of the outputs y_j, so that the existing algorithms can be used. This is known as binary relevance (BR). In case there is no underlying correlation between the outputs, this would be a valid and simple solution to the problem. However, since in this case it is most likely that the dependent variables are somewhat correlated with each other, it is suggested to use algorithm adaptation methods rather than problem transformation methods, hence build a single model that is capable of jointly predicting all H outputs.

In this research two approaches for multi-label classification are addressed: the methodology of sequentially chaining multiple estimators (e.g., Classifier Chain model [108]) and the adaptation of single binary classification algorithms.

4-2-1 Classifier Chains

The Classifier Chain (CC) model is a multi-label model that arranges H independent binary models into a chain, where each model separately makes predictions for one of the outputs y_j in a specified order. The difference compared to a BR model, is that each model that follows gets the predictions of all the preceding model(s) as an additional input (see Figure 4-3). Therefore, this model is capable of exploiting correlations among the target values [108]. Clearly the order of the models in the chain is important as the first model in the chain has no information about the other labels while the last model in the chain has features indicating the presence of all of the other labels. Therefore, it is especially an interesting methodology for sequential data (e.g., time series problems). However, extensions to ensembles of classifier chains (ECC) already have been proposed, which greatly enhance performance when the *best* order is not known beforehand [108]. In the context of time series classification, the most convenient order is straightforward. Namely, the forthright ascending order of horizons (in this case 1 to 7), as reliability of predictions should decrease with increasing horizon.

$\mathbf{h}$:	$\mathbf{x} \rightarrow$	$\mathbf{y}$
h_1:	[0,1,0,1,0,0,1,1,0]	1
h_2:	[0,1,0,1,0,0,1,1,0]	0
h_3:	[0,1,0,1,0,0,1,1,0]	0
h_4:	[0,1,0,1,0,0,1,1,0]	1
h_5:	[0,1,0,1,0,0,1,1,0]	0

$\mathbf{h}$:	$\mathbf{x}' \rightarrow$	$\mathbf{y}$
h_1:	[0,1,0,1,0,0,1,1,0]	1
h_2:	[0,1,0,1,0,0,1,1,0,1]	0
h_3:	[0,1,0,1,0,0,1,1,0,1,0]	0
h_4:	[0,1,0,1,0,0,1,1,0,1,0,0]	1
h_5:	[0,1,0,1,0,0,1,1,0,1,0,0,1]	0

Figure 4-3: Example transformation under BR (left) and CC (right) for $(\mathbf{y}, \mathbf{x})$ with binary attribute space [108, Figure 1].

Given a *training sample* $\{\mathbf{y}, \mathbf{x}\}_1^N$, a BR classifier $\mathbf{h} = [h_1, ..., h_H]$ learns whether $\mathbf{x}$ belongs to the j^{th} label (1) or not (0) by independently training h_j on y_j. Hence, the output of $\mathbf{h}$ is a vector $\hat{\mathbf{y}} \in \{0,1\}^H$ for any instance $\mathbf{x}$ [108]. The adaptation of the BR model to the CC model, involves the augmentation of the attribute space for each model by the predicted labels of all previous estimators. The pseudocode of the algorithm is described in Algorithm 3: Classifier Chain [108]. This methodology is applied to the logistic regression model and the stochastic gradient boosting model.

Algorithm 3: Classifier Chain (pseudocode)

```
1. Let $D = \{(\mathbf{x}, \mathbf{y}\}_1^N$ be the training data set
2. For $j = 1, ...L$
    (a) Do the $j^{th}$ binary transformation and training
        $D'_j \leftarrow \{\}$
    (b) For $(\mathbf{x}, \mathbf{y}) \in D$
        i. $\mathbf{x}' \leftarrow [x_1, ..., x_n, y_1, ..., y_{j-1}]$
           $D'_j \leftarrow D'_j \cup (\mathbf{x}', y_j)$
    (c) train $h_j$ to predict binary relevance of $y_j$
        $h_j : D'_j \rightarrow \{0,1\}$
    (d) Classify$(\mathbf{x})$
        i. $\mathbf{x}' \leftarrow [x_1, ..., x_n, \hat{y}_1, ..., \hat{y}_{j-1}]$
        ii. $\hat{y}_j \leftarrow h_j(\mathbf{x}')$
    (e) Output $\hat{\mathbf{y}}$
```

4-2-2 Adapted Algorithms

In literature several approaches have been made to adapt various kind of (binary) classification algorithms to make them capable of handling multi-output problems. Examples of such algorithms include decision tree based methods [31] and neural networks [146]. A popular choice are multi-output ensemble decision tree methods, as they are conceptual 'simple' but have fairly low computation (training) times and competitive accuracy [85]. For example, Clare et al. [36] proposed an adaptation of the C4.5 algorithm, which modification involves the calculations of the entropy splitting criterion.

Multi-output decision tree methods can be implemented on the basis of the induction procedure developed in subsection 4-1-3, by providing two minor changes to the algorithm. At first, instead of assigning one label to each leaf, like in the single binary classification case, the leaves are labeled with multiple output vectors $\mathbf{y_d} = (y_{d,1}, ..., y_{d,H})$. This is done as previously, using the assignment rule (indicator function). Secondly, the impurity decrease of a split is computed by averaging the impurity decrease over the H output variables. Hence, splits are optimised with respect to all output variables, therefore correlations between the labels may be exploited [89]. This methodology is applied to adapt the random forest model.

Thus, one major advantage of multi-output decision trees is the capability to take dependencies between output variables into account, whereas H individual models cannot exploit such correlations. This may result in improving the generalization ability of the model, which subsequently could lead to more accurate results. Additionally, building a single model is often less computationally expensive than building H different models, both in terms of time and space complexity [89].

4-3 Feature Exploration and Selection

Feature selection is considered as one of the core concepts in machine learning that can have a huge impact on the performance of the learning algorithms. Irrelevant or partially relevant features can negatively impact model performance. Less redundant data means less opportunity to make decisions based on noise, which therefore should improve modeling accuracy. Additionally, due to fewer data points, algorithm complexity reduces which results in shorter training time. There are a wide range of selection procedures that can be performed for feature reduction. In this section, a funnel approach is proposed to obtain the final set of explanatory variables (features) that is used to train the learning algorithms discussed in chapter 4. The funnel approach consists of three steps and is visualized in Figure 4-4.

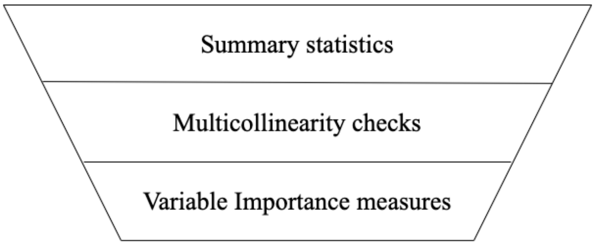


Figure 4-4: Feature Selection Funnel Approach.

4-3-1 Variable Statistics

At first, all data that possibly contains predictive power has been extracted from Picnic's database and transformed into usable features. An explanation of those features is given in section 3-1. Histograms, summary statistics and correlation plots have been created to explore the data quality. Next, multicollinearity checks, such as analysis of correlation tables and the variance inflation factor (VIF), were applied to filter out highly correlated features. In machine learning, variables that are highly correlated, in essence, contain the same information which therefore makes the other redundant. Especially when estimating linear or generalized linear models, multicollinearity is a common problem as it can lead to unreliable and unstable estimates of the regression coefficients [92]. Since a logistic regression model will be implemented as a benchmark model, all multicollinear features are excluded for which both correlation with any other feature exceeds 0.75 and the corresponding VIF is higher than 5, which is often considered as the cut-off value according to Seather [118]. The correlation tables are calculated using the Pearson product-moment correlation coefficient, given by

$$\rho_{xy} = \frac{Cov(x,y)}{\sigma_x \sigma_y}, \tag{4-30}$$

where $Cov(x,y)$ is the covariance, σ_x and σ_y the standard deviations of variables x and y, respectively. The VIF score provides an index that measures how much the variance of an estimated regression coefficient is increased because of collinearity. It is calculated after feature reduction based on the correlation tables in a step-wise procedure where one variable at the time is excluded from the set of features. After exclusion the VIF is again computed and so on till the VIF of all the remaining features is below 5. For feature X_j, it is given by

$$VIF(X_j) = \frac{1}{1 - R_j^2}, \tag{4-31}$$

where R_j^2 is the coefficient of determination when regressing X_j on all other $X's$ [118].

4-3-2 Variable Importance

Final feature selection is performed using variable importance measures. Although, interpretation of the relationship between input and output variables of ensemble tree methods is not as straightforward as with decision trees themselves, an extension to the procedure can provide some transparency. At each node m, one of the input variables $X_{v(m)}$ is used to partition the region associated with that node into two subregions. In the case of single decision trees, the contribution of an input variable to the prediction of the dependent variable then can be inferred by assessing the improvement of the splitting criterion at each split. Breiman et al. [20] proposed

$$\mathcal{I}_j^{2(d)} = \sum_{m=1}^{L-1} \hat{i}_m^2 I(v(m) = j) \tag{4-32}$$

as the squared measure of importance for predictor variable X_j of tree d, where $\hat{i}_m$ is equal to the minimized impurity index in node m. To retrieve the variable importance for the random forest model, one simply takes the average of the decrease in Gini index (Equation 4-11) for each variable over all trees:

$$\mathcal{I}_j^2 = \frac{1}{D} \sum_{d=1}^{D} \mathcal{I}_j^{2(d)}. \tag{4-33}$$

In the case of gradient boosting, a similar approach can be used. By evaluating the effect of each variable in reducing the loss function and taking the average over all trees, will obtain the individual variable importance.

However, both variable importance measures suffer from being computed on statistics derived from the training dataset. Hence, the measured importance of a variable can be high even for input variables that are not predictive of the dependent variable, as long as the model has the capacity to use them to overfit. To overcome this problem, the permutation variable importance measure is introduced. The permutation importance measure evaluates the decrease in model performance when values of a single feature are randomly shuffled [21]. By permuting the values of a single predictor variable at the time, the relationship between that variable and the other predictor variables as well as the dependent variable is broken. Hence, the drop in the model score after permutation provides a good indication of the model dependency on that variable. In case the original variable is not associated to the dependent variable in the first place, the permutation will lead only to a slight random decrease in performance. Or if, by chance, the permutation happens to be more suited in predicting the dependent variable, it may even slightly improve performance. The permutation importance can be computed on any arbitrary data set, such as the training set or a held-out set. By using a held-out set it is possible to examine which variable contributes the most to the generalization power of the inspected model [123]. Thus, rather than analysing the importance in constructing the model, it examines the predictive power of the input variables in relation to unseen data.

In tree ensemble methods, the permutation importance $\mathcal{PI}$ of variable X_j can be obtained by averaging the decrease in model score over all trees D (Equation 4-34) [123].

$$\mathcal{PI}(X_j) = \frac{1}{D}\sum_{d=1}^{D} \mathcal{PI}^{(d)}(X_j), \tag{4-34}$$

where

$$\mathcal{PI}^{(d)}(X_j) = \frac{1}{|\overline{\mathcal{N}}^{(d)}|}\sum_{i\in\overline{\mathcal{N}}^{(d)}} \mathbb{I}\left(y_i = \hat{C}_d(\mathbf{x}_i)\right) - \frac{1}{|\overline{\mathcal{N}}^{(d)}|}\sum_{i\in\overline{\mathcal{N}}^{(d)}} \mathbb{I}\left(y_i = \hat{C}_d(\mathbf{x}_{i,\pi_j})\right) \tag{4-35}$$

is the importance of variable X_j of each individual tree d, with $\overline{\mathcal{N}}^{d}$ the held-out sample, $\hat{C}_d(\mathbf{x}_i)$ the predicted class at observation i before and $\hat{C}_d(\mathbf{x}_{i,\pi_j}$ after permutation of X_j, i.e. with $\mathbf{x}_{i,\pi_j} = (x_{i,1}, ..., x_{i,j-1}, x_{\pi_j(i),j}, x_{i,j+1}, ..., x_i, n)$.

Next to the advantage of assessing the generalization power of the model, the permutation measure accounts for multivariate interaction effects with other input variables [123]. However, one needs to be careful interpreting the results, especially in case the data set contains highly correlated features. For example, when permuting one of two highly correlated features, the performance won't decrease by much as the other variable will compensate for it (e.g. for tree-based methods; a similar split is obtained by using the other feature), which therefore may end up in both features being viewed as irrelevant. Hence, again it is important to avoid multicollinearity of variables. Furthermore, in contrast to impurity-based variable importance for trees, the permutation measure does not exhibit a strong bias towards high cardinality variables in case there are predictor variables of different types [123]. Additionally, it is applicable to any arbitrary (classification) algorithm and provides the possibility to use any arbitrary metric for evaluation. Finally, due to permuting the variable rather than dropping it, the model structure does not change and therefore the model does not need to be retrained for every iteration (permutation). This safes considerable time during the modeling process while obtaining similar effects.

For both importance measures, it holds that the higher the value the higher the importance. Since the permutation variable importance accounts for interaction effects between variables, it is chosen over other popular (linear) techniques, such as LASSO regression, which do not posses that characteristic. In case the measure classifies a feature as irrelevant, it is omitted from the set of input variables.

4-4 Hyperparameter Tuning

After completing the feature selection process, there remains one crucial step in order to boost performance of the models. By finding the optimal set of hyperparameters, the model gets its finishing touch which can help prevent the model from overfitting. Overfitting often occurs in the case of nonparametric and nonlinear models that have more flexibility in learning a target function. A model is said to overfit when it learns the detail and noise in the training data to an extent that it negatively affects performance on new data. Random fluctuations and noise are picked up as concepts that do not apply for new data, thereby decreasing the generalization ability of the model. Hence, hyperparameters are a kind of regularization method which even can lead to underfitting (the model is not able to model the training data at all) when not carefully selected. Therefore, usually a model is preferred that is perfectly balanced between under- and overfitting.

Hyperparameters are model-specific parameters that are not optimised by the model itself, but have to be set 'manually'. In practice, there are many ways which involve various different approaches to find the 'optimal' set of parameters. However, the selecting procedure can be a tedious process that often is considered as nuisance in machine learning. As Snoek et al. [121] state, many perceive hyperparameter tuning as a 'black art' that requires expert experience, rules of thumb, or sometimes brute-force search. In order to be sure that the best model parameters are selected, every possible combination has to be tested and properly evaluated. Thus, the computational complexity increases exponentially with the number of models, parameters and the range of values. Even with a lot of computing resources, this task remains nearly impossible. Therefore, an approach is chosen which involves a Bayesian optimisation to speed up the search process for the best hyperparameters.

Since one does not simply know how well a certain set of hyperparameters will perform, the search process is similar to finding an unknown function which is expensive to evaluate. Bayesian optimisation is a sequential design strategy for global optimisation of such objective functions. By treating them as random functions and using *prior* probability distributions it tries to estimate the shape of the function based on Bayes' theorem. After evaluating a new data point, the prior is updated to the *posterior* probability distribution which is used to construct an acquisition function that determines the next point of search. In other words, the algorithm starts by randomly selecting some sets of hyperparameters over the distribution of possible values and makes predictions for each of them. Then, knowing the performance for each set, a response surface method (RSM) estimates the shape of the function including confidence intervals. Subsequently, more sets of hyperparameters that lie in the most promising areas are sampled for evaluation. At each step the function estimate is updated in order to find the optimal set of parameters. Therefore, this method often reaches the same or even better performance than a brute-force search approach, while requiring fewer searches [121].

The Bayesian optimisation is implemented using the Tree-structured Parzen Estimator (TPE) algorithm as discussed by Bergstra et al. [12]. For the mathematical formulations and thorough analysis of the algorithm, the study refers to their work since this is beyond the scope of the research at hand.

The complete parameter search space considered in the hyperparameter tuning process is given in Table 4-1. The final model configurations along with the AUC scores for different numbers of trees in the random forest model for each horizon are listed in Appendix B.

4-4-1 SARIMAX

The SARIMAX model consists of multiple order coefficients p, d, q, P, D, Q, which along with the number of Fourier terms K and an additional controlling parameter τ for the deterministic trend form the set of hyperparameters. Parameter τ indicates whether to model no trend 'n', constant trend 'c', linear trend with time 't' or both constant and linear trend with time 'ct'. Since the model is intended for short-term purposes, the weekly seasonality is incorporated in the SARIMA part of the model and the yearly seasonality is added to the model as additional regressors.

It is possible to derive the model hyperparameters based on careful analysis and domain expertise, like for example in the Box-Jenkins approach [19]. In most situations this procedure leads to satisfying results, however, not necessarily in every case. It does not only require domain expertise but also can be very time consuming, especially in the case when multiple series have to be predicted. Therefore, Bayesian optimisation is used as an alternative approach to configure the model. This approach has the potential to reveal non-intuitive configurations that result in lower forecast error than through careful analysis, like it is the case in traditional grid search approaches [30], while being considerably faster.

4-4-2 Logistic Regression

For the logistic regression model, the hyperparameters to configure are the type of regularization and the maximum number of iterations that the solver is allowed to take to converge to its solution. In general, the more iterations are performed by the solver the better the performance, as the solver is more likely to converge. However, this does not necessarily hold for the out-of-sample prediction. Beyond some threshold, increasing the value has no practical use or even decreases performance as generalization ability of the model goes down. Thus, optimising this parameter not just decreases computational complexity but also can improve model performance. An additional early stopping criterion is introduced which is regulated using a tolerance parameter *tol*. When the loss score is not improving by at least *tol* for two consecutive iterations, convergence is considered to be reached and training stops.

Regularization is a common technique in machine learning to improve numerical stability in ill-posed problems and to prevent overfitting [25]. The logit model considers three options for regularization. The cost function that is being minimized by a binary class ℓ_1 penalized logistic regression is given by

$$\min_{w,c} ||w||_1 + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1), \tag{4-36}$$

with C the inverse regularization strength. Setting C to a very high value is similar to applying no regularization. Similarly, ℓ_2 regularized logistic regression solves

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1). \tag{4-37}$$

The third method is the Elastic-Net regularization, which is a combination of ℓ_1 and ℓ_2 (Equation 4-38).

$$\min_{w,c} \frac{1-\rho}{2} w^T w + \rho ||w||_1 + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1), \tag{4-38}$$

where ρ controls the strength of one regularization method versus the other. As setting ρ to 1 or 0 is equivalent to either ℓ_1 or ℓ_2 regularization, respectively, the Elastic-Net regularization is selected with C and ρ the hyperparamters to be optimised.

4-4-3 Random Forest

Although hyperparameter tuning is considered not as crucial with random forests as with other machine-learning algorithms [123], slight performance improvements can be obtained by selecting the right values. Next to that, making very poor decisions during the selection process can yet still lead to poor performance of the model.

The number of estimators (trees) D is probably the most important parameter. In general, the higher the number the higher the reliability of the predictions and interpretability of the variables [123]. However, at some threshold the performance converges and increasing the number of trees will not improve the model performance anymore, whereas training time still does. Therefore, this parameter is not included in the Bayesian optimisation but optimised separately first. This reduces model complexity and speeds up further (training) processes. Second important hyperparameter is the maximum number of features n_{sel} that is considered when searching for the best split. In case the number of variables is large, but the fraction of relevant variables small, random forests are likely to perform poorly with small n_{sel}. As at each split the chance can be small that the relevant variables will be selected [55]. However, since a thorough feature selection is performed beforehand, this is not likely the case here. The other hyperparameters that are considered for tuning are the maximum tree depth and the terminal node size (the minimum number of samples required to be at a leaf node). Both of them are *pruning* parameters that reduce the size of the trees by removing sections of the tree that provide little contribution to the classification task. Pruning can be viewed as regularization which can have a smoothing effect by reducing the model complexity. Hence, pruning can lead to improved generalization performance by reducing the chance of overfitting. Whereas some previous studies suggest that each individual tree should be grown as large as possible, others show that full grown trees do not always yield the best results [123]. According to Segal [114], regulating the depth of the tree can slightly improve model performance. Furthermore, Malley et al. [90] argue that for reliable probability estimation there should be some impurity in the trees and suggest the minimum terminal node size as an additional stopping criterion. This notion is supported by Kruppa et al. [80], who explain that a large number of trees are needed when grown to purity in order to ensure consistency of probability estimates. When trees are too small, however, probability estimates get also inaccurate.

4-4-4 Stochastic Gradient Boosting

The boosted tree model and the random forest model share some of the same hyperparameters, however their optimal values can end up being very different from another. For example the tree size (maximum tree depth) is usually smaller in gradient boosting than compared to random forests, where large trees are favorable. In many applications, low-order interaction effects appear to dominate. Hence, models that result in higher-order interaction effects, such as large decision trees, tend to perform worse [55]. For tree-based approximations, the interaction level effects are limited by the number of terminal node regions L. According to empirical studies by Friedman et al. [55], values that lie in the range $4 \leq L \leq 8$ usually provide the best results in the context of boosting.

Whereas for random forests holds that using more estimators (trees) leads to better performance, this is not necessarily true for gradient boosting. Although the training performance reduces with each boosting iteration, increasing this number can in the end lead to overfitting. Hence, setting the number of boosting iterations B is analogous to early stopping strategies in other machine learning applications. The number of iterations however is not the only regularization strategy. The shrinkage parameter ν of the boosting model controls the contribution of the consecutive trees to the current approximation. This can be viewed as the learning rate of the algorithm, where a smaller value results in a larger training risk for the same number of iterations. Therefore, these parameters are operating dependently on each. Friedman et al. [55] suggest to jointly optimise both parameters by taking very small values for ν (≤ 0.1) and select B accordingly. This approach leads to great performance improvements, especially in the case of probability estimation. The minimum loss reduction required to make a further partition on a leaf node γ_{min} and minimum sum of instance weight needed in a child node are two additional parameters which can be tuned for pruning. Increasing these values will make the model more conservative.

The final two hyperparameters that are considered for the boosted tree model, are the subsample ratio of feature columns χ and subsample ratio of training instances η when constructing each tree. Subsampling occurs once in every boosting iteration and shows similar effects as with random forests. Not only does it reduce training time, but often also provides better generalization performance due to variance-reduction [55].

Table 4-1: Hyperparamter search space.

Model	Parameter	Space
SARIMAX		
	$p,\ q,\ P,\ Q$	[0, 1,..., 6]
	$d,\ D$	[0, 1, 2]
	K	[0, 1,..., 5]
	τ	['n', 'c', 't', 'ct']
Logistic Regression		
	max. iterations	[80, 98,..., 800]
	tol	$\{1e^{-5}..1e^{-1}\}$
	ρ	[0.0, 0.1,..., 1.0]
	C	$\{1e^{-4}..1e^{4}\}$
Random Forest		
	D	$\{1..300\}^1$
	n_{sel}	[1, 2,..., 15]
	max. tree depth	[15, 16,...,25]
	terminal node size	$[1e^{-6},\ 1e^{-5},...,\ 1e^{-1}]^2$
Stochastic Gradient Boosting		
	B	[50, 60,...,200]
	max. tree depth	[3, 4,...,8]
	ν	[0.02, 0.04,..., 0.2]
	χ	[0.3, 0.4,...,0.9]
	η	[0.5, 0.6,...,1.0]
	γ_{min}	[0.1, 0.2,...,1.0]
	min. child weight	[1, 2,...,10]

[1] not included in Bayesian optimisation, [2] percentage of all training samples

4-5 Probability Calibration

Ideally, the estimated probabilities of the model reflect the actual purchasing probabilities of the customers. Calibration is a measure for assessing the reliability of the predicted probability distribution in relation to the actual observations [88]. For a given sample of observations with estimated probability $\hat{p}_N$ for the positive class, a classifier is said to be well-calibrated, if the actual proportion of positive observations is equal to that probability. Therefore, the last step that finalizes the models is to examine the quality of the probability estimates and adjust if necessary.

Logistic regression usually yields well-calibrated probabilities as it solves directly for log-loss, whereas probability estimates of decision trees are affected by their high variance [101]. However, due to their statistical properties and variance-reduction, random forests as well as stochastic gradient boosting models provide overall well-calibrated probability estimates. Yet, according to Niculescu-Mizil and Caruana [101], both algorithms have troubles predicting probabilities close to 0 and 1 as they average the predictions over all trees. Hence, all trees have to agree on the probability. However, the high variance of the individual trees make that difficult.

To account for possible non-calibrated probability estimates, certain methods have been developed that can be applied to adjust the output probabilities of the classifiers. In their study, Niculescu-Mizil and Caruana [101] compared two methods, namely Platt Scaling and Isotonic Regression, in the context of various learning methods. They showed that Platt Scaling can improve calibration of boosted trees, while Isotonic Regression can help overcome the variance issue of decision trees. Both methods seem to mitigate the bias of the output by random forests. Whereas Platt Scaling appears to be favorable in cases where the calibration curve is sigmoid and there is limited calibration data, Isotonic Regression seems to perform best when the opposite is the case. Both methods are prone to overfitting when the same data set is used for model training and calibration [101]. Therefore, it is suggested to use cross-fold validation to obtain an independent data set for calibration.

4-5-1 Platt Scaling

Initially, Platt [104] proposed using a sigmoid transformation to obtain calibrated probabilities for a Support Vector Machine classifier. Niculescu-Mizil and Caruana [101] among others demonstrated that it can be applied to other binary classification algorithms as well. The calibrated probabilities are retrieved by passing the output $f(x)$ of any binary learning method through a sigmoid function

$$P(y=1|f) = \frac{1}{1+\exp(Af+B)}, \tag{4-39}$$

with A and B parameters that are fitted using maximum likelihood estimation from some fitting training set $\{f_i, y_i\}_1^N$ [101]. Accordingly, A and B are found by gradient descent that solves the following equation

$$\arg\min_{A,B} \left\{ -\sum_i y_i \log(p_i) + (1-y_i)\log(1-p_i) \right\}, \tag{4-40}$$

where

$$p_i = \frac{1}{1+\exp(Af_i+B)}. \tag{4-41}$$

See Platt [104] for more detail and justification of the algorithm, including the out-of-sample method.

4-5-2 Isotonic Regression

Isotonic Regression, proposed by Zadrozny and Elkan [144], is a more general approach for probability calibration that lies somewhere between binning and sigmoid-fitting. The method is a non-parametric form of regression and is shown to be superior when training data is large enough to avoid overfitting [101].

Assuming the output of the classifier to be ranked correctly, then the mapping m from scores into probabilities is isotonic (non-decreasing). Therefore, isotonic regression can be applied to learn the mapping, using

$$y_i = m(f_i) + \epsilon_i. \tag{4-42}$$

Then, given a training set $\{f_i, y_i\}_1^N$, isotonic regression finds its solution by solving for the isotonic function

$$\hat{m} = \arg\min_z \sum (y_i - z(f_i))^2. \tag{4-43}$$

Niculescu-Mizil and Caruana [101] suggest the use of the pair-adjacent violators (PAV) algorithm proposed by Ayer et al. [8] to solve the isotonic function $\hat{m}$. The PAV algorithm finds a stepwise-constant solution to the problem that best fits the data according to a mean-squared error criterion [144]. Let $\{x_i\}_{1=1}^N$ be the training examples, $g(x_i)$ be the value of the function to be learned for each training example, and g^* be the isotonic regression. Then the algorithm replaces all pair-adjacent violaters $g(x_{i-1}) \leq g(x_i)$ by their average, such that x_{i-1} and x_i comply with the isotonic assumption, until a new isotonic set of values is obtained [144].

4-6 Model Performance Evaluation

The main interest of this research lies in obtaining individual customer purchasing probabilities with the final goal to improve short-term forecasting accuracy of an online grocer. Therefore, a model is desired that has high discriminative power, meaning that it is able to separate customers that will place an order for a specific day from those who will not. There are many different approaches to asses and compare the performance of learning algorithms. The proper selection of evaluation methods depends on the objective and context of the problem.

4-6-1 Separability and Classification

In many practical applications involving classification tasks, the accuracy score is considered as evaluation metric for both modeling and testing purposes. The accuracy score is equal to the fraction of correctly classified observations. The estimated probabilities of the learning algorithm can be translated to predict the two classes. The default choice is a decision threshold of 50%, such that $\hat{y} = 1$ when $\hat{p} > 0.5$ and $\hat{y} = 0$ otherwise. However, since the data set that is used for analysis suffers from a severe imbalance in the class distribution (more negatives than positives), this may not be the most favorable approach in this case. In predictive modeling, imbalanced classification problems are challenging as most machine learning algorithms used for classification were designed based on the assumption of equal class distributions [126]. Heavily imbalanced sets can lead to implications such as poor predictive performance, especially for the minority class. In some cases the accuracy measure indicates excellent performance, although the accuracy is only reflecting the underlying class distribution. This phenomena is known as the *accuracy paradox* [135].

One way to deal with this issue is by re-sampling the data to obtain a more balanced set. This can be done either by deleting instances from the over-represented class (under-sampling) or adding copies of instances from the under-represented class (over-sampling). Sun et al. [126] argue that the overall consensus is that a relatively balanced distribution usually leads to better predictions. However, they follow up that it is not clear where the boundaries lie since factors such as sample size and separability are also affecting performance. According to Japkowicz and Stephen [74], the imbalanced class distribution may not necessarily burden the performance if the data set is large enough and assuming computation time is still acceptable.

Nonetheless, two implications arise when re-sampling training data. Since the classification problem involves a time series multi-label target space, re-sampling for one target column would consequently affect all the other labels as well. This makes simultaneous balancing of all labels (almost) impossible. However, this is necessary for applying the algorithms discussed in 4-2 and being able to capture possible correlations between the labels. Even more important, re-sampling modifies the priors of the training set which subsequently biases the posterior probabilities of the models [46], which is not desired within the application at hand.

Therefore, it is suggested to use a different scoring metric rather than modifying the data. The Receiver Operating Characteristic (ROC) curve parametrically plots the true positive rate (TPR) against the false positive rate (FPR) at various decision thresholds. By computing the area under the ROC curve (AUC), these metric measures the separation performance of the model rather than the accuracy. The true and false positive rates are given by

$$TPR = \frac{\sum \text{True positive}}{\sum \text{Condition positive}} = \frac{TP}{P}, \tag{4-44}$$

$$FPR = \frac{\sum \text{False positive}}{\sum \text{Condition negative}} = \frac{FP}{N}. \tag{4-45}$$

The class prediction for each instance is made based on the estimated probability $\hat{p}$ by the model. Given a decision threshold parameter $0 \leq Tr \leq 1$, the instance is classified as *positive* if $\hat{p} > Tr$, and *negative* otherwise. Probability $\hat{p}$ follows a density $f_1(x)$ if the instance actually belongs to class *positive*, and $f_0(x)$ if otherwise. Therefore, the true positive rate at threshold Tr is given by

$$TPR(Tr) = \int_{Tr}^{\infty} f_1(x)dx \tag{4-46}$$

and the false positive rate is given by

$$FPR(Tr) = \int_{Tr}^{\infty} f_0(x)dx. \tag{4-47}$$

By varying the decision threshold between 0 and 1, the AUC measures the models' performance across all possible decision thresholds. If the AUC takes on a value close to 1, the model is able to almost perfectly classify the samples given a certain threshold, whereas a value close to 0.5 indicates that the model makes predictions more or less at random. According to Hanley and McNeill [67], the area under the ROC curve can be interpreted as measuring the probability of ranking a random pair of observations correctly. Since the measure is scale invariant, it does not provide any information about the probability distribution but rather about the consistency of ordering.

Precision and recall are two other classification metrics that are closely related to the ROC. They evaluate the class prediction performance based on measuring relevance. Whereas recall expresses the ability to find all relevant instances in a data set, precision expresses the proportion of observations the model classified as relevant actually were relevant. Recall is known as the sensitivity and is given in Equation 4-44. Precision, also known as positive predictive value (PPV) is defined as

$$PPV = \frac{\sum \text{True positive}}{\sum \text{True positive} + \sum \text{False positive}} = \frac{TP}{TP + FP} \tag{4-48}$$

In general, there is a trade-off between the two. By varying the decision threshold, one of them increases while the other decreases and vice versa. For the main purpose of the model, neither high recall nor high precision are demanded, but the aggregated probabilities should conform with the actual purchasing behaviour of a group of customers. However, although the actual class predictions are less relevant, yet they can provide additional insights in the behaviour of the classifier as well as serve as an input for various business decisions. For example, identifying customers that are not likely to make a purchase could be useful as it would make them great candidates for some marketing campaign to increase sales. Using the precision score, one could select only the group of customers that can be identified with a probability above some specified threshold in order to optimise budgeting. Similarly, the group of customers which are very likely to make a purchase could be excluded from any promotional activities to save costs.

4-6-2 Calibration Performance

One way to assess the quality of probability estimates is by creating reliability plots [101]. This is done by first discretizing the probability space into several equal sized bins, computing the mean value of all predicted probabilities in each bin, and then plot them against the actual proportion of positive observations in each corresponding bin. A classifier is said to be well-calibrated, if the plotted line lies close to the diagonal. The Expected Calibration Error (ECE) empirically measures the calibration relative to the diagonal by

$$ECE = \sum_{i=1}^{K} P(i)|o_i - e_i|, \tag{4-49}$$

where $P(i)$ is the fraction of all observations that fall into bin i, o_i the actual fraction of positive instances in bin i, and e_i the mean of the predicted probabilities in bin i. A lower ECE corresponds to a better calibration.

Another metric for evaluating class membership probability estimates, that is frequently used in past literature, is the Brier score [144, 80]. The score, proposed by G.W. Brier [22], is a proper score function for mutually exclusive discrete outcomes. It measures the accuracy of probabilistic predictions by computing the mean squared difference of the predicted probabilities $\hat{p}$ and the actual outcomes y. There are some decompositions of the Brier score that provide additional insights in the behaviour of the classifier. For example, the two-component decomposition that generates a calibration term and a refinement term is given by [16]

$$BS = \frac{1}{N}\sum_{k=1}^{K} n_k(\hat{p}_k - \overline{y}_k)^2 + \frac{1}{N}\sum_{k=1}^{K} n_k\overline{y}_k(1-\overline{y}_k). \tag{4-50}$$

Here N is the number of total observations, K the number of unique observations, n_k the number of observations occupying the same probability value $\hat{p}_k$, and $\overline{y}_k$ the observed frequency for the event to occur. "The refinement component measures the extent to which each group of indicators assessed with the same probability is uniform in exhibiting occurrence or no occurrence among its members" conforming to Blattenberger and Lad [16, page 26]. Hence, the refinement score for a group of observations k with the same probability prediction decreases when the frequency of occurrence $\overline{y}_k$ is close to zero or one. Again, a lower value for the Brier score corresponds to better calibration, and refinement.

4-6-3 Forecasting Accuracy

Although assessing separability and calibration are important elements in the context of evaluating probability estimation methods, the models should ultimately be evaluated in the domain that the model is intended for. Since the main goal is to improve short-term demand forecasting, the output of the models is first aggregated to obtain the predicted number of daily purchases and then compared to the actual number of daily observations. One key benefit of obtaining individual purchasing probabilities is that they can be aggregated to any desired level (e.g., delivery area, total, household type, etc.). For each horizon, the aggregated forecast for a group of G customers at day t is given by

$$F_t = \sum_{i=1}^{G} \hat{p}_{t,i}. \tag{4-51}$$

Similarly, the actual number of purchases of that group is given by

$$A_t = \sum_{i=1}^{G} y_{t,i}. \tag{4-52}$$

Currently, forecasting is done for every delivery area separately as it is the lowest granularity of interest. To compare the forecasting performance of the classification models to the performance of a top-line model, relatively small groups of customers are considered that are similar to the size of common delivery areas. In general, larger groups of customers are easier to predict as behaviour of individual customers averages out. Therefore, predictions are generated for three groups of different sizes in order to investigate the effect of sample size on the performance.

In literature, several error metrics have been proposed to evaluate the performance of forecasting methods. Generally there are four types of forecast-error metrics, namely scale-dependent metrics, percentage-error metrics, relative-error metrics and scale-free error metrics (e.g., mean absolute scaled error (MASE) [71]).

For assessing accuracy on a single series, usually scale-dependent metrics are a good choice as they are easiest to understand and compute [71]. The mean-absolute deviation (MAD), defined as

$$MAD = \frac{1}{n} \sum_{t=1}^{n} |A_t - F_t|, \tag{4-53}$$

measures the size of the error in units and is probably the most popular among them. Using the same equation without taking the absolute value of the deviance obtains the mean-signed deviation (MSD) that can be used to measure bias. Note that the sign of the outcome depends on whether the A_t is subtracted from F_t or the other way around. It is chosen to use the former, in order to obtain a minus sign in case a model tends to underestimate.

However, since they are scale-dependent, these measures are less appropriate to compare models across different series. In that case, the use of one of the other metrics is preferred. According to Armstrong [7], the root-mean-squared deviation (RMSD) metric has been superseded by the mean-absolute-percentage error (MAPE) as being the most frequently used error metric to evaluate forecasting performance. The MAPE is a percentage based metric that it is easy interpretable and can be used to evaluate across series. It is defined as

$$MAPE = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|, \tag{4-54}$$

with n the number of fitted instances (days). However, in cases the series is strictly positive, the MAPE tends to favor models that underestimate the actual values as positive errors are bound by 100 percent, whereas this value can be exceeded for negative errors [71]. Next to that, it has the disadvantage of being undefined or infinite for actual values that are zero. Although, the latter is not likely to occur in the context of this research. Usually, this is only the case when the store is not open for purchases at all, like for example during Christmas. Hence, these particular days are not included in the evaluation anyways.

A more recent measure, the logarithm of the accuracy ratio, has been proposed by Tofallis [131]. As it measures the relative accuracy, the range of possible values are equal for both positive and negative values. Therefore, this method embodies valuable symmetry and avoids the bias of the MAPE. By computing the mean-squared logarithm of the accuracy ratio (MSLAR) this measure is appropriate for evaluating the relative performance of competing methods across series [131].

$$MSLAR = \frac{1}{n} \sum_{i=1}^{n} \left(\log \frac{F_t}{A_t} \right)^2, \tag{4-55}$$

The study of Tofallis [131] shows the superiority of the MSLAR over the MAPE in terms of performance in various cases where the data is strictly positive. Additionally, Tofallis proves that the resulting predictions comply with the geometric mean when the measure is used in constructing forecasting models.

4-6-4 Training, Validation and Testing

Since it is a time series problem, particular care must be taken when splitting the data in order to prevent data leakage. All data about events that occur chronologically after time of fitting the model (cutoff) need to be withhold from the training data in order to simulate a real world forecasting environment. Furthermore, to avoid overfitting and account for time dependencies, the models are evaluated using a cross-validation (CV) procedure where the training data is partitioned into k sample folds. Subsequently, each fold is split into a training and testing set. By averaging the error on each partition a robust estimate of the model error can be computed. This is equal to the outer loop of a method called nested CV. For the inner loop, the training set is again partitioned into a training and validation set. The inner loop is used for both feature selection and hyperparamter tuning purposes. Similarly, all results are averaged over all folds for robustness. The advantage of the nested CV method is that it provides an almost unbiased estimate of the true error [138]. A schematic overview of the method is given in Figure 4-5.

All modeling procedures are evaluated using a 21-fold cross-validation (CV). The 21 cutoff dates are picked uniformly over a whole year, where every weekday is chosen three times as

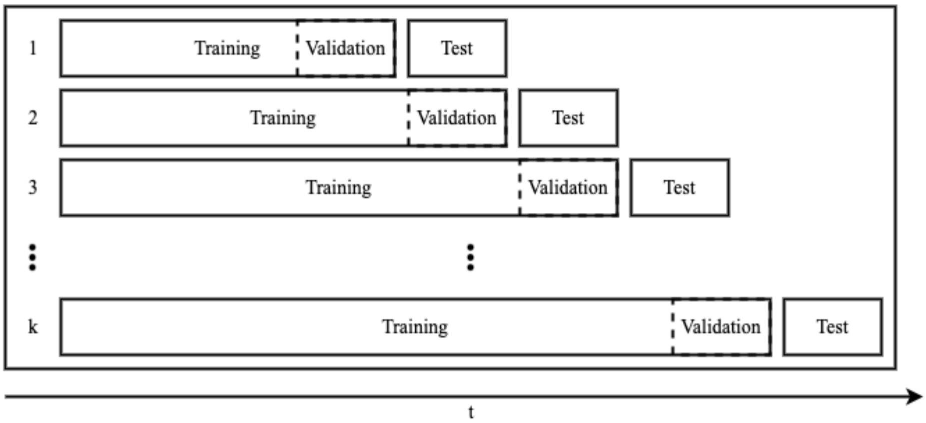


Figure 4-5: k-fold nested cross-validation for time series.

the cutoff date in order to reduce bias towards particular weekdays. The models are fitted to the training set and for each configuration validated on the validation set. In the context of feature selection, the variable importance is obtained by computing the average importance for each variable over all folds. Similarly, each hyperparameter configuration is evaluated by minimizing the average error over all folds. In the context of the classification models, both the AUC and the Brier score are valid candidates for being used as evaluation metric. Both metrics are scale-invariant which is a desirable property when comparing multiple series (e.g., 21 folds) in order to ensure equally weighting of the results. However, since the main interest of this research lies in obtaining accurate predictions by aggregating the estimated probabilities of the models, the Brier score is chosen as evaluation metric to be minimized. The Brier score measures directly the calibration of the probabilities, whereas the AUC just measures the correct ranking of the predictions. For the SARIMAX model, the best set of hyperparameters is obtained by minimizing the MSLAR. This metric is chosen due to its properties of being scale invariant and complying with the geometric mean as discussed in subsection 4-6-3. In all cases the mean loss over all horizons are considered. Hence, all labels are treated equally during the optimisation procedure.

After having obtained the model configurations that yield the best results, the final step is to evaluate and compare the performance of all models. In order to properly asses the performance of the finalized models for in the long run, the models are evaluated on a full year of data. Hence, a 365-fold day forward-chaining CV is used, which is also known as *rolling* forecast. After each prediction the cutoff date shifts by one day, then the model is re-trained on all historical data and new predictions are produced for 7 days ahead. Finally, the error measures discussed in the previous sections are computed for each horizon using the average values over all folds.

Chapter 5

Results

In accordance to section 4-6, the model performance is evaluated in terms of separability, probability calibration and forecasting accuracy. Section 5-1 first discusses the intermediate results that are obtained during the modeling procedure, including some insights on the importance of the explanatory variables. Followed by section 5-2 with a summary of the performance prior and post calibration to discuss the quality of the estimated probabilities. In section 5-3 the performance of the finalized propensity models is evaluated and their aggregated output compared with the top-line SARIMAX model.

The final model configurations obtained by the TPE bayesian optimisation discussed in section 4-4 along with the Brier scores (BS) for different numbers of trees in the random forest model for each horizon can be found in Appendix B. For the number of trees parameter B of the random forest, a value of 90 is chosen in order to ensure the best results while remaining computational efficiency.

5-1 Feature Importance

After assessing multicollinearity (see section 4-3), some of the initial explanatory variables are omitted from the set of features. For the resulting set of 63 variables it is attempted to gain insights on their value in predicting future shopping behaviour using the permutation importance (see Equation 4-34). The permutation importance measures the mean increase in the BS after the values of a single feature are randomly shuffled. Each variable is shuffled 10 times to increase statistical robustness of the prediction results. Note, that the smaller the BS the better the predictions. As mentioned in subsection 4-6-4, the measure is computed over 21 folds. The twenty variables that contribute the most to the overall prediction performance in terms of the BS are listed in Table 5-1.

As expected, the variables that embody recency and frequency are very strong predictors and among the top listed in all cases. Especially, the *number of days since the previous purchase* as well as the *number of purchases per weekday in the last 8 weeks* seem to be the most important predictors. Additionally, the *median of the observed interpurchase times* as well as the *sine* *cosine encoded periodicity of customers* have proven to be important as well. Although the fraction of customers that order more than one day in advance is relatively small, the *number of days till the known date of delivery* still has significant impact on the overall prediction performance, at least for the tree based models. Similarly, the *slot reservation* feature seems to contain a lot of predictive power for these two models. Furthermore, the *number of days since previous activity* in the application can be found high in the ranking as well, whereas other event data still can add some value to that. Other variables such as the *number of purchases one or two weeks ago*, the *total amount spent*, the *purchase rank* as well as well as the *standard deviation of the interpurchase times* all show positive effect on the prediction performance. Remarkably, the features that embody the *number of purchases per weekday in the last 8 weeks* seem to be almost the only features that matter in the context of the logistic regression CC model. Finally, it can be noticed that mainly behavioural attributes and event data contribute the most to the final prediction, whereas customer demographics and satisfaction, promotional activities and meta data seem to be less relevant in this context.

Table 5-1: Permutation importance of the 20 most important variables.

	Logistic Regression CC		Random Forest		Stochastic Gradient Boosting CC	
Rank	Variable	↑ $\overline{\text{BS}}^1$	Variable	↑ $\overline{\text{BS}}^1$	Variable	↑ $\overline{\text{BS}}^1$
1	NrWeekdayPurchasesH7^2	0.00169	NrDaysSPD^3	0.00164	NrDaysTillKnownDelDate^5	0.00198
2	NrWeekdayPurchasesH4^2	0.00169	NrDaysTillKnownDelDate^5	0.00133	NrWeekdayPurchasesH4^2	0.00143
3	NrWeekdayPurchasesH5^2	0.00168	NrDaysTillReservedSlot	0.00076	NrWeekdayPurchasesH7^2	0.00140
4	NrWeekdayPurchasesH6^2	0.00167	NrWeekdayPurchasesH4^2	0.00070	NrWeekdayPurchasesH3^2	0.00134
5	NrWeekdayPurchasesH1^2	0.00165	NrWeekdayPurchasesH5^2	0.00069	NrWeekdayPurchasesH6^2	0.00131
6	NrWeekdayPurchasesH2^2	0.00165	NrWeekdayPurchasesH3^2	0.00068	NrWeekdayPurchasesH5^2	0.00128
7	NrWeekdayPurchasesH3^2	0.00162	NrWeekdayPurchasesH6^2	0.00065	NrWeekdayPurchasesH2^2	0.00127
8	NrDaysSPD^3	0.00025	NrWeekdayPurchasesH2^2	0.00062	NrDaysSPD^3	0.00119
9	NrPurchases1WeekAgo	0.00025	SinCustPeriodicity^4	0.00059	NrWeekdayPurchasesH1^2	0.00077
10	NrDaysSinceLatestActivity	0.00025	TotalSpent	0.00055	MedianInterpurchaseTime	0.00052
11	SinCustPeriodicity^4	0.00017	NrWeekdayPurchasesH7^2	0.00053	NrDaysTillReservedSlot	0.00043
12	NrProductsAddedPriorDay	0.00017	MedianInterpurchaseTime	0.00053	CosCustPeriodicity^4	0.00020
13	NrUniqueSessionsPriorDay	0.00016	CosCustPeriodicity^4	0.00051	SinCustPeriodicity^4	0.00019
14	CosCustPeriodicity^4	0.00015	NrWeekdayPurchasesH1^2	0.00045	NrDaysSinceLatestActivity	0.00018
15	NrPurchases2WeeksAgo	0.00011	NrDaysSinceLatestActivity	0.00041	NrProductsInBasket	0.00009
16	NrDaysTillKnownDelDate^5	0.00009	NrPurchases1WeekAgo	0.00038	TotalSpent	0.00008
17	MedianInterpurchaseTime	0.00009	NrPurchases2WeeksAgo	0.00020	StDvInterPurchaseTime	0.00005
18	NrUniqueSessionsSPD^3	0.00006	NrProductsInBasket	0.00016	NrProductsAddedPriorDay	0.00005
19	AvgPurchaseCreationTime	0.00006	StDvInterPurchaseTime	0.00009	PurchaseRank	0.00005
20	PurchaseRank	0.00003	NrPurchases3WeeksAgo	0.00008	SinWeekPeriodicity	0.00004

^{1}Mean increase in BS after permutation, ^{2}H: Horizon, ^{3}SPD: SincePreviousPurchase, ^{4}Cust: Customer, ^{5}Del: Delivery

To highlight some of the features that are 'new', interesting or have proven to be important in the context of this project, a summary of their relative performance is listed in Table 5-2. The relative performance is calculated based on the permutation importance using the corresponding metric for evaluation. The importance is evaluated on individual customer level by comparing the AUC and BS as well as on the aggregated day level by comparing the MAD, MAPE and MSLAR. Since the AUC is the only metric where a higher score corresponds to better performance, its negative equivalent is given for visualization purposes. Some features are grouped together in case there exists a logical connection. If the scores are not listed, this means that the feature(s) were not adding any value to the prediction at all.

The *average of the interpurchase times* is included in order to compare its predictive value

Table 5-2: Summary of feature impact on performance prior and post variable permutation.

Metric Features	-AUC	BS	MAD	MAPE	MSLAR
Logistic Regression CC					
Number of purchases per weekday	-11.25%	-28.21%	-37.45%	-27.84%	-53.86%
Number of days since previous purchase	-1.18%	-0.61%	-5.81%	-0.25%	-7.40%
Sin Cos customer periodicity	-0.63%	-0.78%	-2.39%	-0.17%	-2.67%
Median interpurchase time	-0.46%	-0.22%	-0.88%	-0.75%	-1.23%
Average interpurchase time[1]	-0.13%	-0.03%	-0.08%	-0.14%	-0.11%
Slot closings	-0.02%	-0.01%	-0.73%	-0.76%	-0.64%
Average purchase create time	-0.13%	-0.13%	-1.66%	-0.86%	-2.11%
Incidents	-0.01%	-0.02%	-0.02%	-0.09%	-0.09%
Last order promotions	-	-	-	-	-
Event data	-1.66%	-1.62%	-7.26%	-6.77%	-9.31%
Slot reservation	-0.09%	-0.04%	-0.05%	-0.01%	-0.02%
Random Forest					
Number of purchases per weekday	-4.13%	-9.90%	-28.18%	-39.25%	-42.24%
Number of days since previous purchase	-2.02%	-4.17%	-14.57%	-18.58%	-20.69%
Sin Cos customer periodicity	-1.16%	-2.79%	-10.73%	-10.99%	-14.26%
Median interpurchase time	-0.60%	-1.36%	-3.81%	-2.57%	-3.30%
Average interpurchase time[1]	-0.09%	-0.16%	-0.76%	-0.56%	-0.98%
Slot closings	-0.02%	-0.04%	-0.79%	-0.64%	-1.05%
Average purchase create time	-0.06%	-0.06%	-0.55%	-2.06%	-2.13%
Incidents	-0.04%	-0.06%	-0.43%	-0.82%	-0.96%
Last order promotions	-0.01%	-0.01%	-0.01%	-0.12%	-0.16%
Event data	-1.13%	-2.22%	-6.02%	-8.91%	-10.95%
Slot reservation	-0.40%	-1.95%	-0.88%	-0.84%	-1.21%
Stochastic Gradient Boosting CC					
Number of purchases per weekday	-9.35%	-18.26%	-44.63%	-35.08%	-40.29%
Number of days since previous purchase	-1.88%	-2.94%	-16.24%	-10.47%	-12.07%
Sin Cos customer periodicity	-0.26%	-1.02%	-5.70%	-2.15%	-2.72%
Median interpurchase time	-0.86%	-1.33%	-11.15%	-5.34%	-9.02%
Average interpurchase time[1]	-0.05%	-0.13%	-1.49%	-0.86%	-1.05%
Slot closings	-0.23%	-0.16%	-1.68%	-1.51%	-1.75%
Average purchase create time	-	-	-	-	-
Incidents	-0.01%	-0.02%	-0.08%	-0.82%	-0.38%
Last order promotions	-0.00%	-0.01%	-0.03%	-0.04%	-0.06%
Event data	-0.97%	-0.89%	-6.02%	-3.29%	-5.91%
Slot reservation	-0.28%	-1.08%	-0.92%	-0.82%	-2.67%

[1] Used instead of the median for comparison purposes

to that of the median. As expected, the *median of the interpurchase times* outperformed the average value by many times over. Interestingly, it seems that the *sine and cosine customer periodicity* have more positive impact than the median of the interpurchase time in both the logistic regression CC model and the random forest model, whereas for the gradient boosting CC model the opposite is the case. However, for logistic regression CC the difference between the two values is rather small. Remarkably, the logistic regression CC model is not able to translate the *slot reservation* feature very well in order to enhance predictions, whereas the other two models tend to do better. As seen earlier, features that contain event data have considerable impact on the prediction as well. The contribution of information about recent promotions seem to carry little to no predictive power, while information about incidents can improve predictions slightly. Last but not least, it is interesting to see that the *average purchase create time* has quite some positive effect in both the logistic regression CC and the random forest model whereas none in the gradient boosted tree CC model. However, the features concerning *slot closings* contribute to the prediction in all three models.

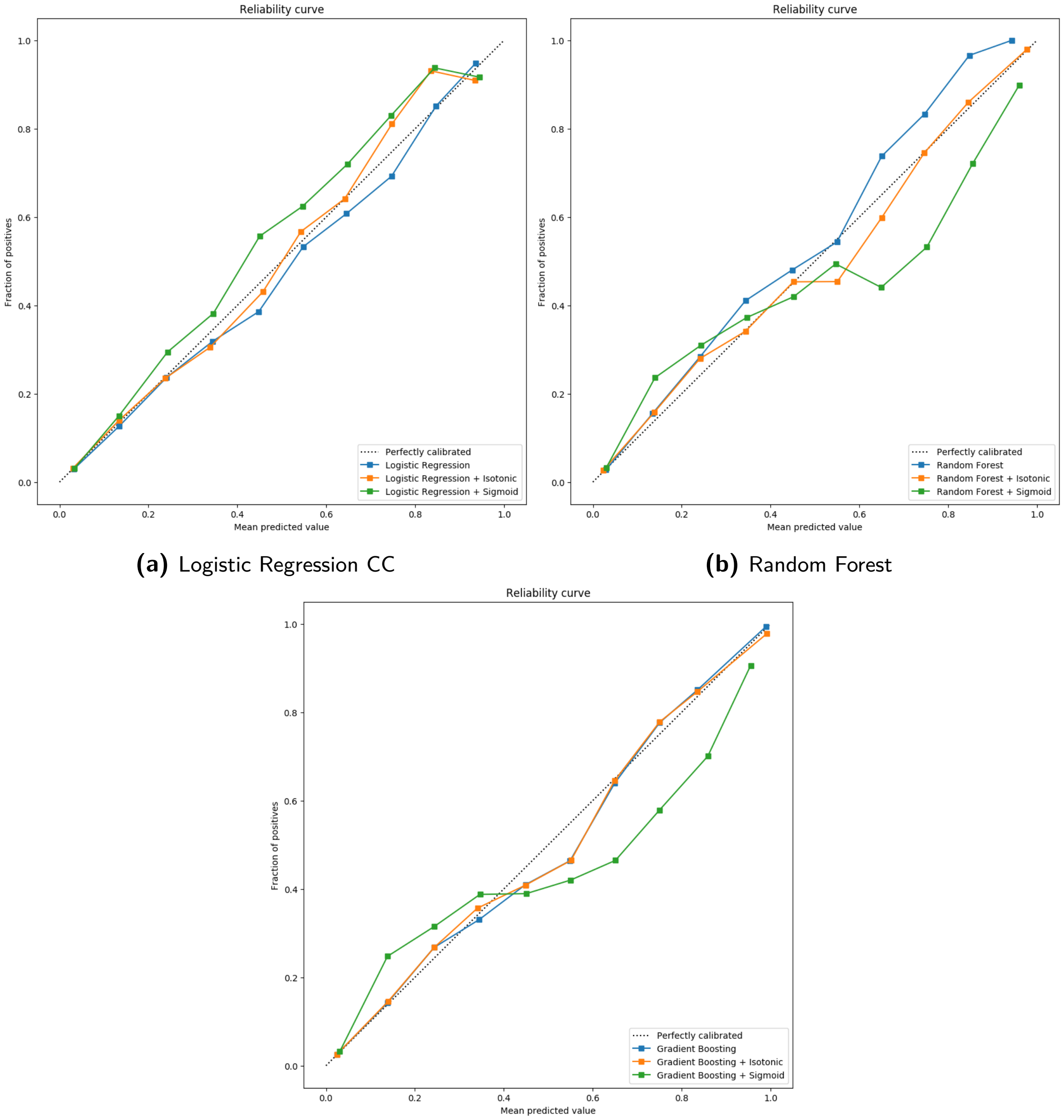
After assessment of the individual variable importance, all features that do not (or barely not) contribute to the prediction are omitted from the set of features. For the logistic regression CC model that leaves 23 features, for the random forest 31 features and for the stochastic gradient boosting CC model 26 features that are used as an input for the final models. A performance comparison prior and post the selection process, in order to ensure no valuable predictors were excluded, is listed in Table 5-3. Again, the negative AUC is used for better visualization. The absolute values of the MSD are compared in order to see whether the bias of the predictions increases or decreases. It can be noticed that in all cases the in-sample performance decreases while the predictions get more accurate on the test sets.

Table 5-3: Performance comparison prior and post feature selection.

	Logistic Regression CC		Random Forest		Gradient Boosting CC	
Sample / Metric	Training	Test	Training	Test	Training	Test
-AUC	+0.09%	-0.01%	+0.06%	-0.15%	+1.06%	-0.10%
BS	+0.29%	-0.25%	+0.44%	-0.47%	+0.18%	-0.13%
MAD	+6.50%	-6.03%	+1.66%	-0.93%	+69.80%	-5.95%
|MSD|	+19.31%	-3.11%	+5.34%	-73.50%	+281.65%	-76.13%
MAPE	+5.63%	-3.65%	+1.51%	-1.16%	+71.59%	-0.99%
MSLAR	+8.78%	-3.41%	+2.64%	-0.56%	+181.10%	-50.10%

5-2 Calibration

In order to asses the quality of the probability estimates, for each classification model reliability curves are created which are visualized in Figure 5-1. The probability estimates of all horizons are considered simultaneously to create the plots. To investigate whether calibra-



(a) Logistic Regression CC

(b) Random Forest

(c) Stochastic Gradient Boosting CC

Figure 5-1: Reliability curves.

tion improves the performance, the reliability plots after applying Platt Scaling and Isotonic regression are included as well (section 4-5). In general, it can be noticed that for all three models the plotted reliability curves prior to the application of any calibration method are close to the diagonal, implying that they already provide reasonable well-calibrated probabilities. For the logistic regression CC model, it can be inferred that the probabilities in the lower and higher bins are very close to the diagonal, whereas some deviations occur in the midsections where the classifier tends to slightly overestimate. The reliability curve of the random forest model shows deviations almost across all probabilities with a bias towards underestimation. In comparison to the random forest, the boosted CC model seems to be better calibrated except for the probabilities that are part of the sixth bin. Interestingly, the boosted CC model seems to underestimate slightly in the lower and upper midsections, while overestimating probabilities that lie around 0.5 and 0.6.

As no sigmoidal shape is noticeable in the reliability plots of any model prior to calibration, Platt scaling is not expected to lead to much improvement. The plots confirm this. Actually, it even seems to worsen the situation in all of them. In the case of logistic regression CC it leads to underestimation across all bins except the first and last one. In the first bin there is almost no visible change whereas for the last one the model slightly overestimates. In both the random forest and the boosted tree CC model the curves take on a sigmoidal shape after applying Platt's method, where the models tend to underestimate in the lower range of the probabilities and overestimate even more in the higher range. In the case of isotonic regression, not much improvement can be recognized either. It seems that it mitigates the slight bias of the logistic regression CC in the midsections, however leads to underestimation in the higher bins but the last one. Similarly, the method mitigates the bias of the logistic regression CC which however leads to overestimation in two bins of the midsection. In the case of the boosting CC model, almost no change is visible compared to the original curve.

In Table 5-4 the performance prior and post the application of calibration methods is given in the context of different metrics. According to the expected calibration error (ECE), the methods do not improve the quality of the probability estimates at all. Only exception is the logistic regression CC model, where the output is scaled using isotonic regression. However, although the ECE is decreased by almost a half, investigating the other metrics shows that it has neither positive effect on the discriminative power (AUC), calibration and refinement (BS) nor on the aggregated forecasting performance. Just in the random forest case, using isotonic regression can slightly improve performance on individual customer level, however still hurts aggregated forecasting performance. In all other situations either the improvement is negligible or the performance decreases. The negative effect is more significant after applying Platt scaling than applying isotonic regression. Since both calibration methods do not provide any improvement in forecasting performance, neither is used to obtain the final results described in section 5-3.

Table 5-4: Performance comparison prior and post calibration.

Model \ Metric	ECE	-AUC	BS	MAD	|MSD|	MAPE	MSLAR
Logistic Regression CC							
+ Isotonic regression	-46.34%	+0.14%	+0.00%	+5.32%	+25.98%	+4.46%	+12.53%
+ Platt scaling	+34.14%	+0.12%	+0.24%	+7.33%	+46.56%	+9.41%	+17.97%
Random Forest							
+ Isotonic regression	+8.00%	-0.23%	-1.01%	+1.99%	-13.88%	+1.18%	+2.62%
+ Platt scaling	+42.00%	+0.56%	+1.26%	+9.97%	+17.55%	+11.63%	+34.50%
Gradient Boosting CC							
+ Isotonic regression	-0.00%	-0.02%	+0.00%	+8.81%	+0.00%	+8.02%	+5.24%
+ Platt scaling	+140.74%	+1.15%	+3.08%	+28.30%	+186.66%	+28.61%	+51.31%

5-3 Final Model Performance Comparison

After the feature selection procedure, configuration of the models and assessment of the calibration performance, the models are finally evaluated on a full year of data in order to investigate their true performance. Since features that concern holidays and yearly seasonality are acknowledged as irrelevant in all three models, only historical data between cutoff and four month before cutoff is used as an input for training. This substantially reduces training computation time while achieving similar performance. As discussed in section 5-2, neither Platt Scaling nor Isonotonic Regression are applied to obtain the final results. The performance of the classification models is compared with the performance of the top-line SARIMAX model described in subsection 4-1-1. Predictions are made for three customer groups of different sizes to simulate different delivery areas. For each group size, 10 groups of customers are randomly sampled from the data base in order to obtain a representative sub sample of the customer base and increase statistical reliability of the results. The average results of the 10 samples for 3250 customers are summarized in Table 5-5. Similarly, the results for 6500 and 13000 customers are listed in Appendix C-1 in Table C-2 and Table C-3, respectively. A comparison of the prediction errors using histogram plots is given in in Appendix C-2.

In general, all tables carry the same message. The stochastic gradient boosting CC model achieves the highest performance across the board. Next to that, it is the only method, out of the classification techniques that are considered in this project, that beats the top-line benchmark SARIMAX model in terms of forecasting performance. The results of the random forest model, however, come fairly close and show a little less bias than the predictions of the SARIMAX model. Logistic regression CC, the benchmark model in the context of the classification techniques, performs the worst. Remarkably, the gap between the logistic regression CC model and the rest is quite large. For example in Table 5-5, it scores about 40% worse in MAD and about 80% worse in MSLAR compared to the random forest model. This gap slightly decreases with growing number of customers, however remains significant. The standard deviation of the MSLAR shows reasonable deviations across the results of all samples, which suggests the statistical consistency of the results.

Generally all models perform better when predicting a larger group of customers. This can be inferred by investigating both the MAPE and the MSLAR, which decrease with a greater number of customers. On individual customer level there is not much noticeable deviation, just a marginal increase in the Brier score loss. Moreover, the standard deviation of the MSLAR slightly decreases, suggesting that the results get more stable when predicting larger customer groups. Furthermore, from the MSD values it can be seen that all models tend to underestimate the target values in all situations. After logistic regression CC, SARIMAX exhibits the most bias, followed by the random forest model. For each model it holds that the bias gradually increases when predicting more customers. By analysing the histogram plots in Appendix C-2, it can be observed that the logistic regression CC and the random forest model actually tend to "slightly" over predict in most of the cases for larger customer groups, while exhibiting some larger under predictions. The prediction errors of the stochastic gradient boosting CC model show an evenly, around the mean distributed bell shape, indicating less bias as well as variance. Interestingly, the bias of the SARIMAX model increases even

Table 5-5: Model performance evaluation - 3250 customers.

Model	Horizon	AUC	BS	MAD	MSD	MAPE	MSLAR	StDv[1]
SARIMAX								
	1	-	-	10.25	-2.48	10.79	0.0204	4.16
	2	-	-	10.73	-2.79	11.27	0.0225	4.14
	3	-	-	10.81	-2.86	11.28	0.0233	4.29
	4	-	-	10.82	-2.86	11.31	0.0235	4.24
	5	-	-	10.85	-2.88	11.32	0.0236	4.38
	6	-	-	10.87	-2.90	11.36	0.0236	4.35
	7	-	-	10.96	-2.97	11.41	0.0237	4.42
Logistic Regression CC								
	1	0.8345	0.0489	16.18	-3.99	16.22	0.0419	5.24
	2	0.7759	0.0490	16.31	-4.01	16.35	0.0423	5.23
	3	0.7665	0.0491	16.35	-4.03	16.36	0.0424	5.12
	4	0.7660	0.0491	16.38	-4.11	16.36	0.0424	5.34
	5	0.7658	0.0491	16.40	-4.13	16.39	0.0424	5.32
	6	0.7657	0.0491	16.41	-4.13	16.39	0.0424	5.31
	7	0.7652	0.0491	16.41	-4.16	16.46	0.0427	5.42
Random Forest								
	1	0.9050	0.0299	10.96	-2.17	11.18	0.0236	3.67
	2	0.8579	0.0389	11.51	-2.20	11.81	0.0237	3.74
	3	0.8331	0.0418	11.56	-2.36	11.97	0.0237	3.62
	4	0.8220	0.0429	11.62	-2.40	12.01	0.0238	3.63
	5	0.8166	0.0430	11.64	-2.49	12.02	0.0240	3.89
	6	0.8140	0.0430	11.66	-2.70	12.02	0.0241	4.01
	7	0.8121	0.0431	11.76	-2.84	12.11	0.0261	3.99
Gradient Boosting CC								
	1	0.9081	0.0292	8.74	-0.12	9.13	0.0178	3.41
	2	0.8612	0.0382	9.47	-0.20	10.05	0.0186	3.31
	3	0.8380	0.0411	9.53	-0.41	10.16	0.0188	3.38
	4	0.8275	0.0421	9.64	-0.42	10.21	0.0189	3.30
	5	0.8231	0.0424	9.69	-0.68	10.21	0.0197	3.86
	6	0.8218	0.0425	9.69	-0.88	10.32	0.0198	3.79
	7	0.8209	0.0426	10.04	-1.13	10.66	0.0200	3.84

[1] Standard deviation ($\times 10^{-4}$) of the MSLAR

faster than for the rest and ends up taking on values that are close to the values of the logistic regression CC model. Hence, for a group of around 13000 customers the stochastic gradient boosting CC model not just performs best in terms of all metrics, but also exhibits a bias that is way lower than that of the top-line model (see Table C-3).

As expected, the prediction performance of the models decreases with increasing number of horizons. However, there are some differences between the models. For example, both the

SARIMAX and the random forest model provide one-day ahead predictions with a disproportional lower deviation compared to the horizons that lie further in the future. In other words, there is a larger gap in performance between horizon 1 and 2 than between the other labels. The same holds for the stochastic gradient boosting CC model when predicting demand of 3250 customers, while the performance gradually decreases over all labels for larger customer groups. The performance of the logistic regression CC model nearly stays the same for all labels in all situations, just a slight decrease is noticeable.

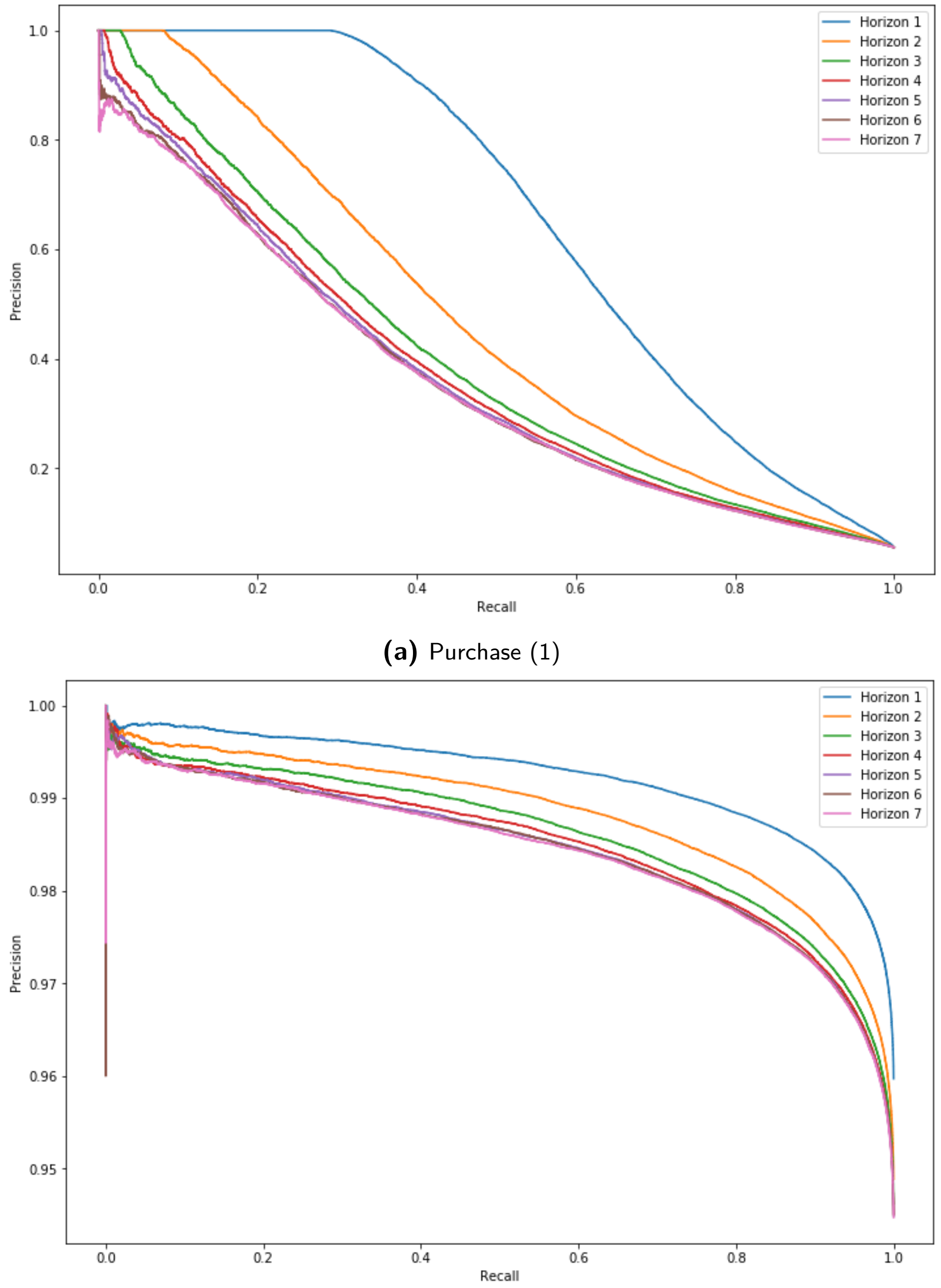
Finally, the the average computation times per iteration for all four models are given in Table 5-6. By comparing the results, it can be noticed that the computation time increases with increasing number of customers for both fitting (training) and prediction. In general, the computational complexity of the propensity models is greater than of the top-line model, which is as expected due to the size of input data. Note that, although the SARIMAX model includes the complete purchase history of the customers within a group, whereas the propensity models include only four month of historical input data, the latter still have to handle substantially more input data than the SARIMAX. Next to that, the computation time of the CC models is greater than the adapted random forest model, which coincides with the expectations as well. Interestingly, the training time of the logistic regression CC model increases rapidly for more input data (larger customer groups), such that it, at some point, takes longer to fit the model than for the stochastic gradient boosting CC. Despite some noticeable differences, the prediction times of all model configurations are reasonably quick.

Table 5-6: Model computation time comparison.

	SARIMAX	Logistic Regression CC	Random Forest	Gradient Boosting CC	Customers
Fit					
	16.05s	14.21s	27.29s	76.48s	3250
	26.99s	204.65s	73.16s	144.21s	6500
	34.44s	380.09s	163.39s	297.20s	13000
Predict					
	0.04s	<0.00s	0.11s	0.11s	3250
	0.05s	0.01s	0.21s	0.24s	6500
	0.07s	0.02s	0.41s	0.44s	13000

5-4 Classification

In order to evaluate the class prediction performance of the models, precision and recall curves are created. The curves are plotted for both the positive (purchase) as well as the negative (no purchase) class for each horizon. They visualize the precision and recall scores under varying decision thresholds. In general a larger decision threshold conforms with high precision, whereas a lower threshold conforms with high recall. As it is the best performing model, the precision versus recall plots of the gradient boosted tree CC model for 3250 customers are included in this section in Figure 5-2. The plots for the logistic regression CC and random forest model along with the plots regarding the prediction of larger customer groups can be found in Appendix C.



(a) Purchase (1)

(b) No Purchase (0)

Figure 5-2: Precision and Recall curves of the stochastic gradient boosting CC model (3250 customers).

As expected, class prediction performance decays when predicting days that lie further in the future, which is in line with the results found in section 5-3. For example, from Figure 5-2 (a) it can be inferred that for horizon 1 around 35% of the positive class can be classified correctly with 100% precision, whereas for horizon 2 and 3 this number goes down to approximately 10% and 5%, respectively. Additionally, it can be noticed that the relative performance for the negative class is significantly better than for the positive class, which is as expected due to the heavy class imbalance. As indicated in Figure 5-2 (b), approximately 95% of the negative class can be classified correctly with a precision greater than 97.5% in all the cases. By comparing the plots with Figure C-4 and Figure C-5, it can be observed that the random forest model shows slightly worse performance than the gradient boosted CC model, whereas the logistic regression CC performs quite poorly. A noticeable difference is, that the random forest for a horizon greater than 2 is not able to classify even a small portion of the positive class with a precision of 100%. Similar to the results obtained in section 5-3, the performance of the logistic regression CC model in terms of class prediction lies closely together for all horizons. By comparing the plots of the class predictions regarding larger customer groups, no major differences are noticeable. Only exception are the results of the logistic regression CC model, that show an significant increase in class prediction performance for both the positive and negative class of horizon 1 (see Figure C-7 and Figure C-10).

Chapter 6

Discussion

This chapter discusses the results, examines the limitations of this research and states directions for future research and practical actions.

6-1 Principal Findings

In this section the major findings are discussed as illustrated in chapter 5. The debated topics concern the performance results in terms of forecasting and probabilistic modeling, as well as the analysis of the explanatory variables.

6-1-1 Model Performance

The model performance can be decomposed into three subtopics; separability, calibration and forecasting, as discussed in section 4-6. The latter being the most important, as it conforms with the main goal of this research. In general, the obtained performance results are very satisfactory. The proposed approach yields good results that are of value for management and decision makers. The results suggest that similar approaches as found in online customer behaviour studies can be used for obtaining customers' daily purchase probabilities of an online grocer. Within this context, it is shown that these predictions indeed have the potential to enhance short-term aggregated forecasting accuracy.

The tree-based models outperform the logistic regression model in terms of customer behaviour prediction, which coincides with the results of previous studies concerning various applications [80, 84, 86, 103, 136]. Furthermore, the stochastic gradient boosting model achieves better results than the random forest model which supports the findings of the research conducted by L. Raasveld [107]. While the results show little difference in performance between the tree-based models in the context of *invites* conversion rate prediction, the boosted tree model was able to improve by about 1.77% in MAPE and 20.95% in MSLAR compared to the random forest in the application at hand. This improvement led to the superiority of the boosted tree CC model in terms of both classification and forecasting.

Despite the fact that the time series benchmark model uses all historical data as an input and zero-order days are imputed, the aggregated predictions of the stochastic gradient boosting CC model resulted in higher accuracy (e.g., an improvement of 1.15% in MAPE and 16.81% in MSLAR). The predictions of both the logistic regression CC model and the random forest are less accurate than the top-line model. However, the difference between the latter two is marginal and the predictions of the random forest even exhibit less bias. Therefore, the stochastic gradient boosting CC model is the only model that beats the top-line model in terms of all forecasting metrics. The model provides consistent results for different customer groups of different sizes, with prediction errors that exhibit less bias as well as variance compared to all the other models.

The results support the notion that the logistic regression model is not able to model interactions between variables very well [50], which could be among the reasons for its bad performance. For example, the low permutation importance of features that concern customer purchase periodicity, indicate that the logistic regression CC model lacks the ability on generalizing very well on such input data. Surprisingly, the model does not even pick up the input feature *number of days till a known date of delivery*, which can be inferred by analysing the corresponding precision and recall curve for the positive class, where the graph should indicate a precision of 100% for at least a fraction of samples (see Figure C-4). In contrary, the results of the tree based models demonstrate their ability to capture interaction effects between variables without needing to specify them.

Although tree-based models are quite robust to the inclusion of irrelevant features, filtering these out can still lead to slight improvements in performance. The results post the selection procedure indicate that the stochastic gradient boosting CC model needs fewer features in order to achieve better results than the random forest model. The logistic regression CC model mainly bases its predictions on a few features, which suggests that the overall generalization power of the logit model is lower than of the tree-based models. The performance gap between horizon 1 and 2 that is observed within the tree-based models, probably is the result of exploiting event data along with purchases that are already known, as the impact of these features is expected to be most significant for one-day ahead predictions. This would also explain why no analogous gap is noticeable in the results of the logistic regression CC model.

Eventually, the application of calibration methods, namely Platt scaling and Isotonic regression, did not lead to much improvement which also coincides with the results obtained by L. Raasveld [107]. The low value in Brier score of the stochastic gradient boosting CC model without the application of these methods, along with the low bias when aggregating the probabilities, suggests that it already provides reasonably well-calibrated probabilities, which is not in line with previous research [100]. Friedman, Hastie, and Tibshirani [56] argue that boosting can be viewed as an additive logistic regression model, with the result that the predictions are trying to fit a logit of the true probabilities, as opposed to the true probabilities themselves. To obtain the true probabilities, the logit transformation must be inverted first. A possible reason for the disparate outcome, could lie in the nature of the data at hand. Most classification problems do not represent time series and involve re-sampling of the training set in order to obtain an equal class distribution. As a consequence, the basic assumption in machine learning that both training and testing sets are drawn from the same underlying distribution is violated [46]. Within the application at hand the underlying distribution is maintained, which suggests that the posterior probabilities approximate the true probabilities. Subsequently, this would be a reason why the application of calibration methods does actually hurt the forecasting performance rather than improving it. By applying calibration methods, the posterior probabilities are modified such that their aggregated values happen to be less accurate than the initial values.

6-1-2 Explanatory Variables

The results demonstrate that variables regarding behavioural attributes and event data contain most predictive value when estimating customers' daily purchase probabilities based on their online shopping behaviour, which is in line with the research concerning various other applications that concern customer behaviour prediction [28, 83, 141, 136]. Within the application at hand, especially the temporal aspects of purchases (e.g., interpurchase times) are significant of the target variables. This differentiates from the above-mentioned research, where these variables either were of low importance or not considered at all. Eventually, features that embody past shopping behaviour and are related to the recency and frequency of both purchases and visits, the day of purchasing and the interpurchase times, were most significant in this context. This outcome is as expected, since it was one of the main motivations for this research. Moreover, the encoded purchase periodicity feature that was introduced in section 3-1-2, has led to compelling improvement in prediction performance.

The results indicate that the impact of customer demographic variables on prediction performance is negligible in the context of this research. This is in contrast to various research, concerning customer churn [84, 86, 141], estimating next-buy probabilities [84, 136] and consumer credit risk prediction [80], which all found demographic attributes among the top ranked variables. This suggests that the customer groups at hand either do not exhibit different behaviours, that the difference is not significant enough or that the features do not lead to the "right" segmentation. Another possibility could be that their behaviour is to most extent covered by the behavioural features, which is a reasonable explanation since these features are most related to the target variables. The past behaviour of an individual does tell much more about its future behaviour than, for example, its household composition. Next to that, the fraction of new customers without much purchasing history is quite small, which suggests that the impact of customer characteristics (to classify a new customer with the hope to enhance predictability) on overall performance will be fairly low. Similar reasoning can be used regarding the meta data features. The behavioural features (e.g., the number of past purchases per weekday) already implicitly contain information about the day of week, such that the weekday features barely add any value to the prediction.

6-2 Business Implications

The obtained short-term purchase probabilities of individual customers are valuable for an online grocer in several ways. First of all, the results suggest that the predictions can be used as an alternative to conventional time series and regression techniques in the context of short-term forecasting with the potential to improve accuracy. Improving forecasting of future demand is of great value for any kind of business, since it can substantially increases the competitive advantage by reducing waste and saving costs. The ability in early adopting the model in new delivery areas makes it even more attractive.

Besides providing the demand in total number of customers, the obtained individual probabilities can serve as valuable input to inbound forecasting practices. By combining the probabilities with the product purchasing history of individual customers, it has the potential to improve inbound forecasting accuracy as well.

Another great advantage of obtaining predictions on individual customer level, is the ability to gain additional knowledge. The estimated probabilities can be exploited for various analysis purposes to retrieve information that was not available before. For example, the possibility to aggregate to any level that is desired, provides a tool for analysts to analyse the obtained predictions, not just on delivery area level, but also on any other segmentation of customers, without the need of building separate models. Similarly, this provides the possibility to evaluate the predictions on a more detailed level. In case there happens to be a (surprisingly) large deviation between the actual purchases and the predicted purchases, the model can be used to identify the customers that did not act as expected. Subsequently, this information can give some valuable insights in the behaviour of certain customer groups. Hence, next to short-term forecasting, this model could be deployed to detect deviations in customer behaviour and may even be used as a basis for a customer churn model.

The information that comes available through the analysis of (changing) customer behaviour is helpful from a marketing perspective as well. The individual purchasing probabilities can facilitate companies to make decisions on how to execute marketing strategies in order to increase sales. The strategy to apply may depend on the type of customer and the probability of him/her making a purchase. Tracking the purchasing probabilities of individual customers allows companies to immediately take action in case a decrease in interest in the service of certain customers is recognized. Targeted marketing is acknowledged as one of the most important steps to increase online conversion rates [24].

6-3 Limitations

Although the analysis conducted in this research confirms that predictions on customer level have the potential to improve aggregated forecasting accuracy, the outcome of this research may would have been different if some other type of model was chosen as top-line model for comparison. As argued before, this does not weaken the relevance of the findings. The SARIMAX model that is chosen as a benchmark, is acknowledged as suitable method for similar time series problems. The performance is expected to be in line with other (machine learning) methods, such that the results are representative and provide a relevant benchmark.

Another limitation concerns the generalisability of the results. For example, as zero-order customers were excluded prior to the analysis, the results cannot tell whether a similar outcome would be obtained in case all customers, including customers with no purchase history, were included during model training.

Moreover, the research within this study is restricted to the data of a single online grocer, which questions the applicability at other companies in both e-grocery retail and e-commerce. In the context of e-grocery retail, the results are expected to be generalisable to other companies as well, since most predictive variables that were used should be available there as well. On the other hand, it is not expected that the implementation of a similar approach in conventional e-commerce would be quite successful. As discussed in the introduction to this research, the interpurchase times in e-commerce are of more volatile nature than in online grocery retail. However, the results indicate that variables concerning interpurchase times, in particular, are most significant of the target variables.

Since the models were evaluated on a prediction horizon of 7 days only, the results cannot tell whether the proposed approach can be used to obtain reasonable predictions for on a longer term as well.

6-4 Recommendations and Future Research

For Picnic it is recommended to deploy the model in the daily operation and compare its performance with the models that are currently in use. Before that, an adjustment to the model is required, since in the current state the model does not take zero-order customers into account. This can be realized by adding the predictions of a conversion model to the output of the probability model, for example, by using the model developed by L. Raasveld [107] or just a simple model based on average conversion rates.

Another possibility would be, to first investigate whether zero-order customers can be included directly in the proposed propensity model. This subsequently introduces the second recommendation of this research. Since information about application usage is included, the model is expected to have some capacity to predict zero-order customers as well. This approach would provide the advantage of having just a single model in use, but would result in substantially more memory usage and longer training times. However, neither of them should be a limitation for practical usage. To minimize input data, one could choose to only include the data of new customers that were recently active in the application.

Since it is beyond the scope of this research to perform feature engineering for the topline model as well, it would be interesting to extend the analysis to other (machine learning) models, that provide the possibility to include explanatory variables. Similarly, future studies could asses other propensity models for obtaining individual purchase probabilities. For example, recurrent neural networks (RNNs) and Long-Short-Term-Memory (LSTMs) models are applied to many sequence modeling tasks, such as natural language processing, as they are able to model time dependencies in the input data. These stateful methods can be used for both time series and classification problems, and are recently applied in the context of consumer behaviour prediction as well [44, 83, 99]. Since consumer behavior is inherently sequential, the above-mentioned models provide a perfect fit.

Future research should consider the validation of the obtained results using data of other companies. Further research is needed to investigate if the results are generalizable to other fields of industry as well.

Future studies should establish the extension of the propensity model to investigate the evolution of performance for larger horizons and its effect on individual feature importance.

For future research, it is recommended to focus mainly on behavioural features that best describe the weekly purchasing behaviour of customers. An extensive feature exploration, investigating different variants of behavioural variables in this context, is expected to further improve the performance of the short-term propensity model. In case the model is adapted to larger horizons, it is recommended to invest in further research concerning yearly seasonality features, as these are expected to become more relevant.

Adjusting some of the features that are used within this study, could yield better results as well. For example, the encoded variables that account for summer and regular holidays could be replaced by dedicated variables for each horizon for each specific holiday separately. Similarly, all other features that could be expressed in more detail (e.g., type of promotions, gifts, incidents, clickstream/event data, etc.) could be assessed in terms of predictive value in future research. Since both slot closings and slot reservations features contribute to the predictive power of the models, it is recommended, at least for Picnic, to further explore possible variations of these features.

Future research should experiment with different evaluation metrics during the modeling process to investigate what the effect is on performance. During this study it is chosen to optimise the propensity model configurations based on the quality of individual probability estimates, whereas it may be favourable to optimise directly on (aggregated) forecasting performance (e.g., by using the MSLAR for evaluation).

The possible correlations between labels motivated the use of a "single" model (e.g., adapted algorithms, classifier chains (CCs) etc.) that simultaneously predicts all labels. Future research is needed to verify whether this motivation is justified. One approach could be, to provide a thorough comparison of binary relevance models, where the hyperparameter configuration is optimised for each horizon separately, with adapted propensity models and propensity models that are extended using CCs. Ideally, such studies include a comparison of the same learning algorithm in all three variations and an evaluation of the features that are introduced by the CC model.

While it is beyond the scope of this research, future studies should establish the implementation of outlier detection and imputation, as it is expected to greatly enhance prediction performance [4]. For the propensity model it is not as straightforward as for an univariate time series model. Questions arise on how to identify an outlier and how to impute it in order to keep the relationship between input and output variables. The challenge of imputation in a propensity model is that multiple features need to be imputed instead of a single regression variable.

Finally, future research should establish an detailed assessment of the performance over time. This includes model prediction performance as well as measurement of the variable importance. During this research, all explanatory variables are evaluated based on the overall performance. To gain additional insights for further analysis, it would make sense to evaluate variables on their specific use cases. For example, evaluating holiday features only during holidays would prevent skewing their importance due to the prevalence of regular days.

Chapter 7

Conclusion

This research proposes a novel approach for short-term customer demand prediction within the e-grocery retail market, which can provide an alternative to conventional time series forecasting techniques. The results show that the stochastic gradient boosting CC model outperformed the other propensity models and was able to achieve a significant improvement compared to the performance of the top-line benchmark model. The model acquired consistent results for customer groups of different sizes, with prediction errors that exhibited the lowest bias as well as variance of all models. The analysis of the explanatory variables indicate that behavioural attributes and variables, that concern interpurchase times in particular, were most significant of the target variables. To validate the obtained results, future studies should extend the analysis to other time series benchmark models. Further research is needed to investigate the generalisability of the results to zero-order customers, other companies and fields of industries. Based on the promising results of this research, it is recommended to focus on further improvement of the proposed methodology, which concerns both the quality of explanatory variables and the learning algorithms.

Appendix A

Correlation Matrix

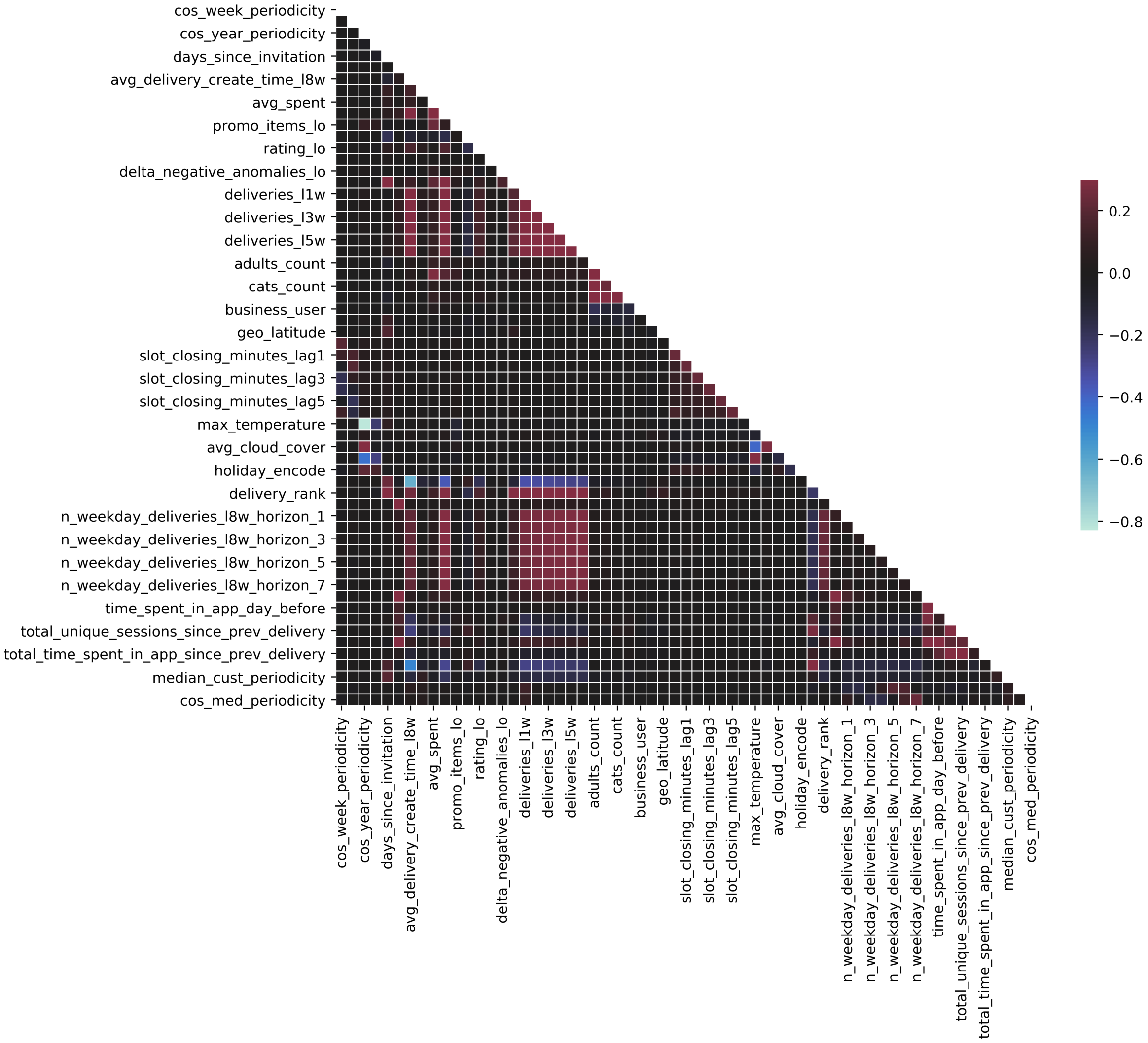


Figure A-1: Heatmap correlation matrix.

Appendix B

Model Configuration

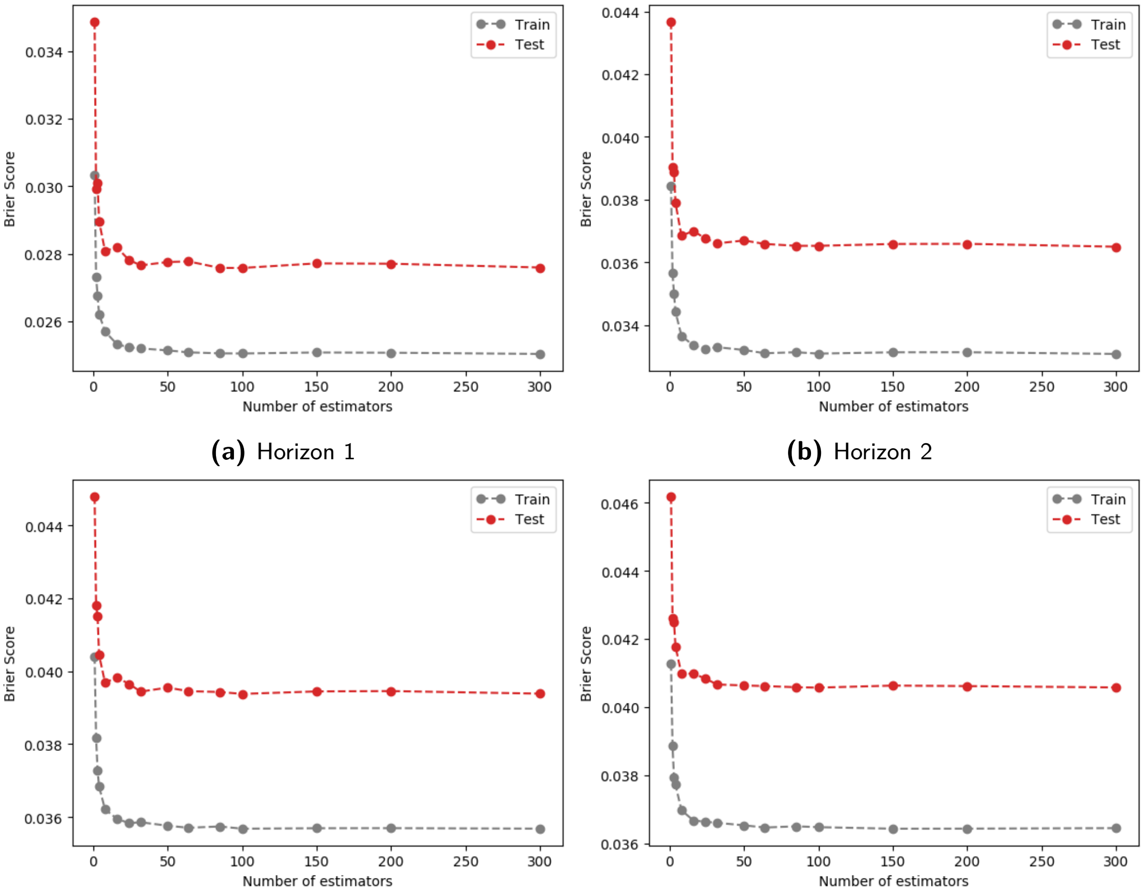
B-1 Final Model Configurations

Table B-1: Final model configurations.

Model	Parameter	Space
SARIMAX		
	p	1
	d	1
	q	4
	P	3
	D	0
	Q	1
	τ	'n'
	K	3
Logistic Regression CC		
	max. iterations	296
	tol	0.0006
	ρ	0.6
	C	0.006
Random Forest		
	D	90
	n_{sel}	8
	max. tree depth	16
	terminal node size	0.0001%[1]
Stochastic Gradient Boosting CC		
	B	120
	max. tree depth	5
	ν	0.1
	χ	0.8
	η	0.9
	γ_{min}	0.7
	min. child weight	6

[1] percentage of all training samples

B-2 Hyperparameter: Number of Trees (Random Forest)



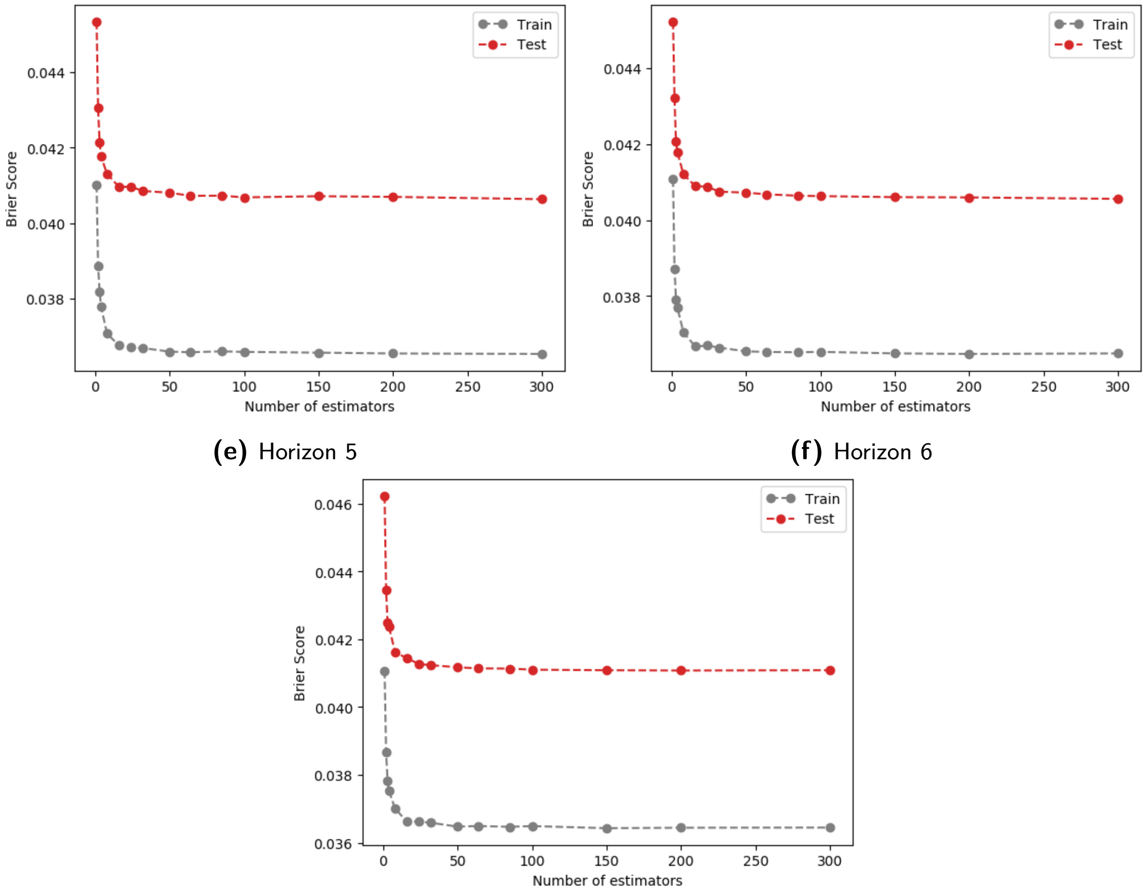
(a) Horizon 1

(b) Horizon 2

(c) Horizon 3

(d) Horizon 4

Figure B-1: Brier score per horizon for different number of trees in the random forest model.



(e) Horizon 5

(f) Horizon 6

(g) Horizon 7

Figure B-1: Brier score per horizon for different number of trees in the random forest model.

Appendix C

Model Performance

C-1 Forecasting

Table C-1: Model performance evaluation - 3250 customers.

Model	Horizon	AUC	BS	MAD	MSD	MAPE	MSLAR	StDv[1]
SARIMAX								
	1	-	-	10.25	-2.48	10.79	0.0204	4.16
	2	-	-	10.73	-2.79	11.27	0.0225	4.14
	3	-	-	10.81	-2.86	11.28	0.0233	4.29
	4	-	-	10.82	-2.86	11.31	0.0235	4.24
	5	-	-	10.85	-2.88	11.32	0.0236	4.38
	6	-	-	10.87	-2.90	11.36	0.0236	4.35
	7	-	-	10.96	-2.97	11.41	0.0237	4.42
Logistic Regression CC								
	1	0.8345	0.0489	16.18	-3.99	16.22	0.0419	5.24
	2	0.7759	0.0490	16.31	-4.01	16.35	0.0423	5.23
	3	0.7665	0.0491	16.35	-4.03	16.36	0.0424	5.12
	4	0.7660	0.0491	16.38	-4.11	16.36	0.0424	5.34
	5	0.7658	0.0491	16.40	-4.13	16.39	0.0424	5.32
	6	0.7657	0.0491	16.41	-4.13	16.39	0.0424	5.31
	7	0.7652	0.0491	16.41	-4.16	16.46	0.0427	5.42
Random Forest								
	1	0.9050	0.0299	10.96	-2.17	11.18	0.0236	3.67
	2	0.8579	0.0389	11.51	-2.20	11.81	0.0237	3.74
	3	0.8331	0.0418	11.56	-2.36	11.97	0.0237	3.62
	4	0.8220	0.0429	11.62	-2.40	12.01	0.0238	3.63
	5	0.8166	0.0430	11.64	-2.49	12.02	0.0240	3.89
	6	0.8140	0.0430	11.66	-2.70	12.02	0.0241	4.01
	7	0.8121	0.0431	11.76	-2.84	12.11	0.0261	3.99
Gradient Boosting CC								
	1	0.9081	0.0292	8.74	-0.12	9.13	0.0178	3.41
	2	0.8612	0.0382	9.47	-0.20	10.05	0.0186	3.31
	3	0.8380	0.0411	9.53	-0.41	10.16	0.0188	3.38
	4	0.8275	0.0421	9.64	-0.42	10.21	0.0189	3.30
	5	0.8231	0.0424	9.69	-0.68	10.21	0.0197	3.86
	6	0.8218	0.0425	9.69	-0.88	10.32	0.0198	3.79
	7	0.8209	0.0426	10.04	-1.13	10.66	0.0200	3.84

[1] Standard deviation ($\times 10^{-4}$) of the MSLAR

Table C-2: Model performance evaluation - 6500 customers.

Model	Horizon	AUC	BS	MAD	MSD	MAPE	MSLAR	StDv[1]
SARIMAX								
		-	-	19.76	-4.75	9.86	0.0183	3.58
		-	-	20.96	-5.32	10.37	0.0184	3.43
		-	-	20.97	-5.42	10.39	0.0184	3.56
		-	-	21.13	-5.64	10.40	0.0184	3.46
		-	-	21.14	-5.64	10.41	0.0185	3.66
		-	-	21.22	-5.70	10.44	0.0186	3.37
		-	-	21.26	-5.76	10.44	0.0191	3.42
Logistic Regression CC								
	1	0.8360	0.0445	30.14	-6.34	14.26	0.0323	4.75
	2	0.7664	0.0467	30.26	-6.42	14.46	0.0327	4.66
	3	0.7651	0.0468	30.36	-6.57	14.46	0.0328	4.91
	4	0.7642	0.0469	30.47	-6.57	14.48	0.0329	4.83
	5	0.7637	0.0469	30.58	-6.60	14.52	0.0329	4.83
	6	0.7635	0.0469	30.60	-6.66	14.53	0.0330	4.65
	7	0.7631	0.0469	30.66	-6.73	14.58	0.0332	4.71
Random Forest								
	1	0.9049	0.0313	21.76	-2.11	10.43	0.0178	3.23
	2	0.8579	0.0405	24.48	-2.24	11.85	0.0215	3.07
	3	0.8347	0.0433	24.55	-2.33	11.97	0.0217	3.02
	4	0.8230	0.0443	24.75	-2.33	12.03	0.0220	3.06
	5	0.8181	0.0445	25.03	-2.60	12.23	0.0226	3.94
	6	0.8157	0.0445	25.15	-2.92	12.29	0.0229	3.08
	7	0.8138	0.0445	25.50	-4.17	12.42	0.0231	3.17
Gradient Boosting CC								
	1	0.9083	0.0306	15.03	-0.36	7.51	0.0119	2.96
	2	0.8622	0.0398	17.38	-0.54	8.80	0.0140	2.83
	3	0.8387	0.0427	18.08	-1.03	9.06	0.0149	2.91
	4	0.8273	0.0437	18.51	-1.12	9.33	0.0154	2.92
	5	0.8216	0.0441	19.14	-1.93	9.58	0.0157	2.86
	6	0.8211	0.0442	19.50	-2.09	9.91	0.0168	2.76
	7	0.8190	0.0442	20.15	-2.19	9.98	0.0177	2.90

[1] Standard deviation ($\times 10^{-4}$) of the MSLAR

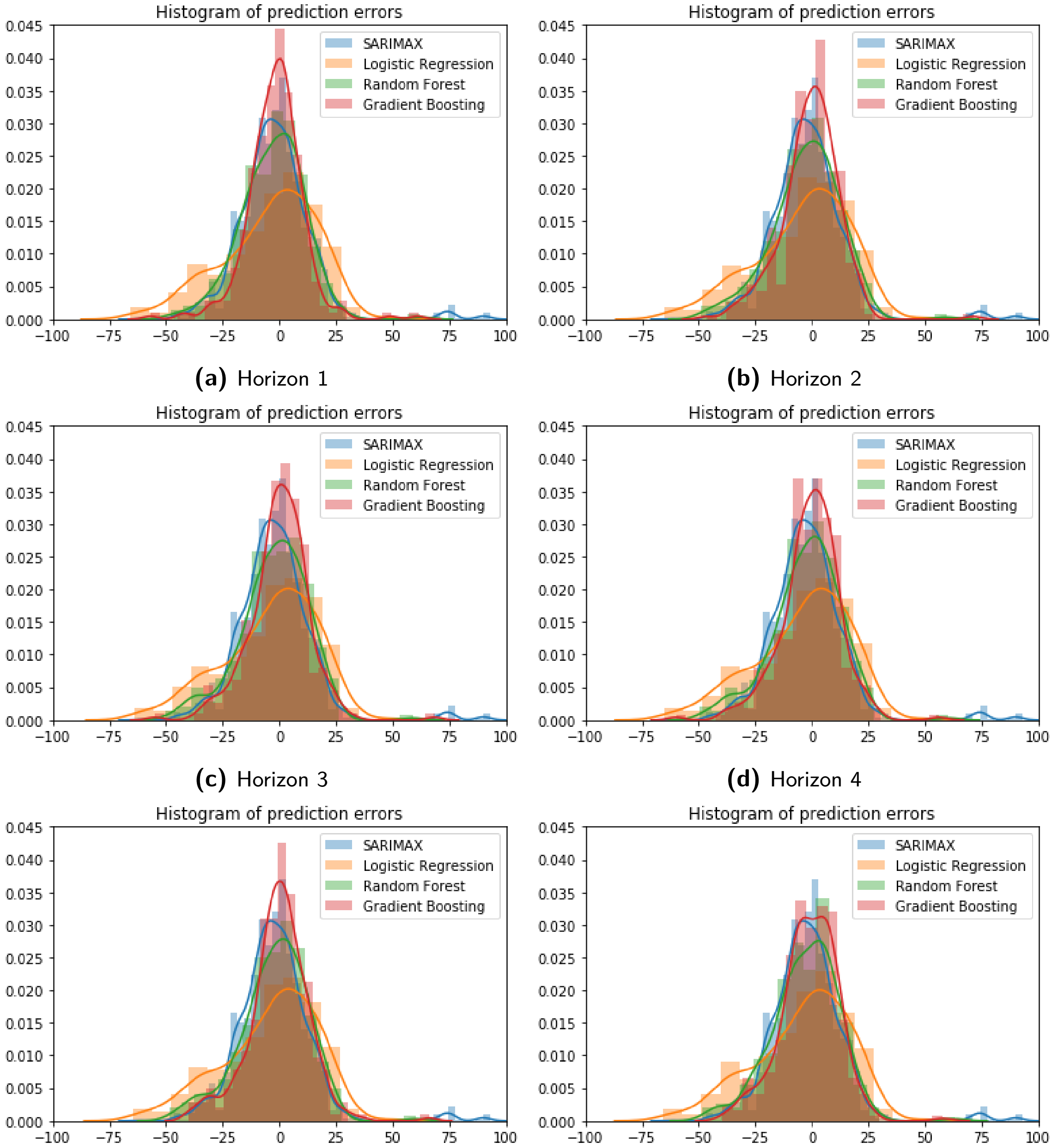
Table C-3: Model performance evaluation - 13000 customers.

Model	Horizon	AUC	BS	MAD	MSD	MAPE	MSLAR	StDv[1]
SARIMAX								
	1	-	-	30.08	-9.16	7.65	0.0151	2.23
	2	-	-	32.83	-10.48	8.27	0.0151	2.38
	3	-	-	33.17	-10.86	8.31	0.0152	2.54
	4	-	-	33.40	-10.98	8.33	0.0152	1.95
	5	-	-	33.44	-11.03	8.35	0.0152	2.14
	6	-	-	33.55	-11.14	8.36	0.0156	2.14
	7	-	-	33.63	-11.22	8.38	0.0160	2.49
Logistic Regression CC								
	1	0.8397	0.0426	48.53	-9.91	11.57	0.0213	3.57
	2	0.7808	0.0454	48.86	-10.04	11.61	0.0215	3.71
	3	0.7795	0.0455	48.89	-10.28	11.66	0.0215	3.58
	4	0.7770	0.0456	49.05	-10.62	11.67	0.0216	3.05
	5	0.7745	0.0456	49.10	-11.13	11.67	0.0216	3.30
	6	0.7723	0.0457	49.55	-11.25	11.76	0.0219	3.25
	7	0.7658	0.0428	49.55	-11.52	11.78	0.0220	3.37
Random Forest								
	1	0.9053	0.0310	39.38	-4.15	9.50	0.0148	2.92
	2	0.8584	0.0401	42.32	-4.59	10.20	0.0166	2.65
	3	0.8356	0.0429	42.50	-4.70	10.29	0.0168	2.13
	4	0.8249	0.0438	42.70	-4.81	10.35	0.0172	2.96
	5	0.8196	0.0441	42.85	-4.81	10.45	0.0175	2.49
	6	0.8174	0.0441	43.54	-5.49	10.61	0.0178	2.66
	7	0.8156	0.0441	44.32	-6.89	10.75	0.0180	2.36
Gradient Boosting CC								
	1	0.9100	0.0301	23.25	-1.87	5.94	0.0083	1.93
	2	0.8636	0.0392	26.17	-2.07	6.63	0.0107	2.16
	3	0.8406	0.0421	26.51	-2.13	6.70	0.0109	2.05
	4	0.8299	0.0431	28.77	-2.68	7.26	0.0129	2.29
	5	0.8239	0.0435	28.79	-3.83	7.31	0.0141	1.99
	6	0.8222	0.0437	30.26	-4.07	7.71	0.0146	1.95
	7	0.8214	0.0437	32.42	-4.34	8.07	0.0151	2.25

[1] Standard deviation ($\times10^{-4}$) of the MSLAR

C-2 Prediction Errors

C-2-1 Comparison of the prediction errors (3250 customers)



(a) Horizon 1

(b) Horizon 2

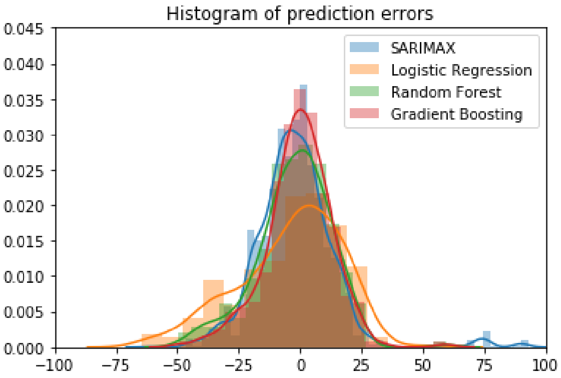
(c) Horizon 3

(d) Horizon 4

(e) Horizon 5

(f) Horizon 6

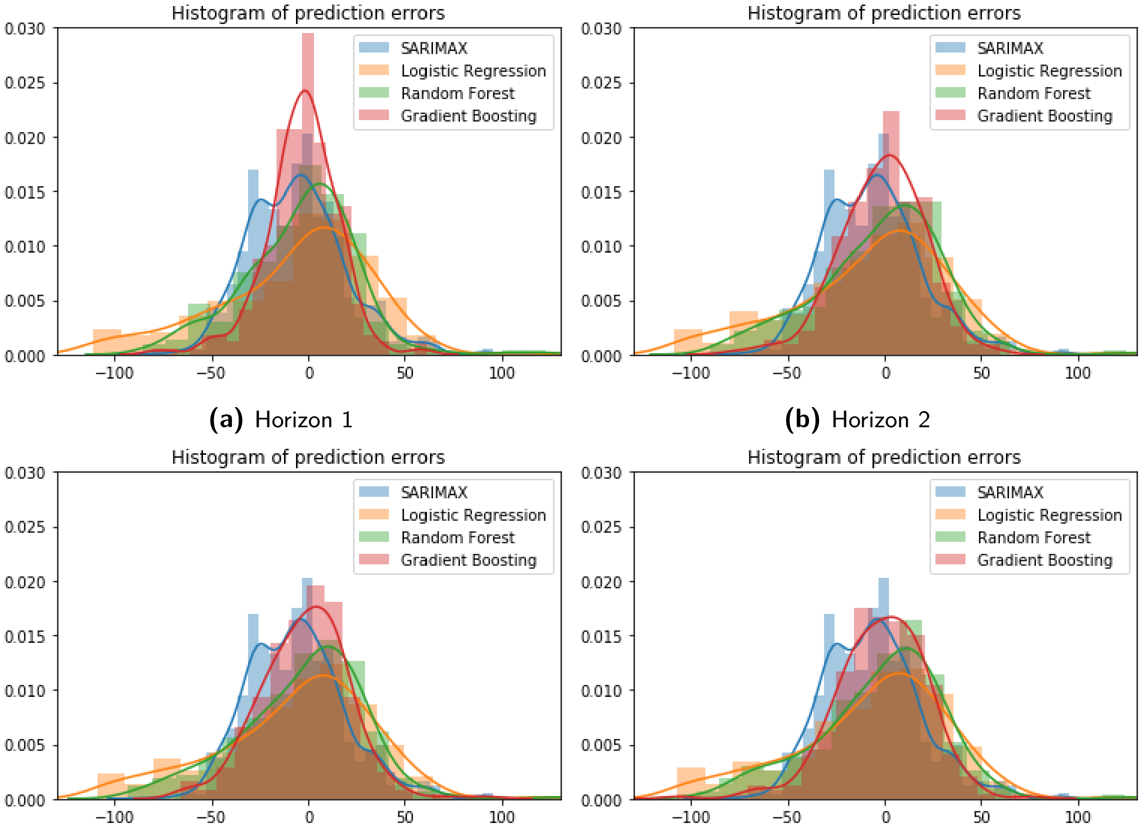
Figure C-1: Histogram plots of the prediction errors.



(g) Horizon 7

Figure C-1: Histogram plots of the prediction errors.

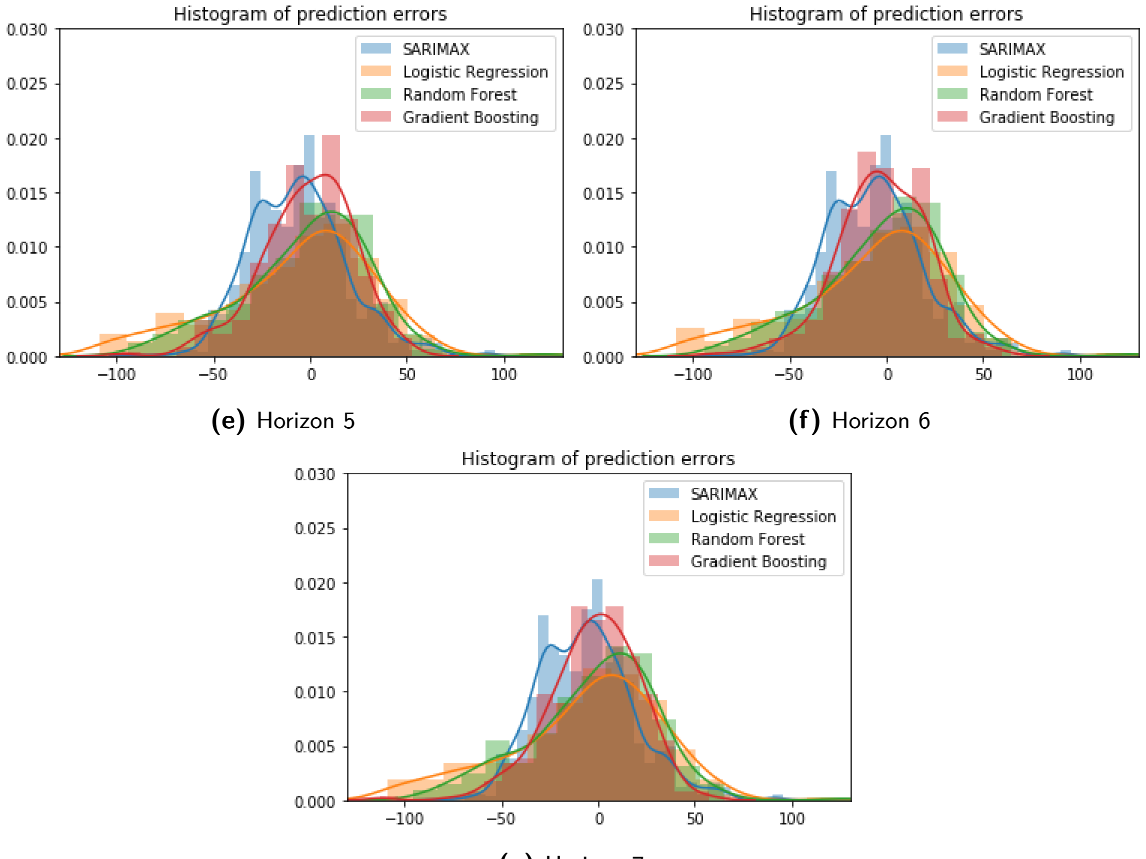
C-2-2 Comparison of the prediction errors (6500 customers)



(a) Horizon 1 **(b)** Horizon 2

(c) Horizon 3 **(d)** Horizon 4

Figure C-2: Histogram plots of the prediction errors.



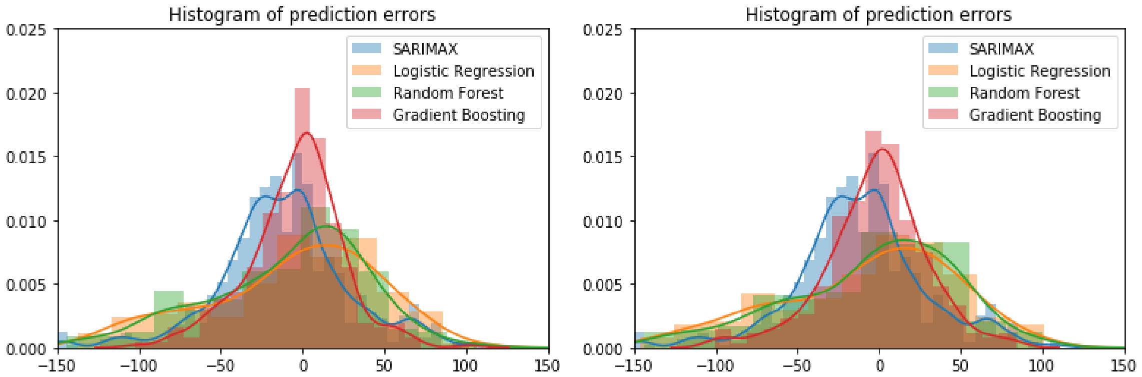
(e) Horizon 5

(f) Horizon 6

(g) Horizon 7

Figure C-2: Histogram plots of the prediction errors.

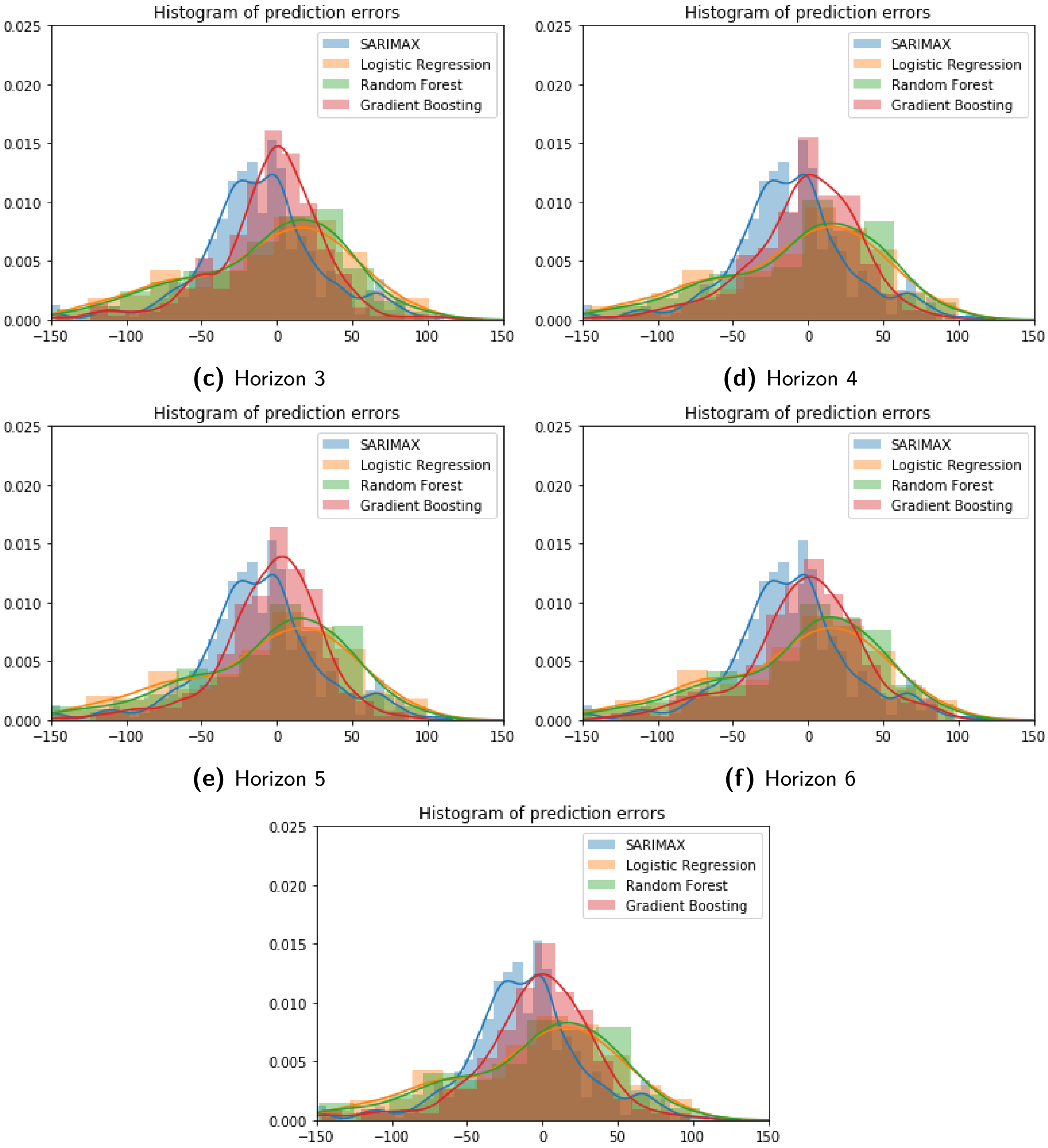
C-2-3 Comparison of the prediction errors (13000 customers)



(a) Horizon 1

(b) Horizon 2

Figure C-3: Histogram plots of the prediction errors.



(c) Horizon 3

(d) Horizon 4

(e) Horizon 5

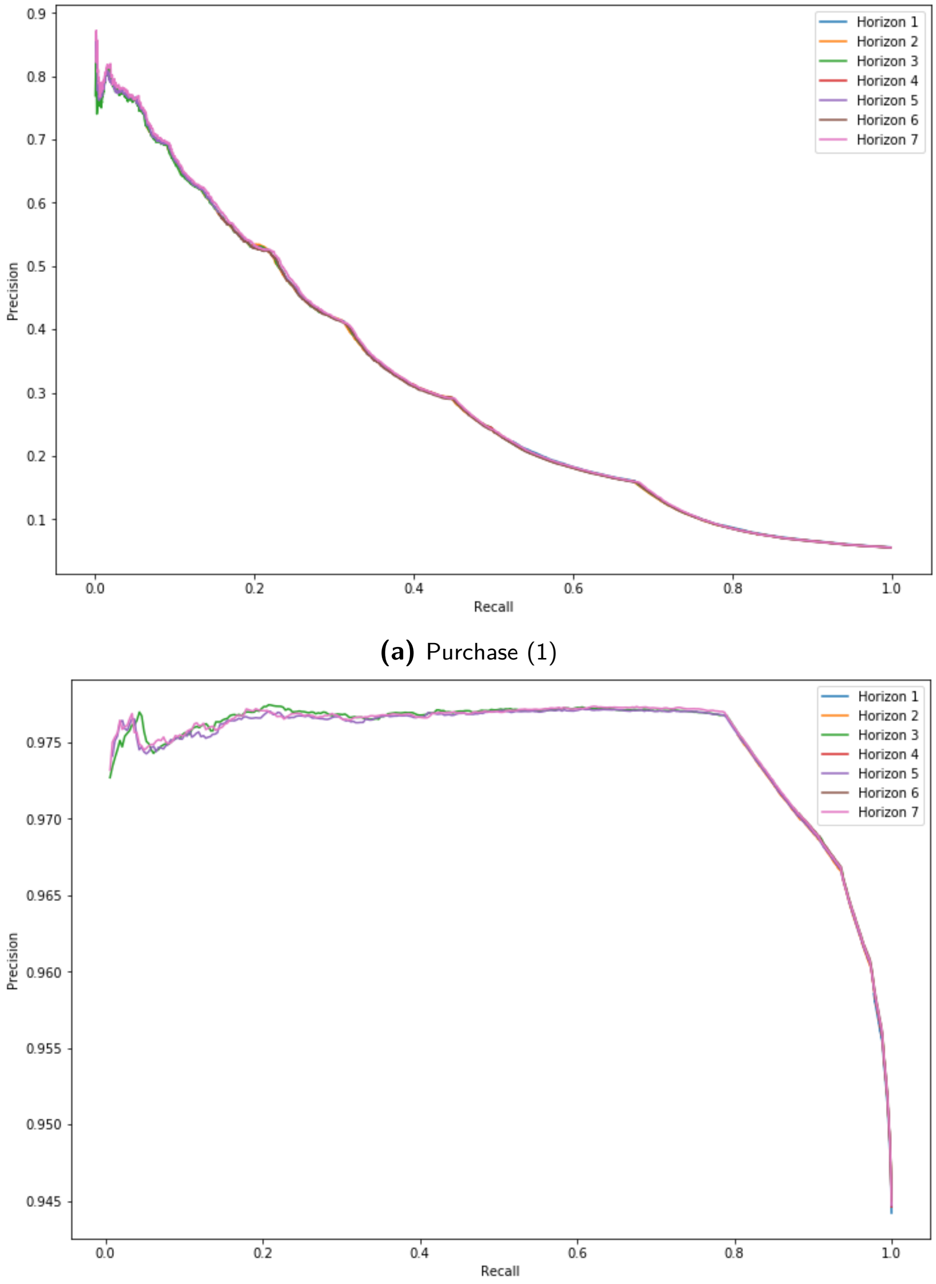
(f) Horizon 6

(g) Horizon 7

Figure C-3: Histogram plots of the prediction errors.

C-3 Classification

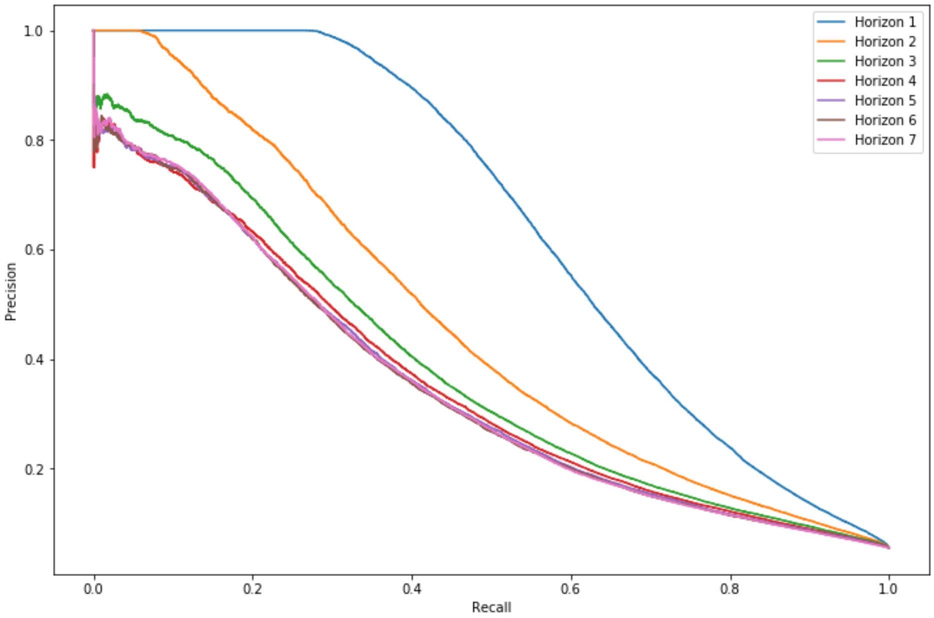
C-3-1 Precision and Recall plots (3250 customers)



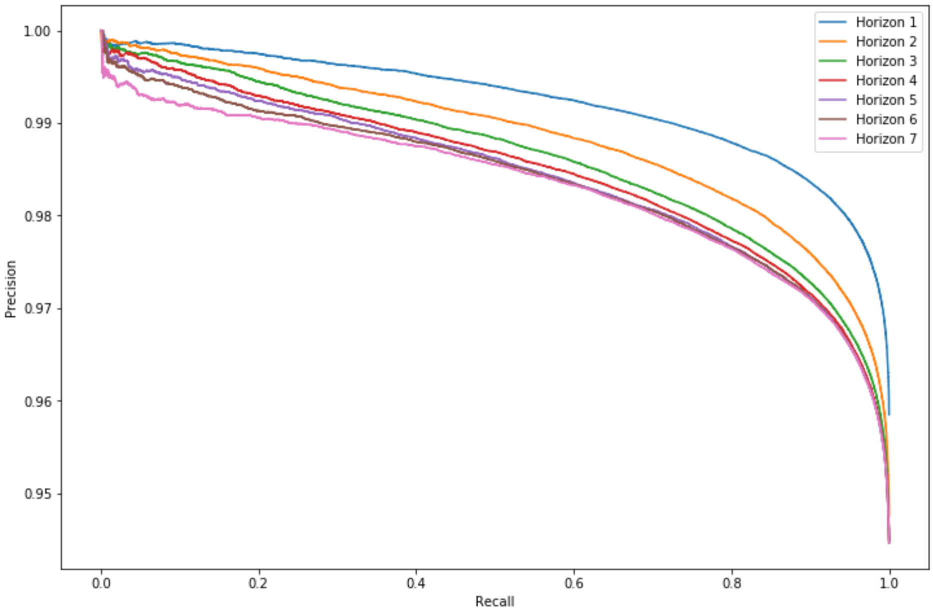
(a) Purchase (1)

(b) No Purchase (0)

Figure C-4: Precision and Recall curves of the logistic regression CC model.

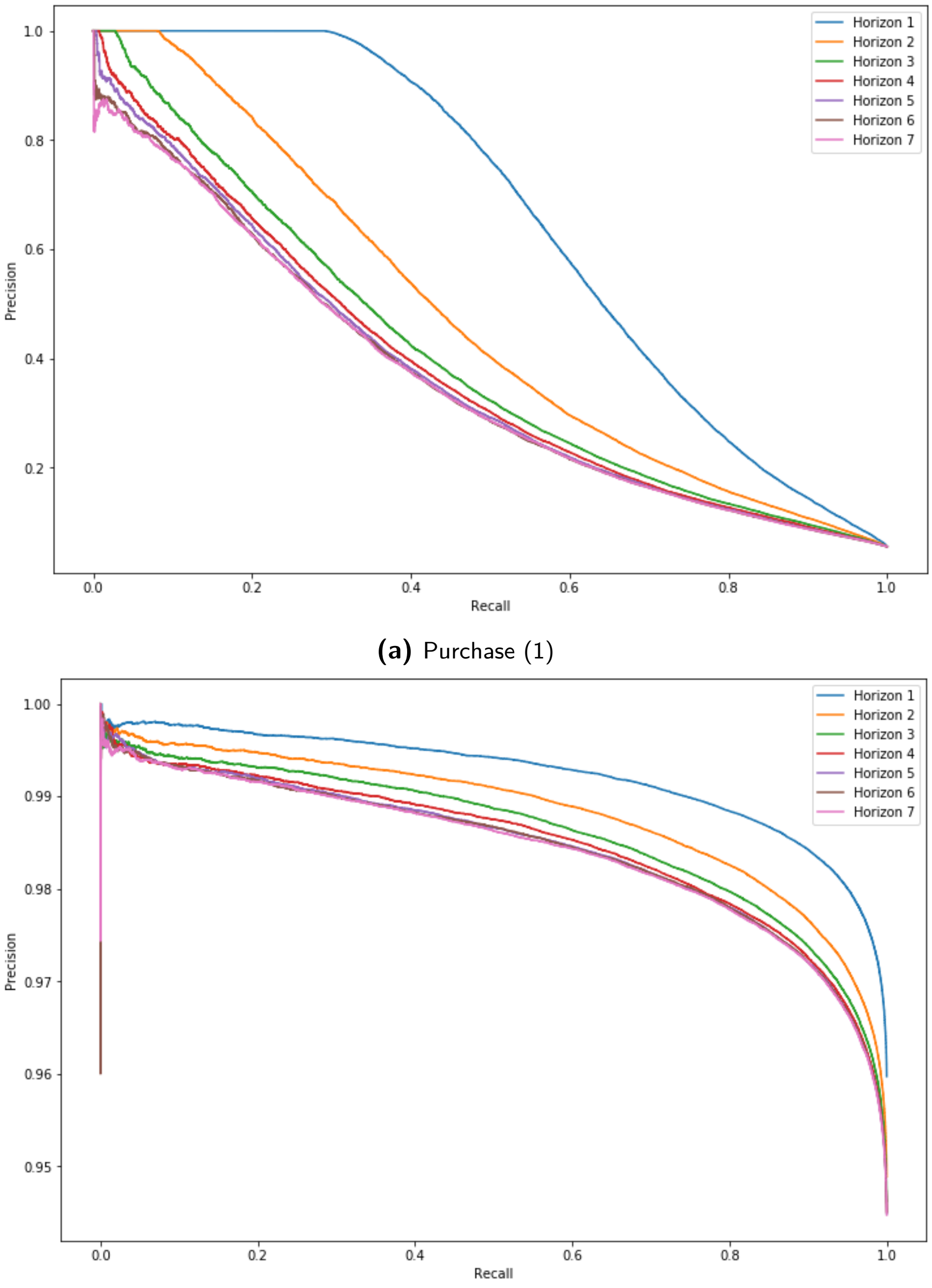


(a) Purchase



(b) No Purchase

Figure C-5: Precision and Recall curves of the random forest model.

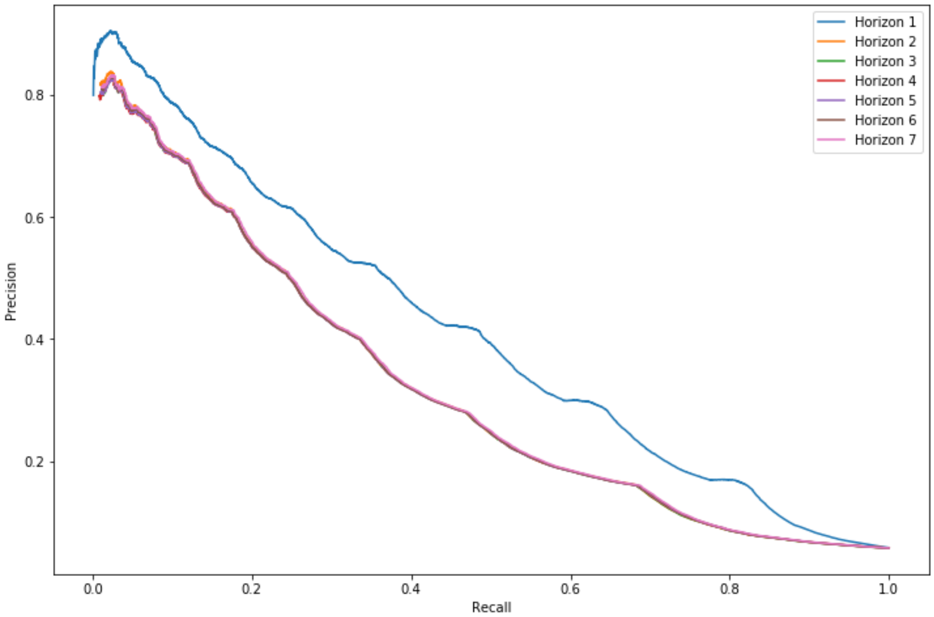


(a) Purchase (1)

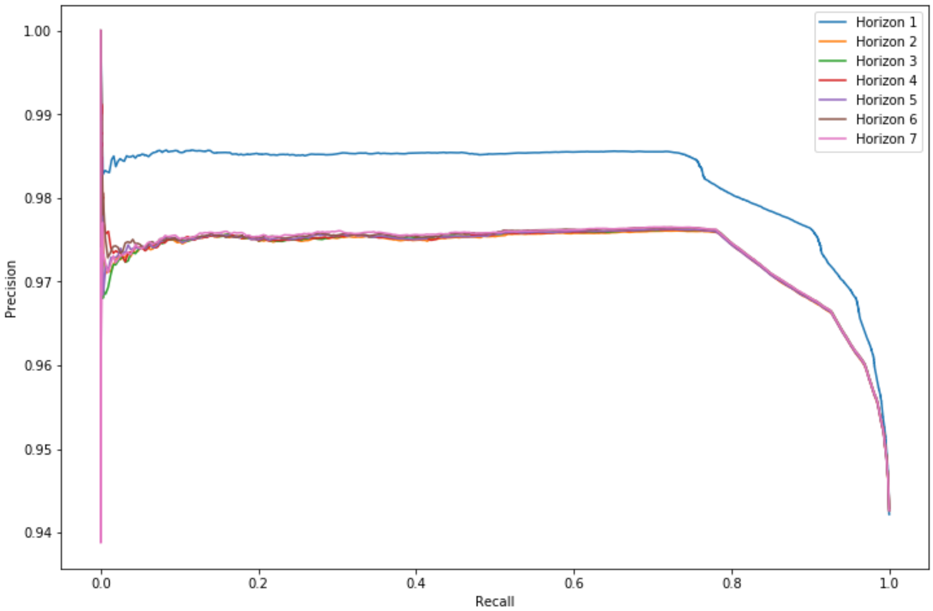
(b) No Purchase (0)

Figure C-6: Precision and Recall curves of the stochastic gradient boosting CC model.

C-3-2 Precision and Recall plots (6500 customers)

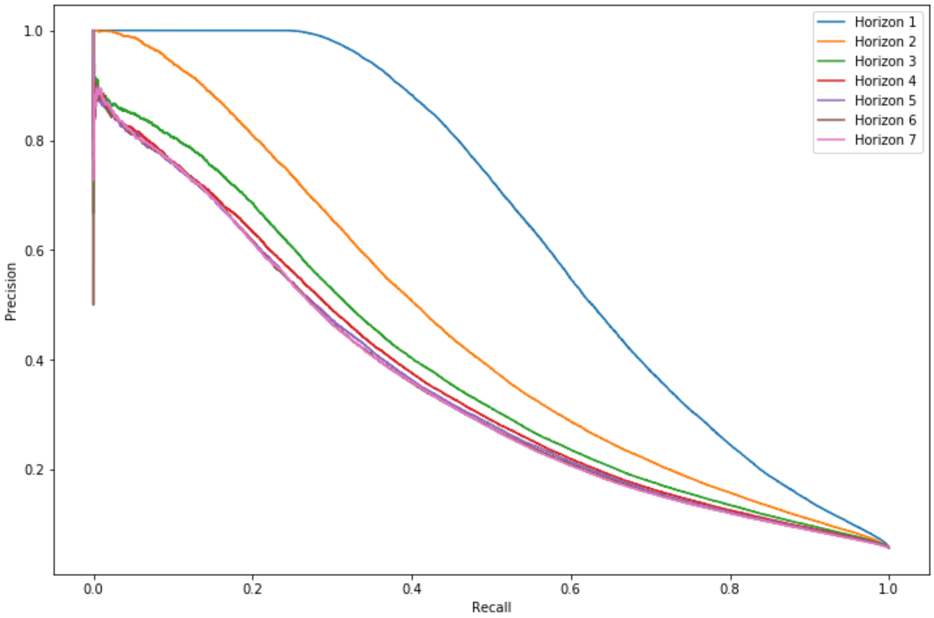


(a) Purchase (1)

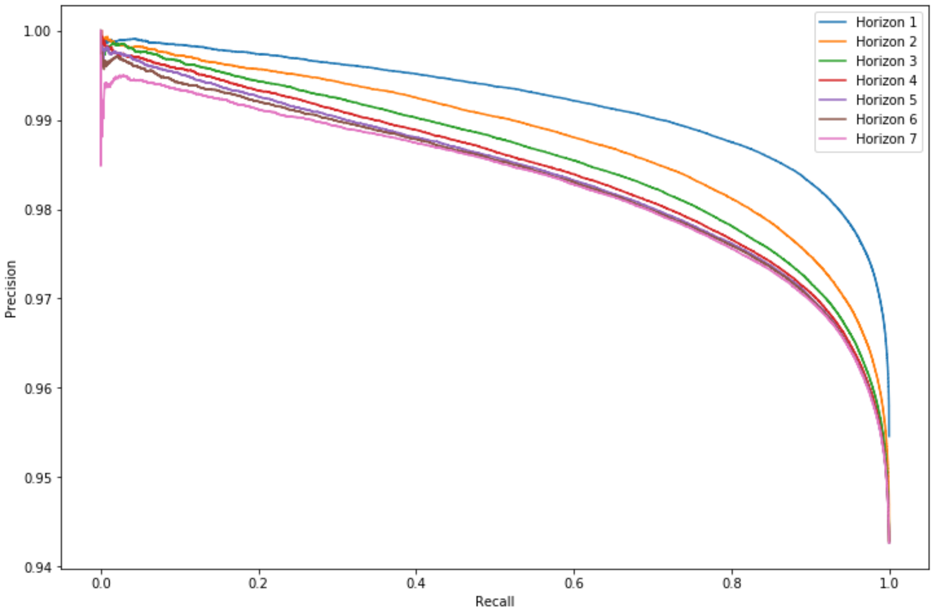


(b) No Purchase (0)

Figure C-7: Precision and Recall curves of the logistic regression CC model.

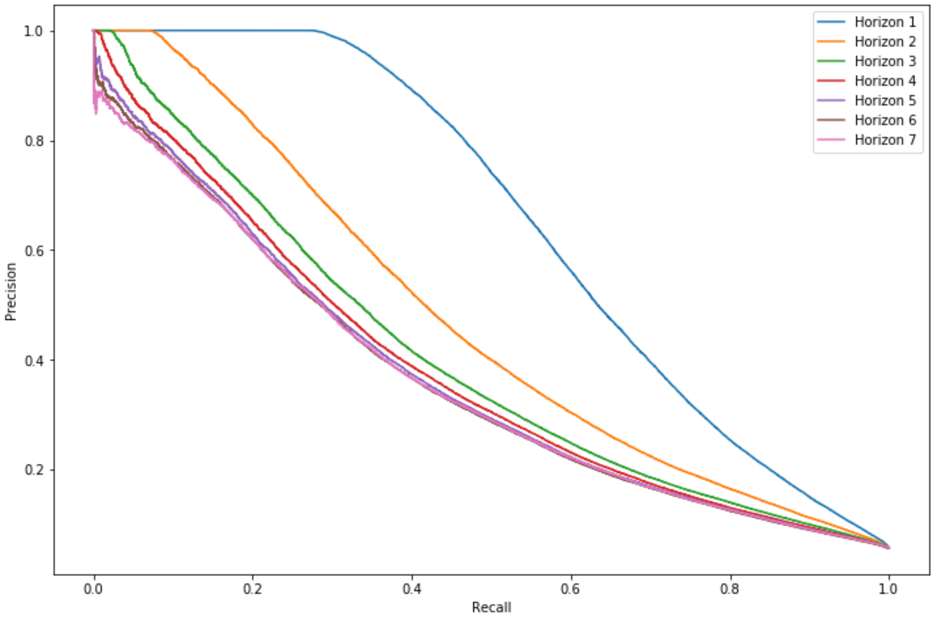


(a) Purchase

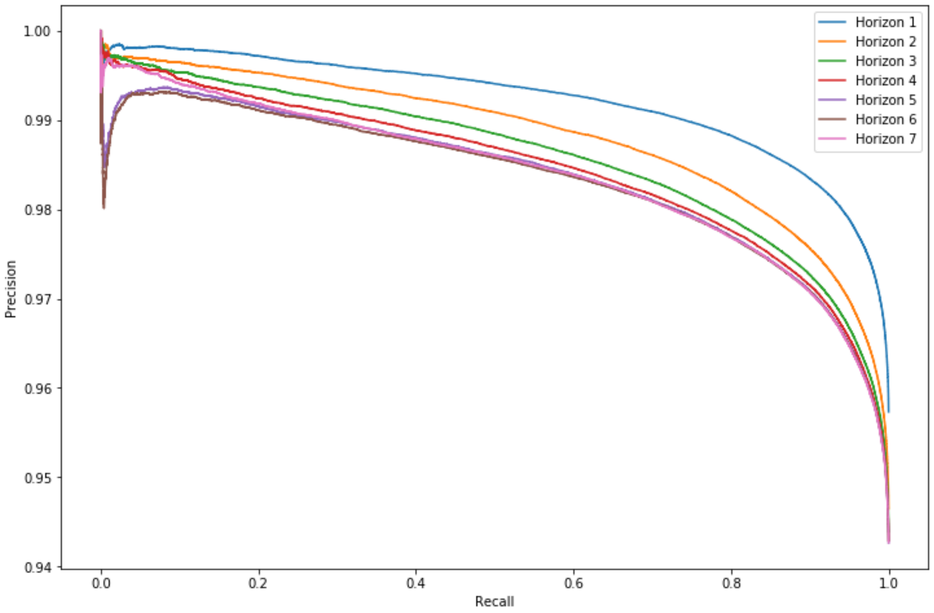


(b) No Purchase

Figure C-8: Precision and Recall curves of the random forest model.



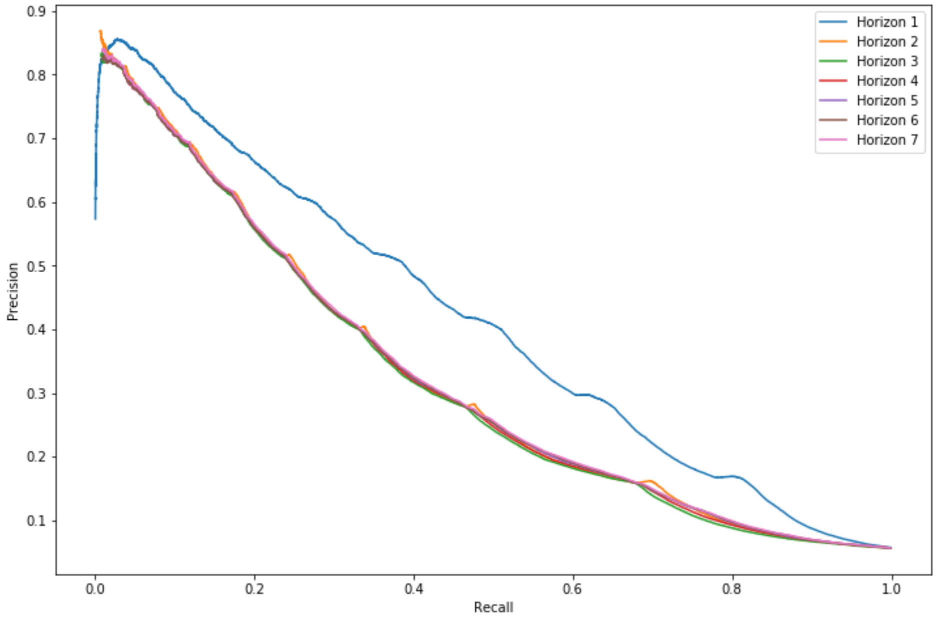
(a) Purchase (1)



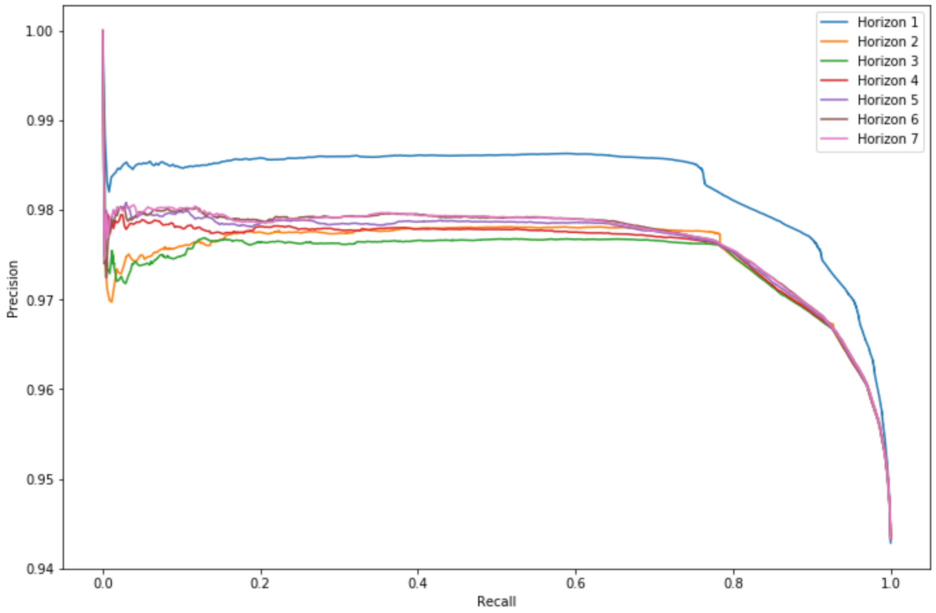
(b) No Purchase (0)

Figure C-9: Precision and Recall curves of the stochastic gradient boosting CC model.

C-3-3 Precision and Recall plots (13000 customers)

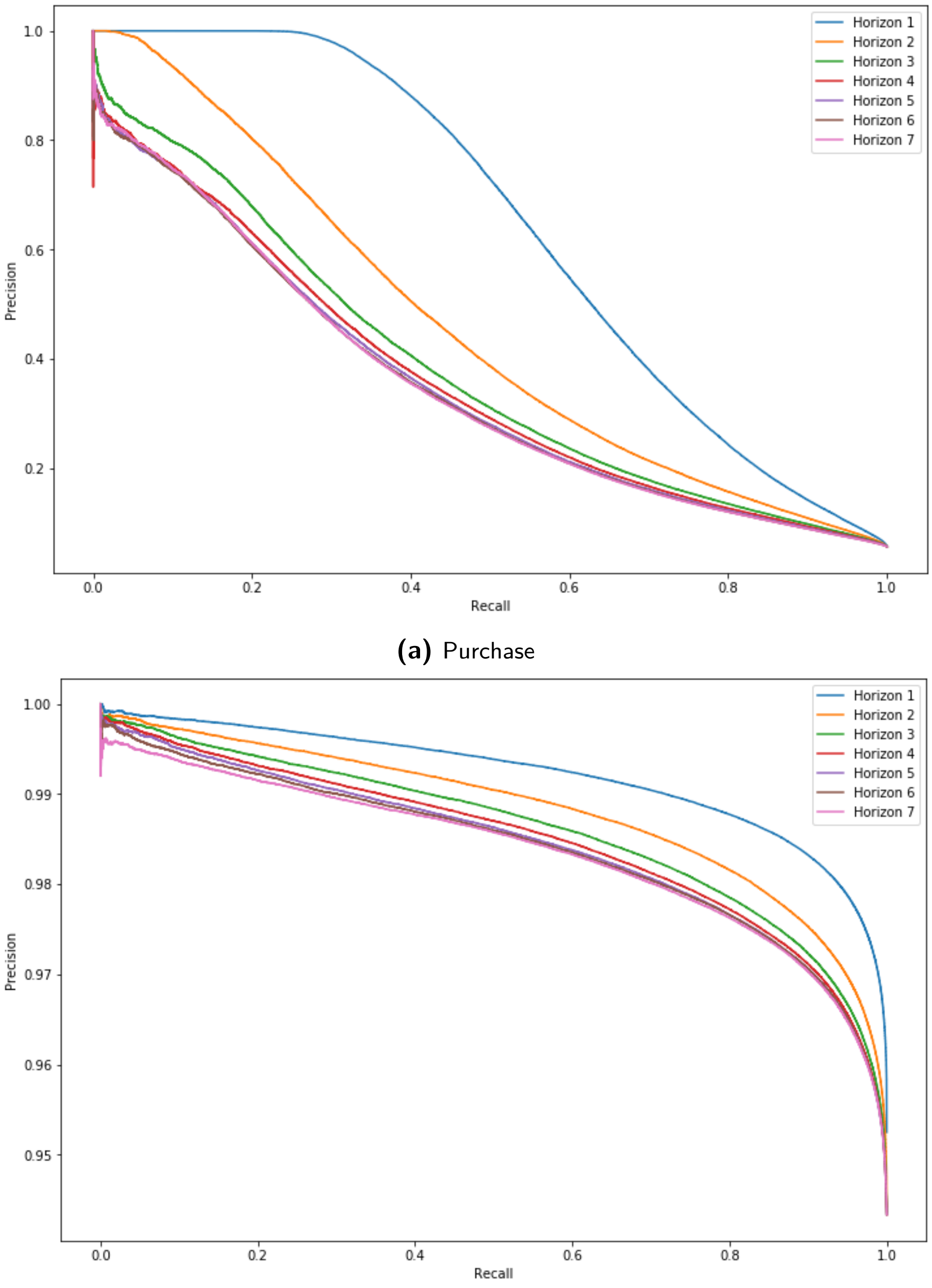


(a) Purchase (1)



(b) No Purchase (0)

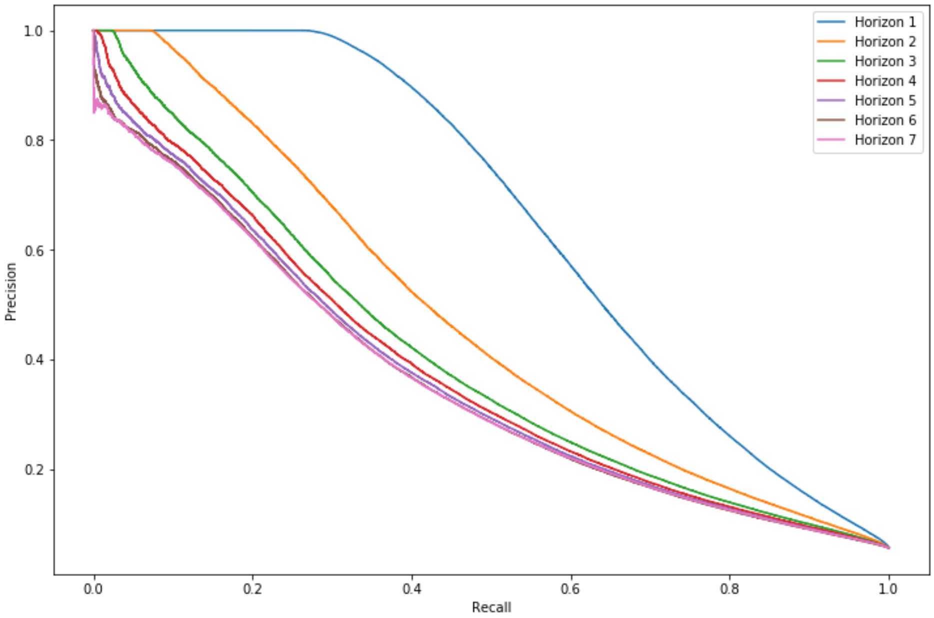
Figure C-10: Precision and Recall curves of the logistic regression CC model.



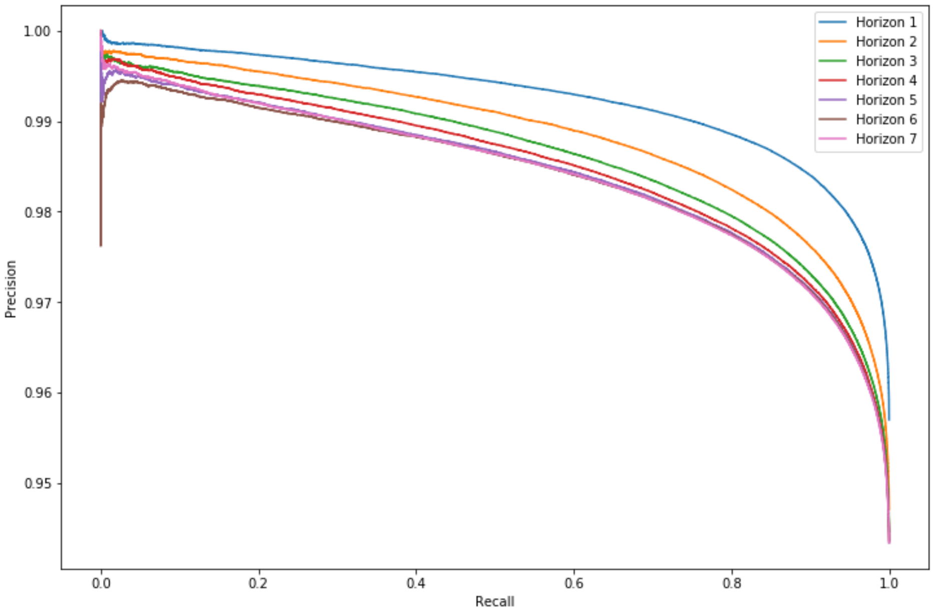
(a) Purchase

(b) No Purchase

Figure C-11: Precision and Recall curves of the random forest model.



(a) Purchase (1)



(b) No Purchase (0)

Figure C-12: Precision and Recall curves of the stochastic gradient boosting CC model.

Bibliography

[1] Bovas Abraham and Johannes Ledolter. *Statistical methods for forecasting*, volume 234. John Wiley & Sons, 2009.

[2] Deepak Agrawal and Christopher Schorling. Market share forecasting: An empirical comparison of artificial neural networks and multinomial logit model. *Journal of Retailing*, 72(4):383–407, 1996.

[3] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621, 2010.

[4] Hermine N Akouemo and Richard J Povinelli. Data improving in time series using arx and ann models. *IEEE Transactions on Power Systems*, 32(5):3352–3359, 2017.

[5] Ozden Gur Ali and Efe Pinar. Multi-period-ahead forecasting with residual extrapolation and information sharing—utilizing a multitude of retail series. *International Journal of Forecasting*, 32(2):502–517, 2016.

[6] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2009.

[7] J Scott Armstrong. Evaluating forecasting methods. In *Principles of forecasting*, pages 443–472. Springer, 2001.

[8] Miriam Ayer, H Daniel Brunk, George M Ewing, William T Reid, and Edward Silverman. An empirical distribution function for sampling with incomplete information. *The annals of mathematical statistics*, pages 641–647, 1955.

[9] Hiram C Barksdale and Jimmy E Hilliard. A cross-spectral analysis of retail inventories and sales. *The Journal of Business*, 48(3):365–382, 1975.

[10] João Francisco Barragan, Cristiano Hora Fontes, and Marcelo Embiruçu. A wavelet-based clustering of multivariate time series using a multiscale spca approach. *Computers & Industrial Engineering*, 95:144–155, 2016.

[11] Steven Bellman, Gerald L Lohse, and Eric J Johnson. Predictors of online buying behavior. *Communications of the ACM*, 42(12):32–38, 1999.

[12] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

[13] Joseph Berkson. Application of the logistic function to bio-assay. *Journal of the American Statistical Association*, 39(227):357–365, 1944.

[14] Michael A Berry and Gordon S Linoff. Mastering data mining: The art and science of customer relationship management. *Industrial Management & Data Systems*, 2000.

[15] J Mentzer-C Bienstock. *Sales forecasting management.* Sage Publications, 1998.

[16] Gail Blattenberger and Frank Lad. Separating the brier score into calibration and refinement components: A graphical exposition. *The American Statistician*, 39(1):26–32, 1985.

[17] Ruth N Bolton, Katherine N Lemon, and Peter C Verhoef. The theoretical underpinnings of customer asset management: a framework and propositions for future research. *Journal of the Academy of Marketing Science*, 32(3):271–292, 2004.

[18] Tonya Boone, Ram Ganeshan, Aditya Jain, and Nada R Sanders. Forecasting sales in the supply chain: Consumer analytics in the big data era. *International Journal of Forecasting*, 2018.

[19] George EP Box and Gwilym M Jenkins. Time series analysis forecasting and control. Technical report, WISCONSIN UNIV MADISON DEPT OF STATISTICS, 1970.

[20] L Breiman, JH Friedman, R Olshen, and CJ Stone. Classification and regression trees. 1984.

[21] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[22] Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[23] Robert Goodell Brown. Smoothing, forecasting and prediction of discrete time series. 1962.

[24] Randolph E Bucklin, James M Lattin, Asim Ansari, Sunil Gupta, David Bell, Eloise Coupey, John DC Little, Carl Mela, Alan Montgomery, and Joel Steckel. Choice and the internet: From clickstream to research stream. *Marketing Letters*, 13(3):245–258, 2002.

[25] Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications.* Springer Science & Business Media, 2011.

[26] Emerce B.V. Picnic investeert 100 miljoen euro in uitbreiding, 2017.

[27] Laura Calvet, Albert Ferrer, M Isabel Gomes, Angel A Juan, and David Masip. Combining statistical learning with metaheuristics for the multi-depot vehicle routing problem with market segmentation. *Computers & Industrial Engineering*, 94:93–104, 2016.

[28] Benjamin Paul Chamberlain, Angelo Cardoso, Chak H Liu, Roberto Pagliari, and Marc Peter Deisenroth. Customer lifetime value prediction using embeddings. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1753–1762. ACM, 2017.

[29] Mu-Chen Chen, Ai-Lun Chiu, and Hsu-Hwa Chang. Mining changes in customer behavior in retail marketing. *Expert Systems with Applications*, 28(4):773–781, 2005.

[30] Peng Chen, Aichen Niu, Duanyang Liu, Wei Jiang, and Bin Ma. Time series forecasting of temperatures using sarima: An example from nanjing. In *IOP Conference Series: Materials Science and Engineering*, volume 394, page 052024. IOP Publishing, 2018.

[31] Yen-Liang Chen, Chang-Ling Hsu, and Shih-Chieh Chou. Constructing a multi-valued and multi-labeled decision tree. *Expert Systems with Applications*, 25(2):199–209, 2003.

[32] Changqing Cheng, Akkarapol Sa-Ngasoongsong, Omer Beyca, Trung Le, Hui Yang, Zhenyu Kong, and Satish TS Bukkapatnam. Time series forecasting for nonlinear and non-stationary processes: a review and comparative study. *Iie Transactions*, 47(10):1053–1071, 2015.

[33] Ching-Hsue Cheng and You-Shyang Chen. Classifying the segmentation of customer value via rfm model and rs theory. *Expert systems with applications*, 36(3):4176–4184, 2009.

[34] Alain Yee Loong Chong, Eugene Ch'ng, Martin J Liu, and Boying Li. Predicting consumer product demands via big data: the roles of online promotional marketing and online reviews. *International Journal of Production Research*, 55(17):5142–5156, 2017.

[35] Ching-Wu Chu and Guoqiang Peter Zhang. A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of production economics*, 86(3):217–231, 2003.

[36] Amanda Clare and Ross D King. Knowledge discovery in multi-label phenotype data. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 42–53. Springer, 2001.

[37] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: a seasonal-trend decomposition. *Journal of official statistics*, 6(1):3–73, 1990.

[38] Angeline G Close and Monika Kukar-Kinney. Beyond buying: Motivations behind consumers' online shopping cart use. *Journal of Business Research*, 63(9-10):986–992, 2010.

[39] Javier Contreras, Rosario Espinola, Francisco J Nogales, and Antonio J Conejo. Arima models to predict next-day electricity prices. *IEEE transactions on power systems*, 18(3):1014–1020, 2003.

[40] Mario Cools, Elke Moons, and Geert Wets. Investigating the variability in daily traffic counts through use of arimax and sarimax models: assessing the effect of holidays on two site locations. *Transportation Research Record*, 2136(1):57–66, 2009.

[41] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[42] Kristof Coussement and Dirk Van den Poel. Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert systems with applications*, 34(1):313–327, 2008.

[43] Keely L Croxton, Douglas M Lambert, Sebastián J García-Dastugue, and Dale S Rogers. The demand management process. *The International Journal of Logistics Management*, 13(2):51–66, 2002.

[44] Yanwei Cui, Rogatien Tobossi, and Olivia Vigouroux. Modelling customer online behaviours with neural networks: applications to conversion prediction and advertising retargeting. *arXiv preprint arXiv:1804.07669*, 2018.

[45] Estela Bee Dagum. modeling, forecasting and seasonally adjusting economic time series with the x-11 arima method. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 27(3/4):203–216, 1978.

[46] Andrea Dal Pozzolo, Olivier Caelen, Reid A Johnson, and Gianluca Bontempi. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166. IEEE, 2015.

[47] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473, 2006.

[48] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[49] Sahibsingh A Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4):325–327, 1976.

[50] Behzad Eftekhar, Kazem Mohammad, Hassan Eftekhar Ardebili, Mohammad Ghodsi, and Ebrahim Ketabchi. Comparison of artificial neural network and logistic regression models for prediction of mortality in head trauma based on initial clinical data. *BMC medical informatics and decision making*, 5(1):1–8, 2005.

[51] Eurostat. E-commerce statistics for individuals, 2018.

[52] Robert Alan Fildes, Shaohui Ma, and Stephan Kolassa. Retail forecasting: research and practice. 2018.

[53] David F Findley, Brian C Monsell, William R Bell, Mark C Otto, and Bor-Chung Chen. New capabilities and methods of the x-12-arima seasonal-adjustment program. *Journal of Business & Economic Statistics*, 16(2):127–152, 1998.

[54] Digital Commerce 360 (formerly Internet Retailer). Us ecommerce sales grow 14.9% in 2019, 2019.

[55] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[56] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[57] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[58] Christopher Gan, V Limsombunchao, Michael D Clemes, and Yong YA Weng. Consumer choice prediction: Artificial neural networks versus logistic models. 2005.

[59] Neil A Gershenfeld and Neil Gershenfeld. *The nature of mathematical modeling*. Cambridge university press, 1999.

[60] Víctor Gómez and Agustín Maravall. Seasonal adjustment and signal extraction in economic time series. *A course in time series analysis*, pages 202–247, 2001.

[61] Joel Grus. *Data science from scratch: first principles with python*. O'Reilly Media, 2019.

[62] Venkat Gudivada, Amy Apon, and Junhua Ding. Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *International Journal on Advances in Software*, 10(1):1–20, 2017.

[63] Anders Gustafsson, Michael D Johnson, and Inger Roos. The effects of customer satisfaction, relationship commitment dimensions, and triggers on customer retention. *Journal of marketing*, 69(4):210–218, 2005.

[64] Martin T Hagan and Suzanne M Behr. The time series approach to short term load forecasting. *IEEE transactions on power systems*, 2(3):785–791, 1987.

[65] James D Hamilton. *Time series analysis*, volume 2. Princeton New Jersey, 1994.

[66] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[67] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[68] Tim Hill, Marcus O'Connor, and William Remus. Neural network models for time series forecasts. *Management science*, 42(7):1082–1092, 1996.

[69] Jakob Huber, Alexander Gossmann, and Heiner Stuckenschmidt. Cluster-based hierarchical demand forecasting for perishable goods. *Expert systems with applications*, 76:140–151, 2017.

[70] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[71] Rob J Hyndman et al. Another look at forecast-accuracy metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, 4(4):43–46, 2006.

[72] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[73] William James, Charles A McMellon, and Gladys Torres-Baumgarten. Dogs and cats rule: A new insight into segmentation. *Journal of Targeting, Measurement and Analysis for Marketing*, 13(1):70–77, 2004.

[74] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[75] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[76] Farid Kadri, Fouzi Harrou, Sondès Chaabane, and Christian Tahon. Time series modelling and forecasting of emergency department overcrowding. *Journal of medical systems*, 38(9):107, 2014.

[77] Kishana R Kashwan and CM Velu. Customer segmentation using clustering and data mining techniques. *International Journal of Computer Theory and Engineering*, 5(6):856, 2013.

[78] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression*. Springer, 2002.

[79] E Kreyszig. Advanced engineering mathematics. fourth edition, 1979.

[80] Jochen Kruppa, Alexandra Schwarz, Gerhard Arminger, and Andreas Ziegler. Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications*, 40(13):5125–5131, 2013.

[81] Vineet Kumar and Werner J Reinartz. *Customer relationship management: A databased approach*. Wiley Hoboken, 2006.

[82] RJ Kuo, LM Ho, and Clark M Hu. Integration of self-organizing feature map and k-means algorithm for market segmentation. *Computers & Operations Research*, 29(11):1475–1493, 2002.

[83] Tobias Lang and Matthias Rettenmeier. Understanding consumer behavior with recurrent neural networks. In *Workshop on Machine Learning Methods for Recommender Systems*, 2017.

[84] Bart Larivière and Dirk Van den Poel. Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, 29(2):472–484, 2005.

[85] Christian Leistner, Amir Saffari, and Horst Bischof. Miforests: Multiple-instance learning with randomized trees. In *European Conference on Computer Vision*, pages 29–42. Springer, 2010.

[86] Aurélie Lemmens and Christophe Croux. Bagging and boosting classification trees to predict churn. *Journal of Marketing Research*, 43(2):276–286, 2006.

[87] Michael Lewis. The influence of loyalty programs and short-term promotions on customer retention. *Journal of marketing research*, 41(3):281–292, 2004.

[88] Sarah Lichtenstein, Baruch Fischhoff, and Lawrence D Phillips. Calibration of probabilities: The state of the art to 1980. Technical report, DECISION RESEARCH EUGENE OR, 1981.

[89] Gilles Louppe. Understanding random forests: From theory to practice. *arXiv preprint arXiv:1407.7502*, 2014.

[90] James D Malley, Jochen Kruppa, Abhijit Dasgupta, Karen G Malley, and Andreas Ziegler. Probability machines. *Methods of information in medicine*, 51(01):74–81, 2012.

[91] Andrés Martínez, Claudia Schmuck, Sergiy Pereverzyev Jr, Clemens Pirker, and Markus Haltmeier. A machine learning framework for customer purchase prediction in the non-contractual setting. *European Journal of Operational Research*, 281(3):588–596, 2020.

[92] Habshah Midi, Saroje Kumar Sarkar, and Sohel Rana. Collinearity diagnostics of binary logistic regression model. *Journal of Interdisciplinary Mathematics*, 13(3):253–267, 2010.

[93] John R Miglautsch. Thoughts on rfm scoring. *Journal of Database Marketing & Customer Strategy Management*, 8(1):67–72, 2000.

[94] Wendy W Moe and Peter S Fader. Dynamic conversion behavior at e-commerce sites. *Management Science*, 50(3):326–335, 2004.

[95] Norizan Mohamed, Maizah Hura Ahmad, and Zuhaimy Ismail. Improving short term load forecasting using double seasonal arima model. 2011.

[96] Michelle A Morganosky and Brenda J Cude. Consumer demand for online food retailing: is it really a supply side issue? *International Journal of Retail & Distribution Management*, 30(10):451–458, 2002.

[97] Michael C Mozer, Richard Wolniewicz, David B Grimes, Eric Johnson, and Howard Kaushansky. Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. *IEEE Transactions on neural networks*, 11(3):690–696, 2000.

[98] John F Muth. Optimal properties of exponentially weighted forecasts. *Journal of the american statistical association*, 55(290):299–306, 1960.

[99] Nicolò Navarin, Beatrice Vincenzi, Mirko Polato, and Alessandro Sperduti. Lstm networks for data-aware remaining time prediction of business process instances. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7. IEEE, 2017.

[100] Alexandru Niculescu-Mizil and Rich Caruana. Obtaining calibrated probabilities from boosting. In *UAI*, page 413, 2005.

[101] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005.

[102] Guangli Nie, Wei Rowe, Lingling Zhang, Yingjie Tian, and Yong Shi. Credit card churn forecasting by logistic regression and decision tree. *Expert Systems with Applications*, 38(12):15273–15285, 2011.

[103] Balaji Padmanabhan, Zhiqiang Zheng, and Steven O Kimbrough. Personalization from incomplete data: what you don't know can hurt. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 154–163. ACM, 2001.

[104] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[105] Nicolas Poggi, Toni Moreno, Josep Lluis Berral, Ricard Gavaldà, and Jordi Torres. Web customer modeling for automated session prioritization on high traffic sites. In *International Conference on User Modeling*, pages 450–454. Springer, 2007.

[106] Irfan Pratama, Adhistya Erna Permanasari, Igi Ardiyanto, and Rini Indrayani. A review of missing values handling methods on time-series data. In *2016 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 1–6. IEEE, 2016.

[107] Loes Raasveld. Predicting invites conversion: Obtaining well-calibrated and discriminative short-term purchase probabilities in the context of waiting lists. Master's thesis, Erasmus School of Economics, 8 2019.

[108] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333, 2011.

[109] SA Roberts. A general class of holt-winters type forecasting models. *Management Science*, 28(7):808–820, 1982.

[110] Lior Rokach and Oded Z Maimon. *Data mining with decision trees: theory and applications*, volume 69. World scientific, 2008.

[111] Olivia Parr Rud. *Data mining cookbook: Modeling data for marketing, risk, and customer relationship management.* John Wiley & Sons, 2001.

[112] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[113] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

[114] Mark R Segal. Machine learning benchmarks and random forest regression. 2004.

[115] Raffi Sevlian and Ram Rajagopal. A scaling law for short term load forecasting on varying levels of aggregation. *International Journal of Electrical Power & Energy Systems*, 98:350–361, 2018.

[116] J Shahrabi, SS Mousavi, and M Heydar. Supply chain demand forecasting: A comparison of machine learning techniques and traditional methods. *Journal of Applied Sciences*, 9(3):521–527, 2009.

[117] Katerina Shapoval and Thomas Setzer. Next-purchase prediction using projections of discounted purchasing sequences. *Business & Information Systems Engineering*, 60(2):151–166, 2018.

[118] Simon Sheather. *A modern approach to regression with R*. Springer Science & Business Media, 2009.

[119] LF Simmons. Time-series decomposition using the sinusoidal model. *International Journal of Forecasting*, 6(4):485–495, 1990.

[120] Catarina Sismeiro and Randolph E Bucklin. Modeling purchase behavior at an e-commerce web site: A task-completion approach. *Journal of marketing research*, 41(3):306–323, 2004.

[121] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[122] Hee Seok Song, Jae kyeong Kim, and Soung Hie Kim. Mining the change of customer behavior in an internet shopping mall. *Expert Systems with Applications*, 21(3):157–168, 2001.

[123] Carolin Strobl, James Malley, and Gerhard Tutz. An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods*, 14(4):323, 2009.

[124] Grażyna Suchacka, Magdalena Skolimowska-Kulig, and Aneta Potempa. A k-nearest neighbors method for classifying user sessions in e-commerce scenario. *Journal of Telecommunications and Information Technology*, 2015.

[125] Grażyna Suchacka and Sławomir Stemplewski. Application of neural network to predict purchases in online store. In *Information Systems Architecture and Technology: Proceedings of 37th International Conference on Information Systems Architecture and Technology–ISAT 2016–Part IV*, pages 221–231. Springer, 2017.

[126] Yanmin Sun, Andrew KC Wong, and Mohamed S Kamel. Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):687–719, 2009.

[127] SuperVastgoed, Supermarkt en Ruimte, and Strabo. Versnelling in groei online supermarktomzet, 2018.

[128] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems*, 39(1):43–62, 1997.

[129] Pang-Ning Tan. *Introduction to data mining*. Pearson Education India, 2018.

[130] Ling Tang, Anying Wang, Zhenjing Xu, and Jian Li. Online-purchasing behavior forecasting with a firefly algorithm-based svm model considering shopping cart use. *Eurasia Journal of Mathematics, Science and Technology Education*, 13(12):7967–7983, 2017.

[131] Chris Tofallis. A better measure of relative prediction accuracy for model selection and model estimation. *Journal of the Operational Research Society*, 66(8):1352–1362, 2015.

[132] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[133] Alice M Tybout, Tim Calkins, and Philip Kotler. *Kellogg on branding: The marketing faculty of The Kellogg School of Management.* Wiley Hoboken, NJ, 2005.

[134] Stylianos I Vagropoulos, GI Chouliaras, Evaggelos G Kardakos, Christos K Simoglou, and Anastasios G Bakirtzis. Comparison of sarimax, sarima, modified sarima and ann-based models for short-term pv generation forecasting. In *2016 IEEE International Energy Conference (ENERGYCON)*, pages 1–6. IEEE, 2016.

[135] Francisco José Valverde-Albacete, Jorge Carrillo-de Albornoz, and Carmen Peláez-Moreno. A proposal for new evaluation metrics and result visualization technique for sentiment analysis tasks. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 41–52. Springer, 2013.

[136] Dirk Van den Poel and Wouter Buckinx. Predicting online-purchasing behaviour. *European journal of operational research*, 166(2):557–575, 2005.

[137] Dirk Van den Poel and Bart Lariviere. Customer attrition analysis for financial services using proportional hazard models. *European journal of operational research*, 157(1):196–217, 2004.

[138] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91, 2006.

[139] Michel Wedel and Wagner A Kamakura. *Market segmentation: Conceptual and methodological foundations*, volume 8. Springer Science & Business Media, 2012.

[140] Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

[141] Buckinx Wouter and Dirk Van den Poel. Customer base analysis: Partial defection of behaviorally-loyal clients in a non-contractual fmcg retail setting. *European Journal of Operational Research*, 164(1):252–268, 2005.

[142] Guo-en Xia and Wei-dong Jin. Model of customer churn prediction on support vector machine. *Systems Engineering-Theory & Practice*, 28(1):71–77, 2008.

[143] Yaya Xie, Xiu Li, EWT Ngai, and Weiyun Ying. Customer churn prediction using improved balanced random forests. *Expert Systems with Applications*, 36(3):5445–5449, 2009.

[144] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.

[145] G Peter Zhang, B Eddy Patuwo, and Michael Y Hu. A simulation study of artificial neural networks for nonlinear time-series forecasting. *Computers & Operations Research*, 28(4):381–396, 2001.

[146] Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.