Erasmus University Rotterdam
Erasmus School of Economics
Master Thesis Econometrics & Management Science
Track: Business Analytics and Quantitative Marketing

Cold-Start Promotional Demand Forecasting with Contrastive Explanations in an E-grocery Setting

Tomas Heemskerk (463319)

Supervisor:	dr. Olga Kuryatnikova
Second assessor:	dr. Hakan Akyuz
Date final version:	11th March 2024

Abstract

This study addresses the challenge of forecasting cold-start promotional demand for an online grocery retailer through the development and application of a contrastive regression model. The research pursues four primary objectives: (i) developing an interpretable model for cold-start promotions, (ii) comparing the model against baseline methods, (iii) extending its application beyond cold-start scenarios, and (iv) applying the model to a real-life business case to enhance the promotional demand forecasting process of Picnic, a leading e-grocery retailer. To achieve these objectives, the study makes use of a contrastive framework that combines the CatBoost algorithm with a k-nearest neighbor search, and extends this framework to accommodate heterogeneous feature data and inter-category training. Results indicate that the contrastive regression model is on par with established baseline methods and largely outperforms Picnic's manual forecasting process in terms of cold-start forecasting accuracy and computational effort, while also providing additional post-hoc explainability of individual forecasts. Moreover, the model shows potential to add value even beyond application on strictly cold-start promotions, showing superior accuracy for at least the first three promotions of a grocery article. Application of the model to Picnic's business case suggests substantial potential to improve the forecasting operation, reducing analyst workload while improving accuracy and still maintaining the ability to check the rationale behind and reliability of forecasts. Overall, the study underlines the effectiveness of the contrastive regression model in forecasting promotional demand for online grocery retailers, offering actionable insights for (e-)grocery retailers, and contributing to the advancement of forecasting methods that provide additional post-hoc explainability of the output. Further research should primarily aim to improve the contrastive regressor's training set-up and nearest neighbor search, and explore the use of asymmetrical loss functions to decrease the level of underforecasting and enhance practical added value in a grocery forecasting operation.

Keywords: Retail demand forecasting, cold-start forecasting, retail promotions, contrastive explanations, interpretable machine learning, e-grocery, regression trees, nearest neighbour search

Contents

Acknowledgements

I would like to express my sincere gratitude to all the people that guided me along the journey of conducting this master thesis research and writing the final report.

First and foremost, I want to thank my supervisor at the Erasmus University of Rotterdam, dr. Olga Kuryatnikova, for her constructive sharp eye, insightful sparring sessions and constant effort in providing valuable feedback on the project. Her critical questions and suggestions challenged me to dig deeper, providing an invaluable driver behind the quality of the project.

I am also very grateful to my supervisor at Picnic, David van der Meer, for his warm and personal mentorship, continuous feedback, and for providing me with the opportunity to immediately implement my findings in a real business setting. His constant push to concretise my insights and his real-world business perspective are an essential element of the value this thesis brings, and thanks to his time and effort the project's deliverables will live on at Picnic.

Furthermore, I would also like to thank dr. Hakan Akyuz from the Erasmus University of Rotterdam for taking an additional critical view of this master thesis project as a second assessor, and Loes Raasveld, Thijs de Lange, Maarten Sukel, and Giorgia Tandoi from Picnic for the insightful discussions and sharing of relevant expertise.

Lastly, I wouldn't have been able to follow and successfully complete the academic path of these last seven years without the unconditional support and love from my family and friends.

Chapter 1

Introduction

Article demand forecasting in retail focuses on predicting how many articles need to be in store at a certain moment to fulfill all customer demand. Muriana (2017) shows that in the grocery retail sector particularly, computing an accurate demand forecast is crucial: sales rotation is high (articles are only in stock for several days or weeks before they are sold) and a significant share of articles is perishable, with shelf lives often less than a week. As a consequence, under-forecasting immediately leads to unsatisfied customers, while overforecasting more than often leads to overcrowded distribution channels and product waste (Christensen et al., 2021).

To complicate matters even more, grocery retailers frequently use promotion mechanisms such as price discounts, highlighted displays and recipe showcases to increase customer demand of a specific set of articles. This rise in demand leads to higher article sales, but also higher sales of complements, lower sales of substitutes, purchase postponement, and stockpiling (Anderson & Fox, 2019). Zhang and Wedel (2009) show that these effects can be even stronger for online grocery retailers, as optimized displaying and personalization lead to higher effectiveness. This disruption of the regular demand pattern caused by a promotion makes forecasting more complex as promotional uplifts vary greatly among product categories, data on promotional periods is sparse, and errors are magnified due to the increased sales volumes (Fildes et al., 2022). Moreover, the consequences of forecast errors are more severe in times of promotions compared to regular sales weeks: going out-of-stock for a successful promotion damages customer satisfaction even more than usual, while unsuccessful promotions result in excess stock that is hard to sell when sales levels return to baseline.

A widely used approach to model article demand during promotions considers a certain baseline demand level for regular (non-promotion) periods, and then assumes the demand to grow with a certain promotional uplift factor whenever an article is put in promotion (Blattberg & Neslin, 1993). As the change of customer behaviour caused by a promotion, and hence the resulting uplift, is very article-specific, the most intuitive way to forecast promotional demand for an article is by looking at the uplift from historical promotions of that same article. Assuming that the relation between promotion mechanism and resulting promotional uplift remains stable over time, and assuming that the baseline demand level is known, then forecasting promotional demand for an article is relatively straightforward. However, the problem becomes more complicated when the article promotion is of the "cold-start" type, meaning that little to no historical promotion data from that same article is available. In absence of this data, an

alternative approach to forecasting demand for a cold-start article promotion is to use data on similar historical promotions from other articles (e.g., involving an article that is from the same category, or applying the same promotion mechanism). Similarity between the forecasted article promotion and historical article promotions can be based on similarity in many variables, such as product category, baseline demand, discount depth, regular selling price, time of the year, or the way the promotion was showcased. Importance of each of these variables in defining similarity between promotions differs strongly across markets and article categories, so a tailored approach is likely to be needed. Furthermore, when historical promotion data of the article in question is sparse or unreliable, it is hard to determine whether the resulting cold-start promotional demand forecast is reasonable. As demand forecasts are often used to automatically order new stock, it should be possible to somehow assess a forecast's reliability and logic. Therefore, methods that forecast demand for cold-start promotions should not only be accurate, but also provide an interpretable form of reasoning or explanation behind the forecast. For example, knowing which historical promotions were deemed similar and which variables were important in calculating this similarity can help analysts determine whether a forecast can be trusted or should be adjusted.

This research is done in cooperation with Picnic, a European e-grocery retailer that often encounters the problem of cold-start promotional demand forecasting in their business operation. Picnic sells their products solely via a dedicated online app and delivers orders directly at the customer's door. In order to prepare capacity along the whole distribution channel and ensure suppliers have enough stock for an upcoming article promotion, promotional demand forecasts at Picnic need to be computed at least five weeks ahead. However, the current article demand forecasting process is not yet able to forecast promotional demand for this longer-term horizon. Instead, promotional demand forecasts are computed manually by forecasting analysts using data on historical promotions and business expertise. Although the error rate of manual forecasting is on an acceptable level, automating the process is a high priority due to the significant operational workload it causes. First initiatives focused on building a relatively simple model that computes the forecast of an upcoming promotion as the average of historical promotions from the same article. Performance of this approach shows to be good for frequent and well-established article promotions, because many useful datapoints are available. However, accuracy shows to be significantly worse for promotions involving new articles or less-established discount mechanisms (i.e, article promotions of the "cold-start" type). Currently, Picnic does not have a model that can provide an accurate demand forecast for these cold-start article promotions. The requirements for such a model to be implemented in the business operation of Picnic are twofold: not only should the forecasting error be low enough to give a reasonable first indication of expected demand, but the rationale behind the forecasts should also be clear such that analysts can challenge and potentially adjust unreliable forecasts.

All the above leads to the following four main objectives of this research:

1. **Develop a model that provides interpretable demand forecasts for cold-start promotions.** This involves designing a method that can handle the lack of historical data associated with cold-start promotions and still compute an accurate forecast. To do so, features that contribute to cold-start forecasting performance in an e-grocery setting should be selected and their individual importance scores evaluated. Furthermore, a crucial prerequisite of the model is to provide a certain level of interpretability of its intrinsic decision making and final output.

2. **Extend the application area of the model beyond strictly cold-start promotions.** To increase the applicability of the cold-start model, its use will be extended to forecasting promotions for articles that have already been in promotion before. To evaluate the effect, the performance of the model will be assessed for varying degrees of "coldness" (i.e. varying amounts of historical promotions availabe from the same article).

3. **Compare the developed model with baseline methods.** The goal here is to compare the cold-start forecasting model with established baseline methods to learn how they relate in terms of forecasting performance, computational load, and interpretability of the output. As a minimal requirement, the forecasting performance should be at least on par with, but preferably higher than the current manual process at Picnic. The comparison is done for different article categories to evaluate the model's performance for various subsets of the assortment and assess its applicability at Picnic.

4. **Apply the developed model to an e-grocery business case.** To assess the model's interpretability and ease of implementation, an example is shown of a real-life business situation. The goal here is to have a concrete view on how the model can be used in a regular forecasting operation, and to display its added value to e-grocery companies. Additionally, this exercise aims at identifying areas of improvement for the model and providing relevant managerial insights.

Chapter 2

Literature review

2.1 Promotional demand forecasting in grocery retail

Forecasting demand during promotion periods in grocery retail has been the focus of numerous academic papers in the past. First methods were based on the simple "base-times-lift" principle, where promotional demand is calculated as the product of baseline demand and a certain uplift factor: Chase (1994) described how estimating an assortment- or category-wide uplift factor for weak, medium, or strong promotional effects can help to more accurately forecast customer demand. This approach was improved by Cooper et al. (1999), introducing a 67-variable regression model that uses article-level data on average sales during historical promotions. Variables included information about pricing, promotional events and level of display. Two prominent limitation of their approach are the fact that the model can not forecast promotional demand for new articles, and the forecast error increases when an existing article is promoted in a new way (e.g., a new discount mechanism, or a different way of displaying the article).

Özden Gür Ali et al. (2009) provide a solid literature overview and evaluation of 30 different methods for article demand forecasting during promotions in a grocery retail setting. The methods vary in complexity of the model (simple exponential smoothing versus regression trees), extensiveness of features (only recent sales and promotion data versus complex marketing features), and level of aggregation (forecasting demand per store versus total aggregated demand). They found that simple time series models perform well in regular, non-promotion weeks, but regression trees with complex features outperform the more simple models during promotion periods. Donselaar et al. (2016) compare moving average-based time series models with regression methods in forecasting promotion demand of perishable grocery items. They predict the model target, in this case promotional uplift factor, based on features concerning price, promotion mechanism, and baseline demand. They found the largest accuracy improvement after distinguishing between "routine" categories (i.e. with a stable demand process and large number of datapoints) and "non-routine" categories. More specifically, forecasting for routine categories can best be done by regressing only on datapoints from the category itself, while forecasting for non-routine categories benefits from also including datapoints from other categories. Ma et al. (2016) developed a methodological framework to use both intra- and inter-category promotional data to improve forecast accuracy of article-level promotional sales in retail, where main focus was on dealing with the high dimensionality that this data usually displays. They conclude that

95% of the accuracy improvement they achieved can be attributed to the intra-category data, while only 5% was added by the inter-category data. More recently, Bojer et al. (2019) compared several regularized regression methods with (ensembles of) decision trees in their ability to accurately forecast company-level promotion demand with a four-week horizon. For articles with historical information, XGBoost with category-level pooling and feature engineering shows best performance. For articles with little to no relevant historical data they observe similar forecast errors for all methods, with LASSO-regression slightly outperforming the others. As an interesting area for further research they propose to explore whether the same results hold for weekly-level sales, or for a different forecast horizon. Falatouri et al. (2022) compared the performance of models based on Seasonal Autoregressive Integrated Moving Average (SARIMA) and Long Short-Term Memory (LSTM) in forecasting demand of fruits and vegetables in grocery retail. They found that LSTM performed better for articles with stable sales history, while SARIMA outperforms for seasonal articles. Futhermore, including promotional data as an exogenous variable in SARIMA significantly improved forecasting accuracy.

2.2 Cold-start demand forecasting

The methods mentioned above naturally fail to provide accurate forecasts for cold-start promotions, because they heavily rely on data from historical promotions from the same article. With the occurrence of cold-start situations increasing and computational possibilities extending, literature has focused more on solving this problem of sparse historical data. Wen et al. (2017) combined Sequence-to-Sequence Neural Networks, Quantile Regression and Direct Multi-Horizon Forecasting in a powerful framework for probabilistic multi-step time series regression; the method is successfully tested on a cold-start demand forecasting case for Amazon, and outperformed well-established methods. Chauhan et al. (2020) introduced a modified Dynamic Key-Value Memory Network for cold-start forecasting of e-commerce sales, which was based on the idea that a model can learn most from related articles, and outperformed conventional LSTM by 25%. Dai and Huang (2021) introduced a cold-start forecasting approach with new inter-article similarity measures and feature random search, building upon the idea that similar articles have similar sales. They found that their method outperforms established machine learning methods in predicting sales in the first months after product launch. Xu et al. (2021) proposed a time series-aware Heterogenous Graph Attention Network that includes a new Category-Property-Value feature, which also takes into account product characteristics such as brand and category. Tests on an industrial sales dataset show that their method deems effective. Fatemi et al. (2023) aknowledged that deep learning-based models do not naturally capture causal relationships between dependent variables, and that Granger causality can be an effective tool to help characterize these interdependencies. However, they rightfully state that application of the Granger causality principle is difficult in cold-start forecasting problems where historical data is sparse or absent. To solve this, they introduced the Cold Causal Demand Forecasting framework, which combines two main components to create a causal forecasting model specifically for cold start problems: first a causal graph representing the cause-effect relationships between variables, and second a forecasting structure consisting of Graph Neural Networks,

Long Short-Term Memory Networks (LSTM), and a dense Neural Network that generates the forecasts. Their method outperformed baseline methods such as standard LSTM and Graph Neural Networks in forecasting cold-start network traffic for 200 Google services in several datacenters. Although the framework uses a similarity-based approach to leverage historical data of existing datacenters to forecast for new datacenters, it provides relatively little and hard to understand insights in which factors contributed to this datacenter similarity, and which variables were predominant in generating the final forecast.

2.3 Interpretable machine learning

The above mentioned literature on cold-start forecasting all have the disadvantage of not providing an intuitive explanation of how the final forecast was computed. Especially more state-of-the-art forecasting architectures involving a Neural Network, LSTM or Temporal Fusion Transformer (TFT) generally perform very well but lack direct interpretability of the output needed to assess how reliable a cold-start forecast is. Linardatos et al. (2021) rightfully note that recent surges in forecasting performance often involve making models more complex, turning them into opaque "black box" systems and losing direct interpretability. This direct interpretability of the model's input data, importance of each feature, and underlying calculations is crucial for analysts to intuitively understand, adjust and explain to others the final demand forecast. A universal definition of the term "interpretability" in this sense does not exist and previous work is not conclusive on the correct way to assess this. Lipton (2016) argues that the desire for interpretability of a model arises when the direct objective of a learning algorithm (i.e. computing accurate forecasts on unseen test data) does not fully cover the needs of the user. He splits the term "interpretability" into transparency (how does the model work?) and post-hoc explainability (besides the predictions, what else does the model tell me?), and argues that together with the evaluation metrics these two terms are essential to characterize a model. On the other hand, Lipton warns that the aversion against opaque learning algorithms should not unjustifiably hamper development of new methods that outperform simpler models, and suggests that the need for interpretability is tested rigorously. Concerning post-hoc explainability, Miller (2018) provides a framework based on contrastive explanations where the difference condition forms the basis of the answer to the question: "Why did the model output P, rather than Q?". This contrastive approach shows to be an intuitive way of explaining the output of a regression model: it specifically elaborates on the information that helped the model differentiate between P and Q, instead of focusing on the (often more trivial) question why the output is in the neighborhood of P or Q in the first place. The added value of contrastive explanations in interpreting model output is underlined by Jacovi et al. (2021), who proposed a method to produce contrastive explanations for classification models by modifying model behavior to only be based on contrastive reasoning. Moreover, the concept of contrastive explanations is particularly interesting for Picnic as well: forecasting analysts primarily base their prediction on the observed demand of historical promotions, hence the current manual process heavily relies on contrastive reasoning to compute a promotional forecast. The use of contrastive explanations is therefore a familiar and widely accepted approach within Picnic.

Finding an appropriate balance between direct, easy-to-understand post-hoc explainability on one side, but also an acceptable error rate for cold-start forecasts on the other side, remains a complicated topic to solve. A promising attempt to fill this gap is the research carried out by Aguilar-Palacios et al. (2020) on interpretable cold-start promotional demand forecasting. They proposed a method involving a gradient boosted decision tree (GBDT) algorithm that can accurately forecast demand for cold-start article promotions, while in addition providing interpretable contrastive explanations for each forecast. Their method is based on the idea of finding historical promotions with similar promotional features, also called "neighbors", and then predicting the difference in promotional demand between those neighbors and the upcoming promotion. As a first step, they train a GBDT to predict the difference in demand between a pair of two promotions based on their promotional features. Second, they applied a greedy k-nearest neighbors algorithm with weighted Euclidean distances to find the k most similar historical promotions to an upcoming test promotion. The weights used in this distance calculation are based on the feature importances from the trained GBDT. Lastly, the trained GBDT is used to generate the final forecast of the upcoming promotion by leveraging the data of the k nearest neighbors. Using the CatBoost algorithm as the GBDT in the contrastive regression framework showed to give the lowest Mean Absolute Percentage Error (MAPE) out of all GBDTs, and was only outperformed by direct regression using the Extremely Randomized Trees algorithm. One of the limitations of their contrastive regression framework is the fact that it is not able to detect upper outliers in a dataset of historical promotions (article promotions that had atypically high sales), and therefore tends to overforecast. Furthermore, the regressor is solely trained on the subset of historical promotions that are from the same category as the forecasted promotion, ignoring any inter-category relations.

This research aims to extend and improve the contrastive regression framework as introduced by Aguilar-Palacios et al. (2020), and implement it in an e-grocery setting to create direct business impact for Picnic. First, an additional outlier detection step prior to model training aims to exclude promotions with atypical sales by scaling for baseline sales and relative discount. Second, a new feature on article category is introduced and the scope of model training is extended to allow for leveraging inter-category information. Third, a different regression tree algorithm, based on the principle of random forests instead of gradient boosting, is incorporated in the contrastive regression framework with the goal of improving forecasting accuracy. Lastly, a heterogeneous distance measure is implemented to allow for mixed-type data in the feature space.

Chapter 3

Methodology

This chapter lays out all the methodological steps used in this research. It starts by explaining in more detail the contrastive regression model from Aguilar-Palacios et al. (2020) in Section 3.1. This model is the starting point for the proposed model extensions listed in Section 3.2. Next, Section 3.3 gives an overview of the model target and potential features available in the data. Section 3.4 then lists the candidate models proposed in this research and the procedure to select the best out of these candidates. Lastly, Section 3.5 describes how the selected contrastive regression model is evaluated and compared to baseline methods.

3.1 Contrastive Regressor base model

The starting point of the model proposed in this research is the contrastive regression model introduced by Aguilar-Palacios et al. (2020), hereafter called the "base model". Their model is based on the idea that similar promotions (with similar explanatory variables) generate similar demand. In the case of cold-start article promotions however, there is a lack of data on similar historical promotions from the same article. As a solution, the contrastive regression framework looks for historical promotions that are relatively similar to the upcoming cold-start promotion, but also takes into account the differences in explanatory variables that might still be present. This eventually leads to a cold-start promotional forecast that is accompanied with a contrastive explanation. An example is: suppose we plan an upcoming ice cream promotion in summer with 25% discount. We expect the demand to be lower than a historical ice cream promotion during summer with 50% discount, because discount has a positive effect on article demand. Furthermore, we expect the demand to be higher than a historical ice cream promotion during winter with 25% discount, because temperature has a negative effect on the demand for ice cream. The demand of the upcoming promotion is now defined in contrast to the demand of these two similar historical promotions, and is likely to be somewhere in between. To facilitate this, the GBDT algorithm is not trained on forecasting the demand of an upcoming promotion directly, but it instead forecasts the difference in demand between the upcoming promotion and similar historical promotions.

A general understanding of the five steps in the base model is crucial to understand the rationale behind the extensions proposed in this paper, hence a brief overview of these steps is given in this section. Figure 3.1 shows a visual representation of the base model. For a more

detailed explanation, please consult Section 2 of the paper by Aguilar-Palacios et al. (2020).

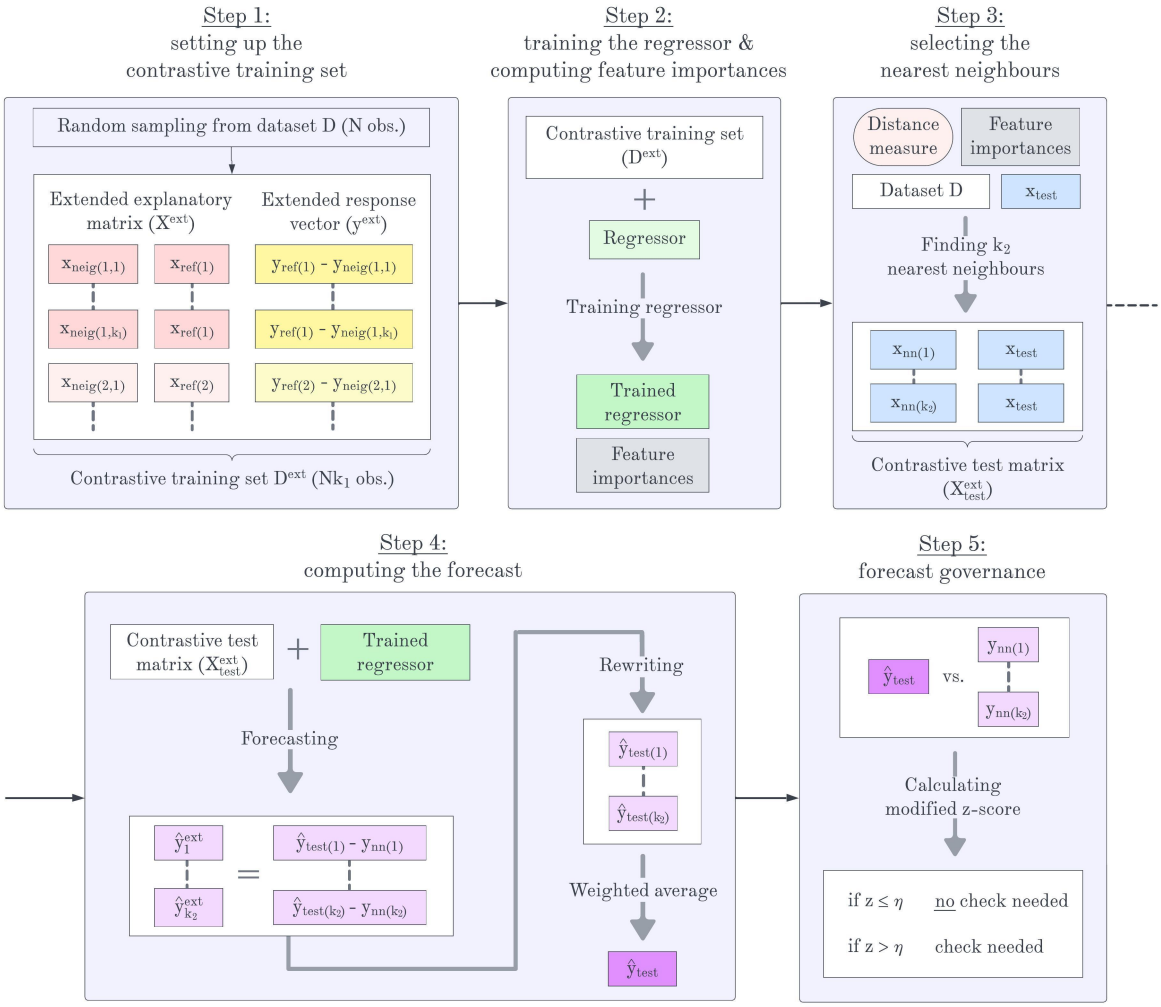


Figure 3.1: Step 1 to 5 of the base model

Step 1: setting up the contrastive training set

The first step consists of setting up a contrastive training set $\mathcal{D}^{\text{ext}} = (\mathbf{X}^{\text{ext}}, \mathbf{y}^{\text{ext}})$. The subscript "ext" in the notation refers to the fact that this contrastive training set is an extended version of the original dataset $\mathcal{D}$. The observations in the contrastive training set are computed as follows: we start with a dataset $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ with N observations, where each observation is a historical promotion with q features in $\mathbf{X}$ and one response variable in $\mathbf{y}$. Now, we select the first observation from $\mathcal{D}$ as the first reference promotion, denoted by $(\mathbf{x}_{\text{ref}(1)}, y_{\text{ref}(1)})$. Next, we randomly draw another observation from $\mathcal{D}$ and call it the first "neighbor" of the first reference promotion, denoted by $(\mathbf{x}_{\text{neig}(1,1)}, y_{\text{neig}(1,1)})$. This random drawing process is done under the restriction that this neighbor promotion occurred prior to the reference promotion to preserve the time-structure of the data. Now, the dependent variables of the first observation in the contrastive training set consist of the promotional features of the reference as well as of the neighbor, resulting in $\mathbf{x}_1^{\text{ext}} = [\mathbf{x}_{\text{neig}(1,1)}, \mathbf{x}_{\text{ref}(1)}]$ with length $2q$. The response variable of the first observation in the contrastive training set is the difference between the demand of the reference promotion and the corresponding neighbor, resulting in $y_1^{\text{ext}} = y_{\text{ref}(1)} - y_{\text{neig}(1,1)}$. For the first reference promotion,

This drawing process is repeated k_1 times to find k_1 random neighbors, yielding the first k_1 observations of the contrastive training set denoted by $\{(\mathbf{x}_1^{\text{ext}}, y_1^{\text{ext}}), ..., (\mathbf{x}_{k_1}^{\text{ext}}, y_{k_1}^{\text{ext}})\}$. The above steps are repeated N times, where each of the promotions in $\mathcal{D}$ is used as a reference promotion, eventually leading to the contrastive training set $\mathcal{D}^{\text{ext}}$ with Nk_1 observations. Note that in the contrastive training set, the response variable (y_i^{ext} for $i = 1, ..., Nk_1$) does not represent article demand directly, but the difference in article demand between a reference promotion and one of its randomly selected neighbors.

Step 2: training the regressor and computing feature importances

As a second step, a Gradient Boosting Decision Tree (GBDT) algorithm is trained on the contrastive training set. In other words, we learn it to predict difference in demand between two promotions ($\mathbf{y}^{\text{ext}}$) based on their explanatory variables ($\mathbf{X}^{\text{ext}}$). After training the GBDT algorithm, the feature importance vector $\mathbf{v} = [\mathbf{v^{neig}}, \mathbf{v^{ref}}]$ is calculated. This vector has length $2q$ and contains the importance scores for the features belonging to the neighbor and reference, respectively. For the GBDT algorithms in this research, the feature importances denote how much on average the prediction changes if the feature value changes. They are therefore calculated individually for each feature, and then normalized such that all feature importances together add up to 100. The feature importance vector forms the basis of the weights that are used later in the nearest neighbor search. To enhance direct interpretability and have one importance score per explanatory variable, the separate vectors are added up when presenting the feature importances to the user. This results in the more compact aggregated importance vector $\mathbf{v}' = \mathbf{v^{neig}} + \mathbf{v^{ref}}$ with length q, which will also represent the weights later on.

Step 3: selecting the nearest neighbors

The third step evolves around selecting the nearest neighbors (most similar historical promotions) for an upcoming test promotion ($\mathbf{x}_{\text{test}}, y_{\text{test}}$). Note that y_{test} is the final target we want to forecast, hence this value is unknown. For this test promotion, the contrastive test matrix $\mathbf{X}_{\text{test}}^{\text{ext}}$ needs to be computed by again finding neighbors. This time though, these neighbors are not selected randomly, but chosen to be the k_2 observations from the original dataset of historical promotions $\mathcal{D}$ that are closest to the test promotion $\mathbf{x}_{\text{test}}$. The base model in Aguilar-Palacios et al. (2020) applies a weighted Euclidean distance measure to calculate the distance between two promotions, where the weights are based on the feature importances $\mathbf{v}$ from the trained GDBT. Finally, the features of the test promotion together with the features of the k_2 nearest neighbors are used to arrange the contrastive test matrix $\mathbf{X}_{\text{test}}^{\text{ext}}$.

Note that we can distinguish between a symmetrical and an asymmetrical approach for leveraging the feature importance vector $\mathbf{v} = [\mathbf{v^{neig}}, \mathbf{v^{ref}}]$ in the weighted distance calculation. The symmetrical approach scales the features of the test promotion and the potential neighbor both with the aggregated importance vector $\mathbf{v}'$. The asymmetrical approach scales the features of the test promotion with $\mathbf{v^{ref}}$, and the features of the potential neighbor with $\mathbf{v^{neig}}$. This research applies the symmetrical approach, as we choose the weights of the reference and neighbor to affect the distance equally regardless of direction (i.e., regardless of which promotion is the reference, and which is the neighbour).

Step 4: computing the forecast

As a fourth step, each of the k_2 rows in $\mathbf{X}_{\text{test}}^{\text{ext}}$ is used seperately as input to the trained GBDT from step 2 to forecast one of the elements in $\hat{\mathbf{y}}_{\text{test}}^{\text{ext}} = [\hat{y}_1^{\text{ext}}, ..., \hat{y}_{k_2}^{\text{ext}}]$. These elements in $\hat{\mathbf{y}}_{\text{test}}^{\text{ext}}$ can be viewed as the forecasted differences in demand between the test promotion and each of the k_2 nearest neighbors, i.e. $\hat{y}_j^{\text{ext}} = \hat{y}_{\text{test(j)}} - y_{\text{nn(j)}}$ for $j = 1, ..., k_2$. Note that the actual demand of the nearest neighbor $y_{\text{nn(j)}}$ is known, but the demand of the test promotion $y_{\text{test(j)}}$ obviously is not. To transform this back to a forecast of the demand of the test promotion, the forecasted difference is added to the actual demand of the nearest neighbor. In other words, we rewrite to $\hat{y}_{\text{test(j)}} = y_{\text{nn(j)}} + \hat{y}_j^{\text{ext}}$ for $j = 1, ..., k_2$, and we gather the resulting forecasts in the vector $\hat{\mathbf{y}} = [\hat{y}_{\text{test}(1)}, ..., \hat{y}_{\text{test}(k_2)}]$. This vector now contains k_2 demand forecasts for the same test promotion, but all using a different nearest neighbor as reference point. Lastly, the final demand forecast of the test promotion $\hat{y}_{\text{test}}$ is computed by taking the weighted average of the k_2 different forecasts in $\hat{\mathbf{y}}$, where the inverse of the distances to each nearest neighbor are used as weights. Hence, we get for the final forecast

$$\hat{y}_{\text{test}} = \frac{\mathbf{w}^\top \hat{\mathbf{y}}}{\mathbf{w}^\top \mathbf{1}},$$

where $\mathbf{1}$ is a $(1 \times k_2)$ vector of ones, and $\mathbf{w} = [\frac{1}{d_1}, ..., \frac{1}{d_{k_2}}]$ with d_j the distance between the test promotion and the j-th nearest neighbor.

Step 5: forecast governance

The fifth and final step concerns verifying whether the demand forecast $\hat{y}_{\text{test}}$ is reasonable given the demand of similar historical promotions. To do so, a modified z-score of the forecast is computed with respect to the actual demand of the k nearest neighbors, given by

$$z = \frac{0.6745|\hat{y}_{\text{test}} - median(\mathbf{y}_{\text{nn}})|}{median(|\mathbf{y}_{\text{nn}} - median(\mathbf{y}_{\text{nn}})|)},$$

where $\mathbf{y}_{\text{nn}} = [y_{\text{nn}(1)}, ..., y_{\text{nn}(k_2)}]$ is the vector containing the actual demand of the nearest neighbors. All forecasts with a modified z-score above a threshold η are flagged as unreliable, and should be reviewed and adjusted if necessary. This research uses $\eta = 2.5$, following the work by Aguilar-Palacios et al. (2020) stating that this yields a relatively conservative governance check.

3.2 Proposed model extensions

This section outlines the four extensions applied to the base model from Section 3.1 that are proposed in this research: an outlier detection method using baseline demand, an inter-category training set-up, a decision tree algorithm with random selection of node splitting values, and a heterogeneous distance measure for mixed-type data.

3.2.1 Outlier detection prior to model training

The base model is able to detect and remove a part of the lower outliers from the training set by checking for promotions that were only active in a small number of stores or for a shorter

period of time. However, it does not detect promotions that are deemed normal in the feature space, but had an unusual amount of sales due to external factors that are not reflected in the features. Failing to exclude these outliers from the training set can lead the model to learning wrong relationships, resulting in higher risk of over- or underforecasting.

As an extension to the base model, this research introduces an additional step of outlier detection and removal applied to the original dataset $\mathcal{D}$ prior to step 1 of the base model (setting up the contrastive training set). The goal of this additional step is to have a standardized, objective way of excluding both the lower and upper outlier promotions from the set of historical promotions. To construct such a method, we first need to define when a historical promotion is actually considered to have an unusual amount of sales. In our proposed outlier detection method, this definition is computed using information on the baseline sales and relative discount associated with the promotion. The method has two sequential steps, which are detailed below:

- **Step 1:** The first step aims at excluding promotions where the resulting sales was lower than the baseline sales of the article, and where we assume this was caused by external factors that are not captured by the model. To do so, we define the sales uplift associated with a promotion as

$$\text{sales uplift} = \frac{\text{promotional sales}}{\text{baseline sales}}.$$

  As a promotion is expected to have an incremental effect on sales, promotions with an uplift below 1 are unreliable by definition. Reasons for an uplift below 1 could for example be supply chain issues, technical issues in the store app, or unusually high baseline sales in the period prior to the promotion. As we do not want the model to learn from these observations, they are are called lower outliers and excluded from the training set first.

- **Step 2:** The second step aims at excluding promotions that are deemed normal in the feature space, but had unusually little or many sales. To define a promotion with an unusual amount of sales among a set of promotions, we introduce a concept called the "discount-normalized lift" (DNL), which for a certain promotion is defined as

$$\text{discount-normalized lift (DNL)} = \frac{\text{sales uplift}}{\text{relative discount}}.$$

  To illustrate why this concept helps in detecting outlier promotions, please consider the simplified example in Table 3.1. All promotions A to H can be considered normal in the part of the feature space that is span by relative discount and sale uplift, as for both features we do not observe any extreme values. However, we do observe pairs of relative discount and sales uplift that can be considered extreme when observed together, and the DNL enables us to detect these. Promotion G has an unusually high sales uplift given the low relative discount. This could for example be caused by an unanticipated newspaper item about Picnic boosting sales during the promotion, or supply chain issues for substitute articles. On the other hand, promotion H has an unusually low sales uplift given the high relative discount. Possible reasons for this are similar to those mentioned earlier in step 1.

Promo ID	Relative discount	Sales uplift	Discount-normalized lift
A	0.20	4	20
B	0.25	6	24
C	0.50	11	22
D	0.15	3	20
E	0.20	5	25
F	0.50	9	18
G	0.20	11	**55**
H	0.50	3	**6**

Table 3.1: Example of outlier detection using discount-normalized lift (DNL).

Now that we have explained how promotions with an unusual amount of sales are found using the DNL, we need to define upper and lower thresholds for the DNL that determine whether a promotion is actually an outlier. To define these thresholds, we apply a method based on the adjusted boxplot introduced by Hubert and Vandervieren (2008). They propose an improved calculation of upper and lower outlier thresholds for skewed data based on the distribution quartiles, interquartile range (IQR) and the medcouple (MC), a univariate measure of skewness. These outlier thresholds account for possible skewness in the distribution of the DNL, which is particularly relevant in this research since the sales uplift (and as a consequence also the DNL) in the data are right-skewed (see Section 4.5). For $\text{MC} \geq 0$, the thresholds are

$$[Q_1 - k \cdot \mathrm{e}^{-4\text{MC}} \cdot \text{IQR}; Q_3 + k \cdot \mathrm{e}^{3\text{MC}} \cdot \text{IQR}],$$

and for $\text{MC} < 0$, the thresholds are

$$[Q_1 - k \cdot \mathrm{e}^{-3\text{MC}} \cdot \text{IQR}; Q_3 + k \cdot \mathrm{e}^{4\text{MC}} \cdot \text{IQR}],$$

where Q_i is the i-th quartile of the data, MC is the medcouple, $IQR = Q_3 - Q_1$ is the interquartile range, and k is a nonnegative constant determining the width of the interval. Our outlier detection method applies $k = 3$, following the research by John Tukey stating that datapoints beyond these thresholds are considered "far out", yielding a conservative detection method (Tukey (1977)). Lastly, it is important to take into account the large differences in sales uplift between different subgroups of articles: while promotions for frequently bought fresh items such as fruit and vegetables tend to have a lower sales uplift of 2 to 5, the sales uplift for ambient items such as deodorant and laundry detergent can easily rise above 10. To account for this, we define the above mentioned interval for each subcategory separately, after which all article promotions from that subcategory with a DNL outside this interval is marked as an outlier. These outliers will then be excluded from the training set, concluding our outlier detection method.

Note that excluded promotions are always examined to learn potentially interesting patterns that suggest the model can be improved (e.g., when one specific article within a subcategory always has unusual promotional sales, or always suffers from supply chain issues).

3.2.2 Inter-category training set-up

The base model is designed for "intra-category" application: it is trained on historical promotions from only one article category, and can be used to forecast promotions from that same category. As a consequence, the pool from which the model can select its nearest neighbors (dataset $\mathcal{D}$ in step 1 of the base model) contains only historical promotions from a single article category, and promotions from other categories can not be used to compute the forecast. This also means that a separate model needs to be trained for each category. This poses two potential problems: first, it could be the case that the upcoming test promotion is actually most similar to historical promotions from outside its own category (e.g. when the specific combination of selling price and relative discount associated with the test promotion is rarely observed in its own category, and more common in other categories). By restricting to intra-category training, this inter-category information can not be utilized. Second, the set of historical promotions can be small or empty for categories that are new or have little promotions in general. When the forecasted test promotion is from a category that suffers from this data scarcity, it could be useful for the model to be able to resort to data from other categories.

As an extension to the base model, this research introduces an "inter-category" set-up. More concretely, this means that the dataset $\mathcal{D}$ in step 1 of the base model contains article promotions from all categories, and that one overarching model is trained that can be used to forecast promotions from any category. The hypothesis is that intra-category information is still crucial for the model (as was also argued by Ma et al. (2016)) and that the majority of test promotions has the nearest neighbors in its own category, so we still want the model to somehow recognize when two historical promotions are from the same category. This is facilitated by adding article category as a feature to the model, hereby allowing this variable to be an explanatory factor both in the internal regression tree and in the nearest neighbor search.

3.2.3 Random forest-based decision tree algorithm

An important component of the base model is the gradient boosted decision tree (GBDT) algorithm that is trained to perform regression on the contrastive training set $\mathcal{D}^{\mathbf{ext}}$ in step 2, and then used to forecast demand of an upcoming test promotion in step 4. Aguilar-Palacios et al. (2020) evaluated performance of the contrastive regressor with different GBDT algorithms, and concluded that the CatBoost-based contrastive regressor had superior forecasting performance. CatBoost (**Cat**egorical **Boost**ing) is a gradient boosting algorithm released by Yandex in 2017 that gained popularity due to its built-in categorical feature handling and high performance in both classification and regression problems (Prokhorenkova et al. (2017)). It distinguishes itself from other boosting algorithms such as XGBoost and LightGBM through three key innovations. First, the trees in CatBoost are balanced due to symmetric node splitting, meaning that the algorithm finds the best combination of feature & split value for a certain depth level, and than splits all nodes from that level using this combination. This symmetrical structure, also called "oblivious trees", both increases computational efficiency and decreases the risk of overfitting. Second, the algorithm applies internal target encoding, which uses the category-level averages and probability distribution of the target to encode categorical features. As a result, the inherent order of categorical features is preserved, and we have an effective training method that does not

need one-hot encoding. Third, CatBoost combines this target encoding with a concept called ordered boosting to prevent target leakage. Target leakage occurs because traditional target encoding is applied using solely data from the training set. This can lead to a prediction shift, because the conditional distribution of the target derived from the training set can be different from what is observed in the test set. To overcome this, the training samples are randomly permuted and an artificial ordering in the data is created, after which target encoding is applied for each permutation seperately. This leads to a different encoding for each permutation (and hence for each boosting iteration), preventing the tree from overfitting on the target encoding.

Despite the promising performance of the contrastive regressor based on CatBoost, which even outperformed direct regression using CatBoost, Aguilar-Palacios et al. (2020) found that contrastive regression was still outperformed by direct regression using the Extremely Randomized Trees (ERT) algorithm. Note that "direct regression" here means that an algorithm is directly trained on dataset $\mathcal{D}$, hence it regresses promotional demand (y) on promotional features ($\mathbf{x}$). This in contrast to the contrastive regressor, where the algorithm is trained on the contrastive training set $\mathcal{D}^{\mathbf{ext}}$ and thus regresses difference in promotional demand on promotional features of both reference and neighbor.

Following the promising results of direct regression using ERT, this research explores the use of ERT within the contrastive framework as a possible improvement to the base model. In contrast to the gradient boosting-based decision tree algorithms considered by Aguilar-Palacios et al. (2020), ERT is a random forest-based algorithm. It was introduced by Geurts et al. (2006) and is based on the same principles as traditional random forests, but with two key differences: first, ERT does not apply resampling of observations when building the tree (i.e. it does not perform bagging) but instead samples from the entire dataset. Second, ERT does not directly select the best split out of a random subset of predictors, but it makes a small number of randomly chosen cuts for each predictor and then selects the best split from these cuts. The ensemble produced by ERT therefore yields trees that are less correlated, resulting in lower variance with only a relatively small increase in bias (Geurts et al., 2006). For the ERT algorithm in this research paper, we use the ExtraTrees implementation from the Scikit-learn library (Pedregosa et al., 2011), with corresponding feature importances denoted as the average accumulation of the impurity decrease within each fitted tree. As the ERT algorithm does not accept non-numerical variables, the categorical variables in the feature space need to be numerically encoded prior to training. In this research, the CatBoost encoder from the Scikit-learn library is used to solve this problem. This encoding algorithm applies the same principles of target encoding and ordered boosting as described earlier.

3.2.4 Heterogeneous distance measure

The base model uses the weighted Euclidean distance measure to calculate the distance between two promotions, with the feature importance scores as weights. This distance measure can only be used when the data is purely numerical or binary, but is not applicable for mixed-type data that also contains categorical variables, date variables, or text.

As an adjustment to the base model, this research implements the concept of Gower's similarity to define the distance between two promotions (Gower, 1971). This hybrid similarity

metric enables to measure similarity between two observations that consist of multiple data types. Just as in the base model, the dimensions are weighted using the feature importance scores from the regression tree algorithm. The resulting weighted Gower's distance between two promotions x_1 and x_2 is defined as:

$$\begin{aligned} D(x_1, x_2) &= 1 - S(x_1, x_2) \\ &= 1 - \left(\frac{1}{p} \sum_{j=1}^{p} w_j \cdot s_j(x_1, x_2) \right) \end{aligned}$$

where $S(x_1, x_2) = \frac{1}{p} \sum_{j=1}^{p} w_j \cdot s_j(x_1, x_2)$ is Gower's similarity between two promotions, p is the number of features, w_j is the weight of feature j, and $s_j(x_1, x_2)$ is the partial similarity function for feature j. To correctly handle mixed-type data, Gower's similarity applies a different partial similarity function for each data type. In addition to the three partial similarity functions for numerical, ordinal, and nominal variables that were originally introduced by Gower, this research uses a fourth similarity function for cyclical variables. Examples of cyclical variables are month of the year, season, or financial quarter. The four partial similarity functions are:

- **For numerical variables:** the range-normalized Manhattan distance is used, which gives:

  $$s_j(x_1, x_2) = 1 - \frac{|x_{1j} - x_{2j}|}{R_j}$$

  where x_{1j} and x_{2j} are the values of variable j for promotion x_1 and x_2, respectively, and R_j is the range of the values for variable j.

- **For ordinal categorical variable:** the ranked range-normalized Manhattan distance is used. This means that the categories of the variable are first transformed to integers using ordinal encoding, after which the range-normalized Manhattan distance is calculated. The partial similarity function then takes the same form as for numerical variables.

- **For nominal categorical variables:** an indicator function is used, which gives:

  $$s_j(x_1, x_2) = \begin{cases} 1 & \text{if } x_{1j} = x_{2j} \\ 0 & \text{if } x_{1j} \neq x_{2j} \end{cases}$$

  where x_{1j} and x_{2j} are the values of variable j for promotion x_1 and x_2, respectively.

- **For cyclical categorical variables:** a cyclical variant of the range-normalized Manhatten distance is used. First, the values are transformed to integers using ordinal encoding (e.g. "April" becomes 4, and "December" becomes 12). Next, the partial similarity is calculated as:

  $$s_j(x_1, x_2) = 1 - \frac{\min(|x_{1j} - x_{2j}|, T_j - |x_{1j} - x_{2j}|}{T_j/2}$$

  where x_{1j} and x_{2j} are the encoded values of variable j for promotion x_1 and x_2, respectively, and T_j is the period of the cyclical variable j (e.g., for month of the year we have $T_j = 12$). This function ensures that the cyclical nature is preserved, and that the distance between for example April (4) and December (12) is seen as 4 months instead of 8.

3.3 Model target and features

This section introduces the model target and available features considered in this research.

3.3.1 Target

The target of the base model is recommended to be (a compound metric of) article-level promotional demand. In this research, we use a compound demand metric that is generally used at Picnic called Article Delivery Rate (ADR). ADR is an article-level metric that indicates how many times an article is ordered on average at Picnic. It is defined as the average article count per customer delivery and is therefore independent of the total number of customer deliveries, making it a suitable proxy for relative article demand. For example: if on average 1 out of 20 deliveries contains a cucumber, the ADR for cucumber is $\frac{1}{20} = 0.05$. In this research we specifically focus on weekly ADR, which is the ratio of total weekly article sales over total weekly delivery count, hence it can be seen as the average ADR during a certain week. The reason for choosing ADR is because the number of deliveries at Picnic can vary strongly between weeks. Picnic predicts this number of weekly deliveries using a seperate forecasting model, and its output has proven to be reliable over the last years. To get a final prediction in terms of number of articles, ADR is multiplied by the number of deliveries. To ensure that our model actually captures the effect of promotions on relative article demand and not the fluctuations in number of deliveries, ADR is used. As an example: if there is a group of 1000 active customers that buys a total of 30 cucumbers in a certain week, the average ADR of cucumber in that week is 0.03. If some time later the group of active customers has increased to 1500 but the relative article demand for cucumber hasn't changed, the total sales would increase to 45 but the ADR would remain 0.30 accordingly. Note that we do not necessarily assume that the weekly set of article promotions does not affect the total number of weekly deliveries; if in a certain week the total promotional offer is exceptionally attractive or not attractive to customers, we might expect the number of deliveries to grow or shrink a little bit accordingly. We do however think that this effect is negligible and assume that the attractiveness of the weekly set of article promotions stays constant over time, thereby also keeping the effect of promotions on the number of weekly deliveries constant. This assumption is plausible given that Picnic carefully configures the promotional offer of each week, making sure that each week contains a similar, balanced set of article promotions. Any deviations in the number of deliveries are then captured by the seperate forecasting model mentioned earlier.

What is important to mention is that ADR, which is directly derived from article sales, does not fully capture the article demand: when an article runs out of stock at a certain fulfilment center during a promotion and can not be sold there anymore, the real article demand is higher than the actual sales. As a result, using ADR (based on actual sales) as the target variable when training the model would lead to underforecasting of the real article demand. To correct the actual promotional sales for this local out-of-stock periods, we use data on customers that saw an article "unavailable" after adding it to their basked in the app to estimate the "lost sales" due to unavailability. As a result, the target variable of our model is this ADR quantity

corrected for unavailability, which we call "clean ADR". Clean ADR is calculated as

$$\text{clean ADR} = \frac{S_{actual} + 0.75 \cdot C_{missed} \cdot a}{D_{weekly}},$$

where S_{actual} is the actual sales, C_{missed} is the amount of customers that saw the article unavailable and did not order it later that week, a is the average number of articles a customer buys per order, and D_{weekly} is the total number of deliveries that week. The conversion factor of 0.75 means that 75% of the customers that add an article to their online basket end up actually ordering that article, and this factor is based on analysis at Picnic.

3.3.2 Features

The explanatory variables (i.e. available model features) considered in this research are the following (ordered by feature type, and followed by the rationale behind including this variable):

- Numerical:
  - **Baseline clean ADR** (i.e. average clean ADR of preceding three non-promotion weeks): we expect that a higher baseline clean ADR leads to higher clean ADR, as promotional demand by definition is an upscaled version of baseline demand.
  - **Regular selling price** (i.e. article selling price without discount): we expect that a higher regular selling price leads to higher clean ADR, as a discount for an article that is more expensive is more appealing to customers.
  - **Relative discount** (e.g. "0.20" for a 20% discount): we expect that higher discount leads to higher clean ADR, as more discount is more appealing to customers.
  - **Promotion group size** (i.e. the number of article promotions in the promotion group, e.g. "4" for a paprika chips promotion that is part of a promotion group with three other flavours): we expect that a larger promotion group size leads to lower clean ADR, as customers can then choose between a larger set of interchangeable article promotions (of which they will likely only choose one or two). Hence, we can think of it as if a larger promotion group size spreads out the total number of customers that buy from this promotion group over more articles, decreasing the promotional sales uplift of each individual article.
  - **Article content** (number of items in one article, e.g. "6" for a sixpack of drink cans): we expect that a higher article content leads to higher clean ADR, as customers get more value for their money if the article promotion contains more items.
  - **Freshness days** (i.e. the number of days for which freshness of the article is guaranteed, e.g. "7" for a milk carton with a shelf life of one week): we expect that higher freshness days lead to higher clean ADR, as customers are more likely to be encouraged to buy an article promotion when it has a longer shelf life (these articles are more likely to be stockpiled by customers).
  - **Multibuy quantity** (i.e. the number of articles a customer should buy to qualify for the promotion, e.g. "1" for a 50% discount promotion, and "2" for a 1+1 free

promotion): we expect that a higher multibuy quantity leads to higher clean ADR, as we expect that the decreased number of customers that buy the article promotion (because some customers are put off by a higher multibuy quantity) does not outweigh the increased number of articles bought per customer.

- Binary:
  - **Superdeal** (i.e. whether the promotion was highlighted as a superdeal in the store app): we expect that a superdeal has higher clean ADR, as the visibility towards customers for this promotion is higher than that of promotions that are not a superdeal.
  - **Freshness guarantee** (i.e. whether the grocery retailer guarantees freshness of the product at delivery): we expect that a promotion with a freshness guarantee has lower clean ADR, applying the same reasoning as for the feature "freshness days".
- Categorical:
  - **Promotion mechanism** (e.g. "x for y% discount" or "x plus y free"): we expect each promotion mechanism to have a different effect on customer demand. For example, a "25% discount" promotion is likely to have a different effect on customer demand than a "2nd for 50% discount" promotion.
  - **Article category** (e.g. "Pasta, Rice & International"): we expect each article category to have a different underlying relationship between features and article demand.
- Time-date:
  - **Promotion month** (month in which article promotion was active): we expect each month to have a different effect on promotional demand (e.g., ice cream promotions might have a larger uplift in summer)

Other variables that are generally known to affect promotional demand could not be included because this data is not yet available at Picnic. These include promotions from competitors, additional marketing effort carried out by Picnic (e.g. weekly email to customers or TV commercial highlighting certain promotions), and a more detailed indication of the visibility of the promotion in the store app (e.g. whether the promotion was shown on the home page, and how high it was ranked on the promotion page or search menu). Note that only one time-date variable is considered, because multicollinearity between features is undesirable for the contrastive regression framework. The reason why is discussed in more detail in Subsection 4.5.3.

3.4 Model selection

This section describes the methodology used to select the best performing contrastive regression model out of the ones proposed in this research. First, the feature selection procedure is explained. Next, an overview of all candidate models is given, together with a detailed outline of the training & validation scheme used for model selection. Furthermore, the error metric that is used to evaluate all models is introduced.

3.4.1 Candidate models

As stated in Section 3.2, the primary goal of this research is to extend and potentially improve the contrastive regression model as introduced by Aguilar-Palacios et al. (2020), and compare its performance in an e-grocery setting with baseline methods. As a first step towards this goal, we should determine which (if any) of the proposed extensions improve the performance of the base model and thus should be included in the final model that will be benchmarked against baseline methods. To achieve this, four models will be trained, optimized, and evaluated to find the winning model. The four candidates are based on the contrastive regression framework, in combination with the following decision tree algorithms and training set-up:

- **CatBoost** algorithm and **intra-category** training (denoted 'CR-CBintra')
- **ExtraTrees** algorithm and **intra-category** training (denoted 'CR-ERTintra')
- **CatBoost** algorithm and **inter-category** training (denoted 'CR-CBinter')
- **ExtraTrees** algorithm and **inter-category** training (denoted 'CR-ERTinter')

A comparison of the above four candidates determines whether (and if so, which of) the model extensions improve forecasting performance. Note that all candidate models implement the outlier detection method proposed in Subsection 3.2.1 and the heterogeneous distance metric proposed in Subsection 3.2.4. Furthermore, all candidates use five neighbors when computing the contrastive training set and five neighbors when forecasting the final demand, hence $k_1 = k_2 = 5$, following the results from Aguilar-Palacios et al. (2020). Please find a more detailed discussion on the choice for the number of neighbors in Section 8.1.

3.4.2 Selection procedure

To gain insight in the forecasting performance of the four candidate models, confidence intervals are computed for their generalization errors. Here, "generalization error" denotes the model error on unseen test data. These intervals provide a better understanding of the relative performances of the candidate models, because we not only compare the average error but also give a sense of its variance. The generalization error we aim to estimate in this section is denoted by ${}_{n_1}\mu$, which is defined as the error the model makes on unseen test data when trained on a training dataset of size n_1. To compute confidence intervals for ${}_{n_1}\mu$, we use the estimators for the generalization error and its variance that were introduced by Nadeau and Bengio (2003). The pseudocodes for the cross-validation schemes used to compute these estimators are given in Appendix B. The dataset used in these procedures is called D^{select} (size n), where n is the number of observations.

First, to estimate the generalization error we apply repeated random sub-sampling cross-validation (also known as Monte Carlo cross validation), which was first introduced by Picard and Cook (1984). We perform $J = 15$ rounds of cross-validation, each round with a random train-validate split ratio of 80:20. As a result, $n_1 = 0.8n$ is the number of observations in each training set and $n_2 = 0.2n$ is the number of observations in each validation set. For the j-th round, let D_j be the subset of n_1 training observations randomly sampled from dataset D_{select},

and let D_j^c denote the remaining n_2 validation observations. The resulting cross-validation estimate of the generalization error after 15 rounds is then given by

$$ {}^{n_2}_{n_1}\hat{\mu} = \frac{1}{15}\sum_{j=1}^{15}\hat{\mu}_j, $$

where $\hat{\mu}_j$ is the "average test error" belonging to the j-th round, denoted by

$$ \hat{\mu}_j = \frac{1}{n_2}\sum_{i\in D_j^c} L(j,i). $$

Here $L(j,i)$ is the error a model trained on training set D_j makes on an unseen observation i from validation set D_j^c. Note that ${}^{n_2}_{n_1}\hat{\mu}_J$ is an unbiased estimator of the generalization error ${}_{n_1}\mu$, given the assumption that the observations in D_{select} are independent and all follow the same underlying distribution. In this research, we assume that indeed the article promotions in D_{select} come from the same data generating process that independently creates promotions. It is known that in practice, each set of weekly promotions is likely to be tuned by the company, and their resulting sales are never fully independent as promotions for substitute and complement articles affect one another. However, supported by the facts that we have a large range of available promotions (on average 800-1000 are active at the same time) and inter-promotional dynamics are known to be less prominent at Picnic, we assume approximate independence.

Second, to estimate the variance of this generalization error estimator, we apply a slightly different cross-validation procedure. We start off by randomly splitting the dataset D_{select} into two datasets D_1 and D_1^c with split ratio 50:50. Then, for each of these two datasets, 15 rounds of repeated random sub-sampling cross-validation are performed using a train-validate split ratio of 60:40 (to ensure that the validation set is again of size n_2). This computes two independent estimates of the cross-validation error $\hat{\mu}_{(1)}$ and $\hat{\mu}^c_{(1)}$. Note that this time, these estimates of the generalization error are computed using a training dataset of size $n_1' = \frac{n}{2} - n_2 < n_1$, hence we actually compute ${}^{n_2}_{n_1'}\hat{\mu}$ instead of ${}^{n_2}_{n_1}\hat{\mu}$. These steps are repeated 5 times to get 5 pairs of cross-validation estimates of the generalization error. The estimated variance is now given by

$$ {}^{n_2}_{n_1'}\hat{\sigma}^2 = \frac{1}{10}\sum_{m=1}^{5}(\hat{\mu}_{(m)} - \hat{\mu}^c_{(m)})^2. $$

Nadeau and Bengio (2003) state that this estimator overestimates the variance of the generalization error estimator ${}^{n_2}_{n_1}\hat{\mu}$, because ${}^{n_2}_{n_1'}\sigma^2 \geq {}^{n_2}_{n_1}\sigma^2$. This statement is made using the conjecture that the decision function computed by our learning algorithm becomes less variable as the training set becomes larger. In this research in particular, this means that we assume that the variance of the final tree that results from training on a set of observations (and hence, the variance of the generalization error) decreases as we increase the number observations.

Eventually, for each of the four candidates the cross-validation schemes produce an estimation of the generalization error and its variance. These are then used to compute confidence intervals for the generalization error, leveraging the work by Nadeau and Bengio (2003) which states that the estimated generalization error follows an approximate normal distribution. Here the central

limit theorem is used to argue that the distribution of $\frac{n_2}{n_1}\hat{\mu}$ is approximately normal, because it is the mean over many (in this case $15n_2$) forecasting errors. The central limit theorem is valid under the assumption that these errors are independent, i.e. the errors from different cross validation rounds as well as within each cross-validation round should be independent. First, due to the random sampling of the test set in each round, errors from different rounds are expected to be independent. Second, within each round we also assume the errors to be independent as we propose that the observations in D_{select} come from a data generating process that independently creates promotions, following the same reasoning as mentioned earlier.

For each model, the confidence interval of the generalization error is calculated as:

$$[\hat{\mu} - c\sqrt{\hat{\sigma}^2}, \hat{\mu} + c\sqrt{\hat{\sigma}^2}]$$

where $\hat{\mu} = \frac{n_2}{n_1}\hat{\mu}$, $\hat{\sigma}^2 = \frac{n_2}{n_1'}\hat{\sigma}^2$, and $c = z_{1-\alpha/2}$ is the $100 \cdot (1-\alpha)$-th percentile of $N(0,1)$. In this research we compute 95% confidence intervals, hence $\alpha = 0.05$. The interpretation of the estimated mean and confidence interval of the generalization error of the four candidates will be as follows: in essence, the mean of the generalization error will be used to decide which model has the highest forecasting accuracy and should be selected for further analysis. In addition, the estimated confidence intervals serve solely as an indication of how stable this generalization error is (and hence, how stable the performance of the model is). As the variance of the generalization error is overestimated, this indication of stability is conservative (a desirable property when proposing new model extensions in literature). Note that we do not aim to draw statistical conclusions from the confidence intervals on which model significantly outperforms the others.

3.4.3 Hyperparameter tuning

In the cross-validation schemes described above, each individual round requires the training and tuning of a new model on a new subset of data. As mentioned already, in this research the hyperparameters of the model are tuned using Bayesian optimization. This form of optimization gradually builds a surrogate probability model of the objective function, and uses this model to select the most promising hyperparameters to evaluate in the true objective function in the next trial. For a more detailed explanation, please consult the paper by Snoek et al. (2012). In this research, the number of trials is set to 30, and in each trial the dataset is split into a training and validation set with ratio 80:20. The three hyperparameters to be tuned (with corresponding search ranges between parentheses) are number of iterations for CatBoost or number of estimators for ERT ([1, 1000]), learning rate ([0.001, 0.1]), and tree depth ([1, 8]).

3.4.4 Error metrics

The error metric that is used to train the models and assess their forecasting performance in the selection procedure s the Mean Absolute Error (MAE). This metric is defined as:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} \mid \hat{y}_i - y_i \mid$$

where N is the number of predictions, $\hat{y}_i$ are the predictions, and y_i are the observed values. This metric applies an absolute loss where the penalty is proportional to the absolute value of the error. As a result, MAE treats all errors (from small to large) equally and weighs them linearly proportional to their magnitude. It therefore is less sensitive to outliers compared to, for example, Mean Squared Error (MSE). MAE is very suitable for situations in which the contribution of each error to the overall assessment of model performance should be linearly proportional to its magnitude. This behaviour is particularly relevant in this research: each error leads to a certain amount of articles that gets over- or underforecasted at Picnic, and the primary goal for the business is to minimize the sum of absolute errors across the whole promotion assortment (i.e., minimize the sum of all customers that see an article promotion unavailable on one side, and all articles that potentially become waste on the other side). Because MAE is based on the sum of absolute errors, this metric is naturally more strongly influenced by forecasting errors for articles that have a higher sales level. This property is desirable at Picnic, as these errors also cause the most customer dissatisfaction or product waste.

In addition to the MAE, two relative error metrics are used to help interpret the size of the errors with respect to the actual observed clean ADR. These are the Weighted Absolute Percentage Error (WAPE) and the Weighted Percentage Error (WPE), which are defined as

$$\text{WAPE} = \frac{\sum_{i=1}^{N} |\hat{y}_i - y_i|}{\sum_{i=1}^{N} y_i},$$

$$\text{WPE} = \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)}{\sum_{i=1}^{N} y_i},$$

where N is the number of predictions, $\hat{y}_i$ are the predictions, and y_i are the observed values. The WAPE and WPE are the standard metrics used at Picnic to evaluatue the performance of article demand forecasting processes. The WAPE weights the sum of absolute errors of a set of forecasts by the sum of the observed values, and is therefore a normalized version of the MAE. As a result, it is a relative error measure where the contribution of each error to the overall assessment of model performance is linearly proportional to its magnitude (just as was the case for the MAE. The WPE is similar to the WAPE, but uses the sum of errors instead of the the sum of absolute errors in the numerator. As a result, this metric gives a clear view on whether the model tends to over- or underforecast article demand. Both WAPE and WPE allow Picnic to compare the performance of a forecasting model across different sets of forecasts with varying total demand (e.g., across different weeks, or different parts of the article assortment).

3.5 Final model evaluation

This section describes the methodology used to evaluate the final model (i.e. the winning candidate model from Section 3.4) on a test set of unseen promotions. This evaluation is twofold: first, the feature importance scores are interpreted to get a better insight in the factors that are deemed important for cold-start promotional demand forecasting. Second, the forecasting performance of the final model is compared to existing baseline methods to determine how the accuracy of the contrastive regression framework relates to more established forecasting models.

3.5.1 Models to be evaluated

This subsection lists the models that are considered in the final model evaluation:

1. **Winning contrastive regressor model:** the contrastive regression model that had best performance out of the four candidates in Section 3.4.

2. **Direct regression with intra-category training using CatBoost algorithm:** the CatBoost algorithm trained directly on the X (features) and y (clean ADR) of the original dataset $\mathcal{D}$ containing historical promotions. Hence, it directly forecasts the demand of an upcoming promotion by using its features as input. This as opposed to the contrastive regression model, where the regression tree is trained on the contrastive training set $\mathcal{D}^{\text{ext}}$ and used to forecast the difference in demand between the upcoming promotion and its k_2 nearest neighbors. For the direct regression model, a separate regression tree is trained for each article category (i.e. the intra-category training set-up is applied), as this was also done in Aguilar-Palacios et al. (2020) and we want to isolate the effect of inter-category training in the contrastive regression models.

3. **Direct regression with intra-category training using Extremely Randomized Trees algorithm:** the same procedure as described above, but with the Extremely Randomized Trees algorithm as regressor. Note that this is the only model that outperformed the contrastive regression model in Aguilar-Palacios et al. (2020).

4. **Weighted nearest neighbor regression with intra-category training:** a weighted Nearest neighbor Regression (NNR) algorithm that forecasts the demand of an upcoming article promotion in two steps. As a first step, it finds the k nearest neighbors (most similar historical promotions) to an upcoming article promotion using the same weighted Gower's distance metric that is also used in the contrastive regression model. The same dataset of historical promotions as used for the other models is also used here, with exactly the same features. Note that again the intra-category approach is applied, because this was done in Aguilar-Palacios et al. (2020) and we want to see the impact of inter-category training in contrastive regression. Hence, we look for neighbors only within the own article category and use weights specific for that category in the distance calculation. These weights are based on the feature importances from one of the earler mentioned direct regression models with intra-category training. The choice for model 2 (CatBoost) or model 3 (ERT) will depend on which of the two algorithms shows best performance in the contrastive regression framework. As a second step, it forecasts the demand of the upcoming article promotion by simply taking the weighted average of the demand of the k neighbors, where the weights are equal to the inverse of the distance between the upcoming promotion and each neighbor. Hence, the forecasted demand for article promotion i is denoted by

$$\hat{y}_i = \frac{\mathbf{w}^\top \mathbf{y}_{\text{nn}}}{\mathbf{w}^\top \mathbf{1}}$$

where $\mathbf{1}$ is a $(1 \times k)$ vector of ones, $\mathbf{w} = [\frac{1}{d_1}, ..., \frac{1}{d_k}]$ with d_j the distance between the test promotion and the j-th nearest neighbor, and $\mathbf{y}_{\text{nn}} = [y_{\text{nn(1)}}, ..., y_{\text{nn(k)}}]$ with $y_{\text{nn(j)}}$ the actual

ADR of the j-th nearest neighbor. As explained above, this distance is computed using the weighted Gower's distance metric with the feature importances from direct regression as weights. For a more detailed explanation of weighted nearest neighbor regression, please consult Altman (1992). Note that by definition, the forecasted demand will always be in between the minimum and maximum of the demands of the k neighbors. Therefore, we expect this model to perform well for article promotions that are more frequent and well-known, but worse for article promotions that are more "cold-start" and have less near neighbors. Hence, it can be seen as a useful compromise between contrastive regression on the one hand (which has explainable forecasts and focuses on good accuracy for cold-start promotions), and direct regression on the other hand (which has less explainable forecasts, but focuses more on good accuracy all-round).

5. **Naive model:** a naive model that produces forecasts in a simple and computationally efficient manner. For an upcoming article promotion from a certain article category, we take the average promotional uplift of all promotions in the training set belonging to that category and multiply this uplift with the baseline demand of the article. Hence, the forecasted demand for article promotion i belonging to article category c is given as

$$\hat{y}_i = \bar{u}_c \cdot b_i,$$

where $\bar{u}_c$ is the average uplift of all promotions in the training set from article category c, and b_i is the baseline demand of the article.

6. **Manual forecast:** a manual forecast that was created by an analyst specialized in promotional demand forecasting at Picnic. These forecasts are computed approximately five weeks prior to the upcoming promotion based on the features of the upcoming promotion, data on similar historical promotions, and business expertise. They were already computed in the past and are retrieved from an internal database retrospectively for this research. The manual forecast can be considered an educated guess of the demand for an upcoming promotion, and is currently the standard way of computing promotional demand forecasts at Picnic. Although the forecasting accuracy of this process is on an acceptable level, it is prone to human errors and very time-consuming because the forecasts have to be computed one-by-one and always involve subjective interpretation. Therefore, the main goal of the contrastive regressor proposed in this paper is to at least match, and potentially even improve, the forecasting accuracy of this manual process currently used at Picnic.

3.5.2 Evaluation procedure

Step 1: training the models that involve a learning algorithm

Before evaluating the performance of the models described above, the ones involving a supervised learning algorithm (model 1-3) need to be trained and tuned. The training set is the same dataset D_{select} that was used in Section 3.4 to select the best contrastive regressor model. Hyperparameters are again tuned using Bayesian optimization with 30 trials and a train-validate split ratio of 80:20, using the same hyperparameter ranges as were introduced in Subsection 3.4.3.

Step 2: evaluating the models on unseen data

The models are tested on an evaluation set D_{eval} containing unseen observations. For the contrastive regressor (model 1), the feature importance scores of the internal tree algorithm are interpreted to understand which factors are deemed important for cold-start promotional demand forecasting with the contrastive framework. Furthermore, the forecasting errors are evaluated to get an objective view on model accuracy. To test whether the forecasts of model 1 are statistically different from the forecasts of models 2-6 individually, Wilcoxon signed-rank tests are performed with a significance level of $\alpha = 0.05$. This test enables pairwise comparison of the forecasts and does not require the normality assumption thanks to the non-parametric nature of the test. To account for multiple testing, the significant level is corrected via the Bonferroni procedure, giving $\alpha^* = \frac{\alpha}{n_{tests}} = \frac{0.05}{n}$ where n is the number of tests. Lastly, the computational load of each model is assessed by tracking the runtime associated with one cycle of training & prediction.

3.5.3 Error metrics

The error metric used to train and tune the models is again the MAE. To assess their performance relative to the actual observed clean ADR, we again use WPE and WAPE. In addition to these three metrics, the coefficient of determination (R^2) is calculated. This metric denotes the proportion of the variance in the target (clean ADR) that is predicted by the features the model, and is calculated as

$$R^2 = 1 - \frac{\text{Sum of Squared Residuals (SSR)}}{\text{Total Sum of Squares (SST)}} = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2},$$

where N is the number of predictions, $\hat{y}_i$ are the predictions, $\bar{y}_i$ is the mean of the observed values, and y_i are the observed values. In other words, the R^2 checks to what extent the regression model outperforms a model that simply takes the mean of observed values as each prediction. Here, 0 denotes that the regression model is just as good as simply predicting the mean, and 1 denotes a perfect fit.

Chapter 4

Data

The data used in this research is provided by Picnic, an online supermarket delivering groceries in the Netherlands, Germany and France. The Dutch market is currently their main market with around 200,000 customers every week, but their customer base in Germany and France are expanding rapidly. At Picnic, customers place an order with a minimal value of €40,- via the dedicated Picnic mobile app by selecting items out of the 10,000+ items currently in assortment. After filling their basket, customers select a morning, afternoon or evening delivery slot on any weekday of choice, after which the groceries are delivered to their home free of charge. Any additional items can be added to this same order basket up to the evening before delivery, providing a flexible grocery shopping experience.

The data source from which all data is extracted is the Picnic Data Warehouse. This data warehouse continuously collects data from multiple sources, such as customer orders, deliveries, logistic operations, demand planning, financials, and many more. It provides a comprehensive and centralized source for Picnic employees to extract data from to use in analyses and reporting.

4.1 *ArticlePromotion* data class

The contrastive regression method proposed in this paper forecasts demand of upcoming article promotions based on demand of historical article promotions that share similar values for certain features. To facilitate the search for similar historical promotions, we introduce a data class *ArticlePromotion*. This class holds a variable for the observed value of the model target (subsection 3.3.1), each of the features (subsection 3.3.2), the corresponding manual forecast (model 5 from subsection 3.5.1), and the so-called "coldness" of a promotion. In this research, the coldness of an article promotion is the number of historical promotions that were done earlier for that particular article at the start of the promotion (i.e., how many times has this article been in promotion before, so how much useful historical data is available?). As an example: if an article has never been in promotion before and the forecast is considered completely cold-start, then this field is 0. However, if an article has already been in promotion three times earlier and hence some historical data is available, then this field is 3. The variable "coldness" enables us to evaluate the forecasting performance of a model for varying degrees of coldness to investigate the effect of increasing promotion data availability.

Each observation in our dataset is a historical article promotion characterized by a seperate

instance of the *ArticlePromotion* data class. An example of such an instance is:

$$\left\{\begin{array}{ll} \textit{Clean ADR} & = 0.039 \\ \textit{Baseline clean ADR} & = 0.017 \\ \textit{Regular selling price} & = 2.79 \\ \textit{Relative discount} & = 0.25 \\ \textit{Promotion group size} & = 14 \\ \textit{Article content} & = 1 \\ \textit{Freshness days} & = 50 \\ \textit{Multibuy quantity} & = 1 \\ \textit{Superdeal} & = \text{yes} \\ \textit{Freshness guarantee} & = \text{no} \\ \textit{Month} & = \text{August} \\ \textit{Promotion mechanism} & = \text{x\% discount} \\ \textit{Article category} & = \text{Breakfast \& Snacks} \\ \textit{Manual forecast} & = 0.042 \\ \textit{Coldness} & = 3 \end{array}\right\}$$

4.2 Filtering

The observations used in this research are extracted from the Picnic Data Warehouse. Several filters are applied to construct the full dataset used for model selection and evaluation. The list below gives an overview of these filters, which can be seen as the requirements an *ArticlePromotion* instance must meet to be included in the research:

1. **Market:** the promotion was active in the Dutch market. The rationale behind this is that the Dutch market is most matured and contains the highest data quality amongst Picnic's markets. As the market dynamics in Germany and France are different and data quality and completeness are lower, we exclude these observations.

2. **Time frame:** the promotion was active between 2023 week 25 and 2023 week 46. The starting week is the first week in which article promotions were sold via a dedicated promotion page in the Picnic app, hence denoting the first week in which the data is based on an app environment and promotional strategy as Picnic has it today. The ending week denotes the most recent sales week at the time of conducting this research.

3. **Promotion type:** the promotion is of type "Regular Weekly". This means that it concerns a regular weekly promotion that was planned to be active for seven days (Monday to Sunday). This is in contrast to promotions of type "Day deal", which were planned to be active for only one day, or of type "Continuous", which are always active.

4. **Active period:** the promotion was active for 5-7 days. Important to mention is that promotions of type "Regular Weekly" are not necessarily always active for seven days. On the one hand, it could be active for less than 7 days when for example a weekly promotion starts later than Monday due to delivery issues at the supplier, or ends earlier than Sunday due to the article going completely out-of-stock. On the other hand, it could be more than 7 days when for example Picnic decides to extend the promotion to prevent product waste. This range is selected in accordance with the business operations at Picnic: promotions

with an active period outside this range are considered not representative of a "normal" weekly promotion, hence the sales can not be reliably corrected. The correction procedure for promotions with an active period within this range is outlined in Section 4.3

4.3 Correction

As detailed in Subsection 3.3.1, in this research the target variable of the model (clean ADR) is aggregated on a weekly level. This means that we specifically focus on the ratio of weekly article sales over weekly delivery count, which can be seen as the average ADR of a weekly article promotion. For promotions that were active for less than 7 days, the corresponding sales need to be scaled accordingly to represent a full-week promotion. Hence, the following correction is applied to all *ArticlePromotion* instances:

$$\text{weekly clean ADR} = \text{observed clean ADR} \cdot \frac{7}{d_{active}}$$

where $d_{active} \in \{5, 6, 7\}$ is the number of days the promotion was active.

Furthermore, for promotions concerning ambient articles with an unspecified shelf life, the variable "Freshness days" is automatically set to 50.

4.4 Dataset split for model selection & evaluation

The dataset that remains after extracting all *ArticlePromotion* instances from the Picnic Data Warehouse and applying the filters from Section 4.2 needs to be split in two parts: a selection set that is used for model selection (Section 3.4), and an evaluation set that is used for final model evaluation (Section 3.5). In this research, the data is split chronologically, where the selection set contains all promotions between 2023 week 25 and week T, and the evaluation set contains all promotions between 2023 week $T + 1$ and week 46. This chronological split ensures that a cold-start promotion in the evaluation set is in fact cold-start (i.e. it prevents that promotion data from the same article from a later period in time leaks into the selection set and diminishes the meaning of the time-related variable "Coldness" in the evaluation set). Week T will be chosen such that the split ratio is approximately 70:30 for the selection and evaluation set, respectively, ensuring sufficient unseen test observations are available for the final model evaluation.

4.5 Preliminary data analysis

This subsection contains a preliminary data analyses performed prior to building the regression model. The analyses aim at providing a better understanding of the model target and available feature data. This understanding is used to substantiate modeling choices and feature selection, and helps to interpret the results from model selection and evaluation later on.

4.5.1 Final dataset

Figure 4.1 visualizes the result of the sequential process of data collection, filtering, outlier detection, and splitting. Here, N denotes the number of observations in the final dataset $\mathcal{D}$.

Collecting all *ArticlePromotion* instances and applying the filters from Section 4.2 leads to a full dataset of 13,822 observations. A preliminary check excluded 155 observations because they involve article categories that have too little observations to apply the intra-category training set-up. As a result of the outlier detection method proposed in Subsection 3.2.1, 446 observations are excluded because the sales uplift is smaller than 1, and 106 observations are excluded because the discount-normalized lift falls outside of the adjusted interquartile range. Hence, filtering and outlier detection do not lead to a serious loss in data. After splitting, the selection set contains 8,369 observations between 2023 week 25 and 40, and the evaluation set contains 4,743 observations between 2023 week 41 and 46.

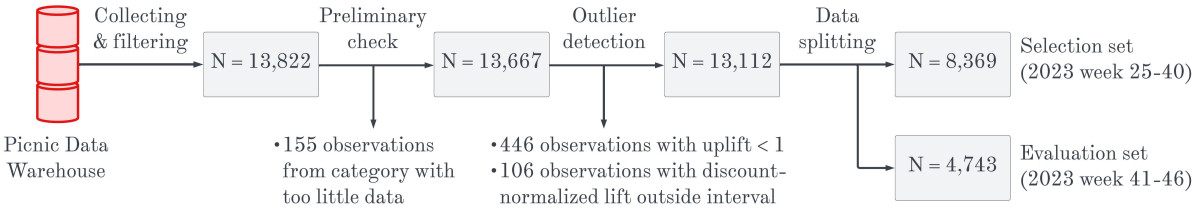
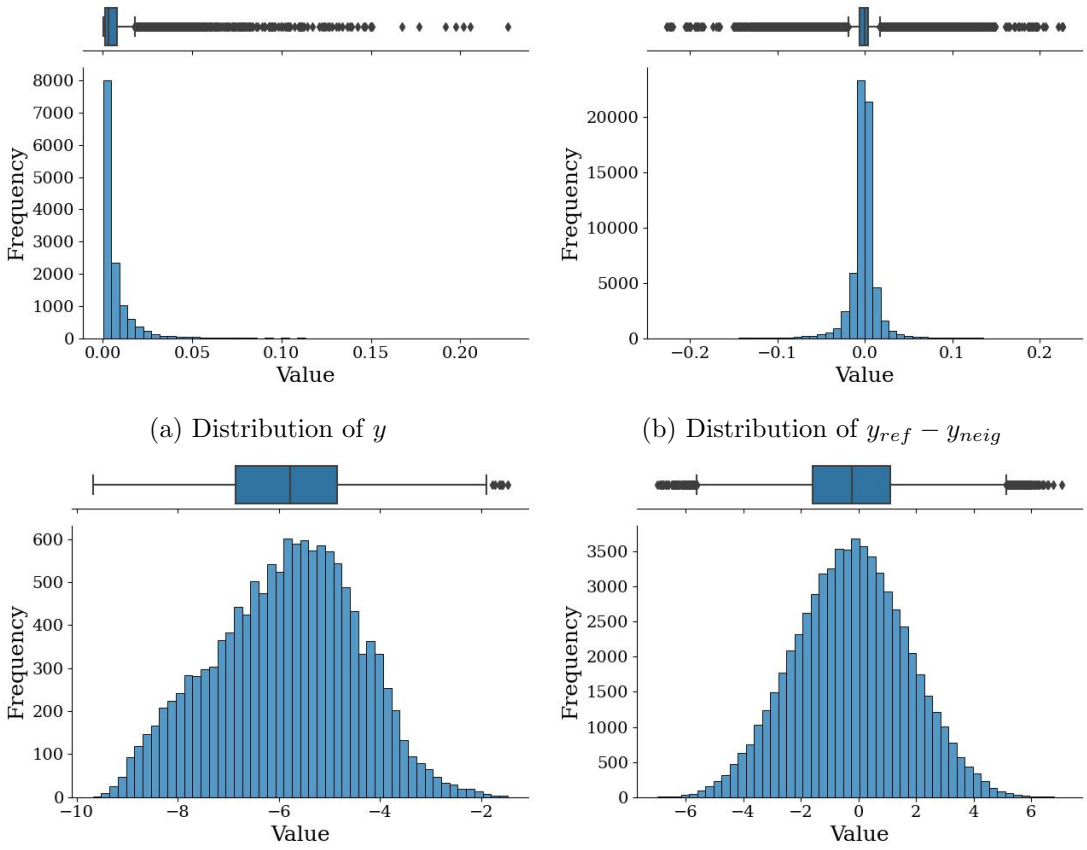


Figure 4.1: Process of data collection, outlier detection, and splitting

4.5.2 Distribution of target variable and features

Histograms (for numericals) and countplots (for categoricals) are computed for the target variable and features to evaluate their distributions. A full overview of all distributions is given in Appendix A.1. The following subsection discusses the noteworthy findings that have direct implications for the modeling choices made later.

Regarding the target variable, Figure 4.2a shows that the data for clean ADR (for simplicity here called y) is strongly positive skewed, with a mean of $6.94 \cdot 10^{-3}$, median of $3.08 \cdot 10^{-3}$, skewness of 6.1, and range of $[6.27 \cdot 10^{-5}, 2.23 \cdot 10^{-1}]$. This is caused by a handful of popular routine articles (mainly fresh fruit & vegetables such as cucumber, tangerine, and bell pepper). As a result, the target of the tree algorithm in the direct regression models (models 2-3) is strongly positive skewed. Furthermore, recall that the response variable of the observations in the contrastive training set, which is used for training the contrastive regression model, is defined as $y^{\text{ext}} = y_{\text{ref}} - y_{\text{neig}}$ (for an arbitrary reference promotion and randomly sampled neighbor). Figure 4.2b shows that this variable (as a natural consequence of the skewness of y) contains extreme positive and negative values. Hence, the target of the tree algorithm in the contrastive regression model (model 1) has very heavy tails. When the model target in a regression tree algorithm is strongly positive skewed or has heavy tails, the extreme values will more prominently affect the variances and the node splits will be drawn towards extreme values as well. As a result, the tree algorithm can potentially be forced to isolate the tails of the data from the rest of the data points, leading to less balanced node splits. This in turn will eventually create a regression tree that performs better in forecasting the extreme values, while its accuracy is lower for data points with the target variable around the median. As a solution to the skewness of y and heavy tails of y^{ext}, we propose to apply a log-transformation. For y, we transform via $y_{\log} = \log(y)$, which indeed yields the more desirable symmetrical distribution shown in Figure 4.2c. For y^{ext},



(a) Distribution of y

(b) Distribution of $y_{ref} - y_{neig}$

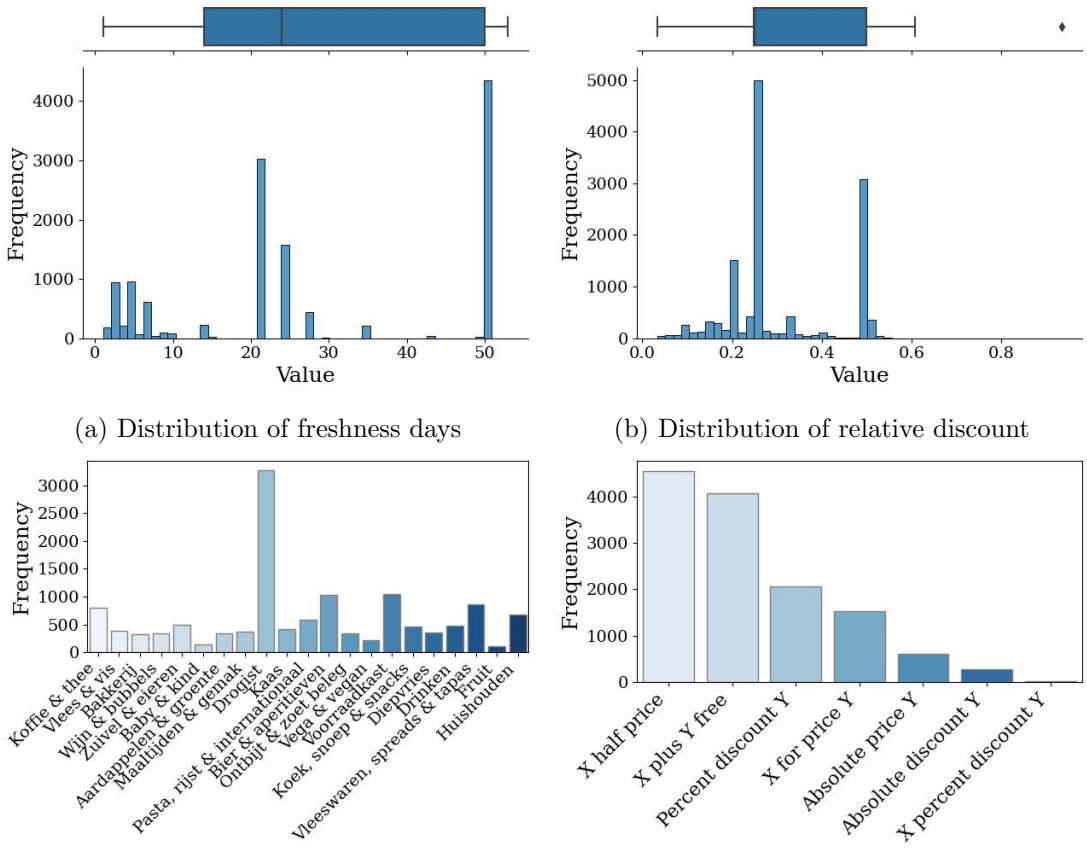
(c) Distribution of $\log(y)$

(d) Distribution of $\log(y_{\text{ref}}) - \log(y_{\text{neig}})$

Figure 4.2: Distribution of model targets before and after log-transformation (y = clean ADR)

we transform via $y_{\log}^{\text{ext}} = \log(y_{\text{ref}}) - \log(y_{\text{neig}})$ to ensure the log-transformation is not taken for negative numbers. Note that we can also rewrite this to $y_{\log}^{\text{ext}} = \log(\frac{y_{\text{ref}}}{y_{\text{neig}}})$, so a different way of looking at this is: we first take the ratio between the demand of the reference promotion and the neighbor (instead of the difference), and then we log-transform this ratio. Figure 4.2d shows that $y_{\log}^{\text{ext}}$ also has a more desirable symmetrical distribution. The log-transformations as proposed above will be carried out prior to training the regression tree algorithms in models 1-3, after which the output of these algorithms will immediately be back-transformed to clean ADR before continuing with possible remaining steps of the model.

Regarding the features, the variables baseline ADR, article content, promotion group size and selling price are also positive skewed. There is however no need to log-transform this data, because this transformation will not change the node splits in the regression tree. The reason for this is that a tree algorithm builds node splits for each numerical feature based on a "greater or less than" condition, where the split value that yields the largest impurity decrease is selected. Hence, any monotonic transformation that does not change the order of the values (such as a log-transformation) will also not change the relative position of the node splits, leading to the exact same final tree. Further, Figure 4.3a shows that most promotions have a freshness of around 3-7 days (fresh food articles), 21 days (ambient food articles), or 50 days (non-food articles). Similarly, Figure 4.3b shows that the relative discounts of 20%, 25%, and 50% are most prominent. The peaks in the distribution of these variables may cause the model to



(a) Distribution of freshness days

(b) Distribution of relative discount

(c) Distribution of article category

(d) Distribution of promotion mechanism

Figure 4.3: Distribution of subset of features

underperform for promotions that do not correspond with these standard values for discount or freshness. Next, Figure 4.3c shows that approximately one-third of the observations comes from the article category "Drugstore" (NL: "Drogist"). For the contrastive regression model that applies inter-category training, this might bias the tree algorithm towards performing relatively well for this category while performing worse for the others (as opposed to the intra-category approach, which trains a seperate tree for each category). However, the imbalance in the article category variable is fully representative of the real-life situation at Picnic, because the majority of article promotions indeed comes from the "Drugstore" category. As the business goal for Picnic is to minimize the total sum of absolute forecast errors, we deliberately decide not to handle this imbalance by for example applying stratified sampling. Lastly, Figure 4.3d shows that the promotion mechanism "absolute discount Y" is underrepresented, which could also lead to the models underperforming for future promotions that apply this mechanism.

4.5.3 Correlation between features

Although the forecasting performance of tree-based algorithms naturally does not suffer from multicollinearity of features, the feature importance calculation does run the risk of becoming unstable when features are highly correlated (Nicodemus et al. (2010)). As the contrastive regression framework proposed in this research directly makes use of feature importance scores to compute a forecast, severe multicollinearity is undesirable. In this research, the correlation

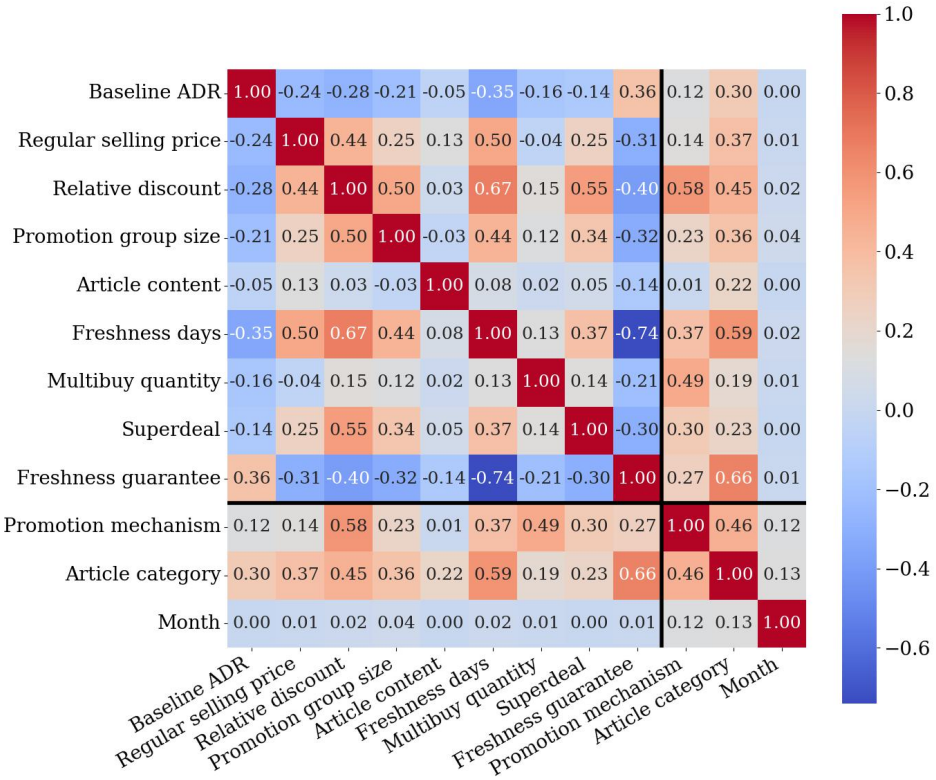


Figure 4.4: Correlation heatmap of features

for numerical-numerical variable pairs is denoted by Spearman's rank correlation coefficient, a non-parametric measure that describes the monotonic relationship between two variables by assessing the degree to which their ranks are correlated. A value of -1 indicates a perfect negative correlation, 0 denotes no correlation, and 1 denotes a perfect positive correlation. In this research, pairs with an absolute Spearman's coefficient of $|r| > 0.7$ are marked as collinear, following the work and corresponding suggestion by Dormann et al. (2013). For numerical-categorial pairs, the correlation is denoted by eta-squared, the proportion of the variance in the numerical variable that is explained by the different categories in the categorical variable. Eta-squared ranges between 0 (no association) and 1 (perfect association), where $\eta^2 > 0.6$ is used as collinearity threshold in this research Cohen et al. (2013). For categorial-categorical pairs, the correlation is denoted by Cramér's V, a measure of association between two nominal variables that is based on the chi-squared statistic normalized by the number of observations and the minimum number of category levels. A value of 0 denotes no association, and a value of 1 denotes perfect association. In this research, pairs with a Cramér's V of $V > 0.6$ are marked as collinear, following the work carried out by Rea and Parker (1992) and Kyu (2016) suggesting that above this threshold variables have a "strong association".

Figure 4.4 shows a heatmap plot of the correlation between all 12 features, calculated as describe above. For the variable pair freshness days & freshness guarantee we observe $r = -0.74$, which is to be expected as they describe the same article characteristic (namely, freshness of an article) in an opposite manner. Based on the fact that freshness guarantee can be considered a binarized version of freshness days, we expect freshness days to contribute more to the forecasting

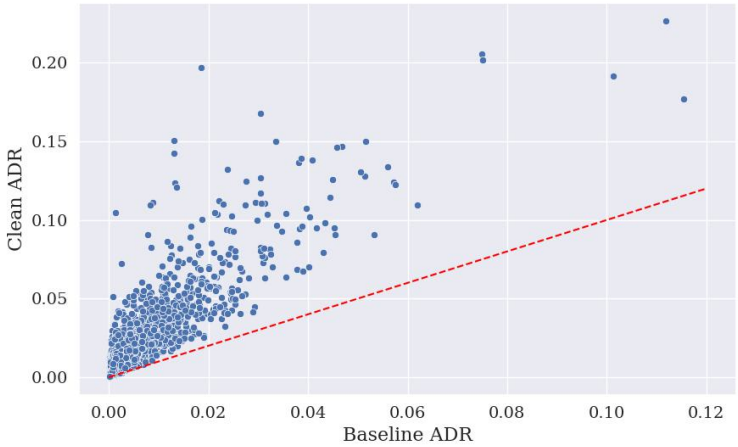


Figure 4.5: Correlation between clean ADR and baseline ADR

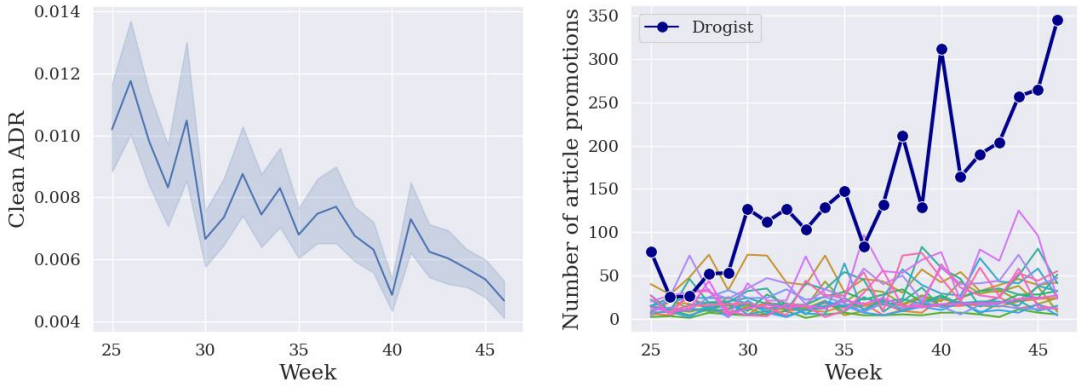
performance. Furthermore, for the variable pair article category & freshnes guarantee we observe $\eta^2 = 0.66$. Based on these two insights, we decide to exclude the variable freshness guarantee from the features when building the models. Other noteworthy correlations that are just below the earlier defined thresholds are observed for the pairs freshness days & discount ($r = 0.67$, i.a. because non-food articles in general have higher discounts), promotion mechanism & relative discount ($\eta^2 = 0.58$, i.a. because "X plus Y free" in general has higher discounts), and article category & freshness days ($\eta^2 = 0.59$, i.a. because articles from "Meat & Fish" in general have less freshness days). Lastly, the variable article category shows to be fairly correlated to almost all other variables. This might suggest that some article categories are associated with a distinctive set of feature values that is different from other categories, and hence that the intra-category training approach might work better than the inter-category approach.

4.5.4 Correlation between model target and features

Scatter plots are computed to display potential correlations between the model target and all features. An overview of the resulting scatter plots is given in Appendix A.2. These plots will indicate to what extent the features have a positive or negative association with the target variable, and hence gives a first indication of which features will be important in the forecasting model. The one insight worth mentioning here is the obvious correlation between clean ADR and baseline ADR, which is displayed in Figure 4.5. The red-dotted line denotes where clean ADR is equal to baseline ADR, and hence where the sales uplift is equal to 1. The scatter plot shows a clear positive correlation, where each observation lies above the red-dotted line suggesting a sales uplift larger than 1. This positive correlation is logically explained by the fact that promotional demand is always an upscaled version of the baseline demand.

4.5.5 Model target over time

Lastly, we analyse the behaviour of the model target (clean ADR) over time to check for a possible trend or seasonality that can not be explained by the behaviour of the features. A full overview of the behaviour of the model target and all features over time is given in Appendix



(a) Clean ADR (b) Number of article promotions per category

Figure 4.6: Clean ADR and number of article promotions per category over time

A.3. Figure 4.6a plots the weekly mean and corresponding 95% percentile interval of clean ADR, showing a clear downward trend. This trend can be logically explained by the gradual shift in the composition of the set of weekly article promotions at Picnic. This shift is visible in Figure 4.6b, which gives the number of article promotions per category over time. The number of promotions for the category 'Drugstore' (NL: 'Drogist') increased significantly from week 25 to 46, while this number remained relatively stable for other categories. This increase resulted when Picnic intensified the promotional strategy for this category in 2023. As articles from this category generally have low baseline demand (and hence, low promotional demand) compared to other categories, this led to a decrease of the average weekly clean ADR. This shift in the composition of the set of weekly promotions can affect the performance of the CR-CBinter model in the final model evaluation: as the share of promotions coming from the category 'Drugstore' is lower in the training set (week 25-40, also called the selection set) than it is in the evaluation set (week 41-46), we can not officially state that observations in both sets come from the same underlying data generating process. Therefore, we should critically assess whether the performance of the inter-category model during model selection also generalizes to the unseen data in the final model evaluation. This will indicate whether the inter-category contrastive regression framework is able to handle this shift in composition of weekly set of promotions, and whether it is robust enough to maintain high accuracy when the promotion strategy changes.

As the trend in clean ADR can be fully attributed to deliberate changes in Picnic's promotion strategy, and these changes are fully captured by the features, no trend variable is included in the models. Considering seasonality, we still hypothesize that the time of the year in which an article promotion is done to some extent affects the promotional demand. Part of this seasonal effect is expected to be captured by the seasonality in baseline ADR, but the remaining part is expected to be caused by a seasonality in the promotional uplift (e.g., an article promotion for hot chocolate is likely to have a higher uplift in December than in July). To allow for the models to learn this seasonality, the cyclical variable "Promotion month" is included as a feature. Note that the dataset used in this research contains only six months of promotions, hence more data is needed for the model to learn the full-year seasonal pattern.

Chapter 5

Results

In this chapter, the results of the research carried out in this report are displayed. First, Section 5.1 provides the outcome of the comparison of the four candidate contrastive regression models, and the selection of the final model. Next, Section 5.2 interprets this final model and shows the results of its comparison with the baseline methods. All results are computed using Python version 3.11.2, Jupyter Lab version 3.6.6, an Intel Core i7-8665U processor with 1.90GHz CPU frequency, and Windows 11 Pro version 23H2.

5.1 Model selection

This section describes the results of the model selection procedure explained in Section 3.4. The performances of the four candidate contrastive regressors in forecasting clean ADR are given in Table 5.1. The numbers regarding MAE are scaled by factor 10^3 for readability. The mean MAE is $1.80 \cdot 10^{-3}$ and $1.90 \cdot 10^{-3}$ for CR-CBinter and CR-CBintra, respectively. For CR-ERTinter and CR-ERTintra, the mean MAE is $1.98 \cdot 10^{-3}$ and $2.27 \cdot 10^{-3}$, respectively. Regarding the confidence intervals, we observe that the MAE of the CR-CBinter, CR-CBintra and CR-ERTintra models display similar variance. The variance of CR-ERTinter is slightly lower, which could be a consequence of the earlier mentioned property of lower variance for the ExtraTrees algorithm. The results suggest that the CatBoost-based models outperform their ExtraTrees-based counterparts, because the mean MAE for CR-CBinter and CR-CBintra is notably lower than for CR-ERTinter and CR-ERTintra, respectively. Furthermore, the results also suggest that the overall performance of inter-category training is higher than that of intra-category training, as the mean MAE for CR-CBinter and CR-ERTinter is lower than for CR-CBintra and CR-ERTintra, respecively. However, we hypothesize that the two training set-ups (inter- vs. intra-category) might perform differently for different subsets of article promotions. Therefore, we decide to evaluate both the CR-CBinter and CR-CBintra model in more detail in the final model evaluation of Section 5.2.

Candidate model	MAE (10^{-3})		WAPE (%)	WPE (%)
	Mean	95% CI	Mean	Mean
CR-CBinter	1.80	[1.68, 1.92]	21.6	-4.5
CR-CBintra	1.90	[1.76, 2.04]	23.9	-6.0
CR-ERTinter	1.98	[1.89, 2.07]	24.6	-11.4
CR-ERTintra	2.27	[2.12, 2.42]	28.8	-7.3

Table 5.1: Performance of four contrastive regression candidates in forecasting clean ADR (results of generalization error for 15 rounds of cross-validation)

Regarding the error rate relative to the actual observed clean ADR, the WAPE of the models ranges between 21.6% and 28.8%. This result is promising and suggests that the contrastive regression framework has potential to improve the manual forecasting process at Picnic (with a current WAPE of around 30% on average). The results in the next section however will provide a more detailed comparison between the contrastive regression framework and manual forecasting. Furthermore, the WPE shows that the CatBoost-based models on average underforecast by 4.5% and 6.0% with inter- and intra-category training, respectively. The ExtraTrees-based models on average underforecast by 11.4% and 7.3% with inter- and intra-category training, respectively. From this we can conclude that the contrastive regression framework on average tends to slightly underforecast the clean ADR, and that this underforecasting is more prominent for the ExtraTrees-based models. A possible explanation for underforecasting could be the presence of omitted variables that are not stable over time, preventing the contrastive regressor to correctly identify that an upcoming promotion has higher demand compared to its historical neighbors. Another explanation could be that relationship between two random neighbors in the contrastive training set is not fully representative of the relationship between a test promotion and similar historical promotion, causing to underforecast in the latter case.

5.2 Final model evaluation

This section describes the results of the final model evaluation procedure from Section 3.5.

5.2.1 Overall forecasting performance

The results of the comparison of the overall performance between the CR-CBinter & CR-CBintra models and the five baseline models are given in Table 5.2. The table gives the MAE, WAPE, WPE, and R^2 for each model, as well as the p-value for the six Wilcoxon signed-rank tests (WSRT) that compare the CR-CBinter model with the other models, and lastly the computational load for each model. The numbers regarding MAE are scaled by factor 10^3 for readability.

The CR-CBinter model, direct regression with CatBoost, and direct regression with ExtraTrees show similar overall forecasting accuracy, with a MAE of $1.60 \cdot 10^{-3}$, $1.58 \cdot 10^{-3}$, and $1.59 \cdot 10^{-3}$, respectively. The CR-CBintra model and weighted nearest neighbor regression slightly underperform compared to the aforementioned three, with a MAE of $1.68 \cdot 10^{-3}$ and $1.79 \cdot 10^{-3}$, respectively. Lastly, all five regression models largely outperform the manual forecasts and naive forecasts, which have a MAE of $2.32 \cdot 10^{-3}$ and $3.61 \cdot 10^{-3}$, respectively. The WAPE

Model	Forecasting performance				WSRT	Computational load
	MAE (10^{-3})	WAPE (%)	WPE (%)	R^2	p-value	Runtime (s)
CR-CBinter	1.60	27.5	-0.1	0.88	-	210
CR-CBintra	1.68	29.1	3.1	0.89	<0.001	437
Direct CatBoost	1.58	27.3	3.4	0.89	<0.001	345
Direct ExtraTrees	1.59	27.5	1.6	0.90	<0.001	21
Weighted NNR	1.79	31.0	9.8	0.84	<0.001	364
Naive forecast	3.61	62.3	43.7	-0.13	<0.001	<1
Manual forecast	2.32	40.1	23.0	0.78	<0.001	<1

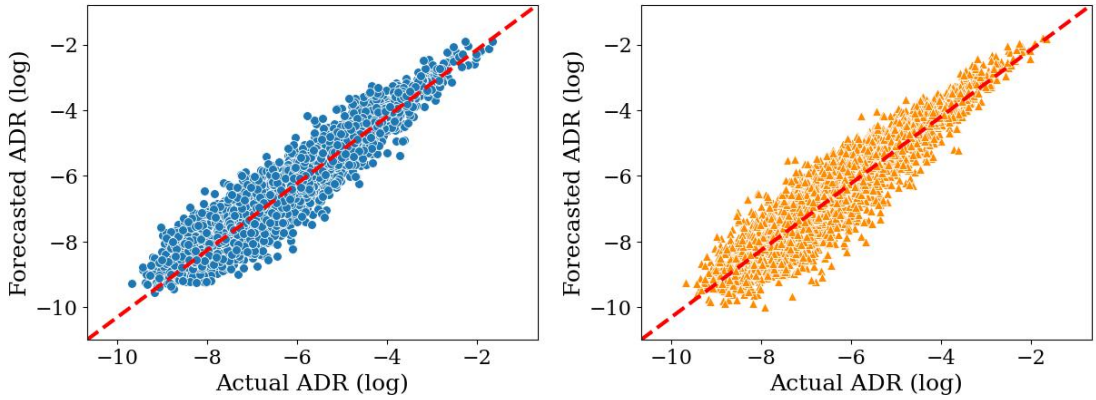
Table 5.2: Accuracy and computational load of contrastive regressors versus baseline models (results of training on 8,369 observations and forecasting 4,743 observations)

of the CR-CBinter and CR-CBintra models (27.5% and 29.1%, respectively) is higher than their WAPE during model selection (21.6% and 23.9%, respectively) while the MAE is actually lower, from which we can deduct that the mean of the observed clean ADR is lower in the evaluation set than it is in the selection set (this is in accordance with our earlier conclusion drawn from Figure 4.6a in Section 4.5). From the WPE we conclude that the CR-CBinter model on average equally over- and underforecasts, while the other four regression models slightly overforecasted clean ADR. Both the manual and naive forecast primarily overforecast clean ADR, with a WPE of 23.0% and 43.7%, respectively. The results for R^2 show that all five regression models are able to explain largest part of the variance of the observed clean ADR, with an R^2 ranging between 0.84 and 0.90. The manual forecast shows to have reasonable explanatory power with an R^2 of 0.78, while the naive prediction shows to be worse than predicting the mean ADR (R^2 of -0.13).

The Wilcoxon signed-rank tests comparing the forecasts of the CR-CBinter model to the other six models produce p-values below 0.001. Hence, we conclude that we can reject the hypothesis that the median difference between the paired observations of the CR-CBinter model and the other six models is zero, given a $\alpha^* = \frac{0.05}{6}$ significance level.

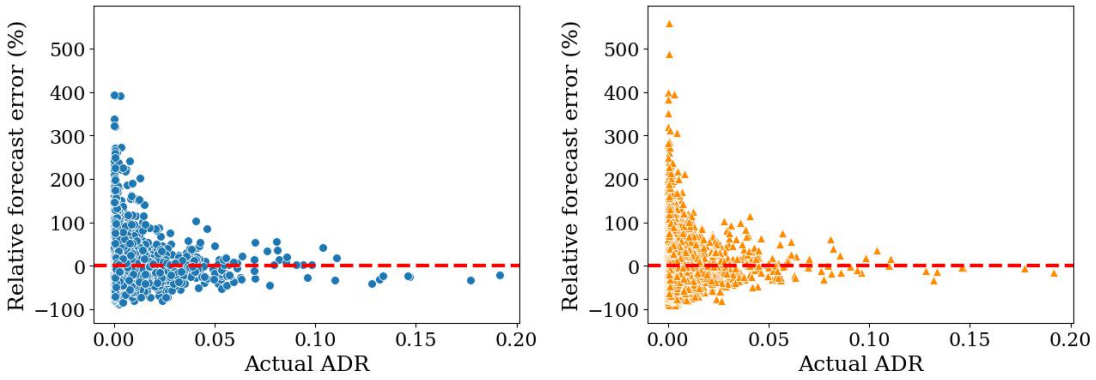
Figure 5.1 shows scatter plots of the forecasts versus observed values of clean ADR for the two contrastive regression models. Both the x-axis (actuals) and the y-axis (forecasts) are log-transformed to facilitate an easier interpretation of the plot. The red-dotted line follows the equation $y = x$, hence it represents a "perfect model" where the forecasts are equal to the observed values. From these two figures (combined with the R^2 scores of 0.88 and 0.89) we can infer that both contrastive regression models successfully capture the relationship between the promotional sales and the resulting clean ADR, and that they can compute reasonable forecasts.

Figure 5.2 shows scatter plots of the relative forecasting error (the forecasting error divided by the actual value, given as a percentage) versus the actuals for the two contrastive regression models. It can be clearly observed for both models that the relative forecasting errors (both in positive and negative direction) are largest for small values of actual clean ADR, and then consistenly decrease as the actual clean ADR becomes larger. This pattern is a logical consequence of the way we optimized the models: the goal is to minimize the sum of absolute errors at Picnic, hence the Mean Absolute Error (MAE) was used as the error metric in model training. As a result, the relative error for popular article promotions with high ADR is very well contained, while (inevitably) more "room for error" is left for article promotions with lower ADR. Eventually, this is the primary objective for Picnic's promotion demand forecasting process: ensuring



(a) CR-CBinter (b) CR-CBintra

Figure 5.1: Forecasted vs actual clean ADR for two contrastive regression models (red-dotted line denotes where forecast is equal to actual value)



(a) CR-CBinter (b) CR-CBintra

Figure 5.2: Relative forecasting error vs actual clean ADR for two contrastive regression models (red-dotted line denotes where relative error is equal to zero)

that the absolute number of wrongly forecasted articles is kept to a minimum, to minimize the total customer impact and food waste across the whole promotional assortment. Comparing the plots of the two models, we observe that the CR-CBinter model tends to underforecast more than the CR-CBintra model, which is also in line with the higher WPE of the latter in Table 5.2.

To gain a better understanding of the potential causes behind inaccurate forecasts computed by the contrastive regression framework, we leverage its ability to produce contrastive explanations and apply this to two CR-CBinter forecasts with a large relative error. The contrastive explanations for these two forecasts are given in Appendix C. The first test observation, with contrastive explanation given in Table C.1, was overforecasted with a relative error of 392%. The test promotion is very similar to the five neighbors for all features but the relative discount: while the nearest neighbors are "1+1 free" promotions with 50% discount, the test promotion has a "2+1 free" mechanism with 33% discount. The actual observed ADR for the five neighbors ranges between $36.12 \cdot 10^{-4}$ and $99.70 \cdot 10^{-4}$, which corresponds to uplifts between roughly 10-20. The contrastive regressor successfully predicted that the lower relative discount of the test promotion will lead to a lower ADR, hence the forecasted differences in ADR between

the five neighbors and the test promotion are negative (ranging between $-14.01 \cdot 10^{-4}$ and $-48.69 \cdot 10^{-4}$). These forecasted differences however were not large enough: the final forecasted ADR was $33.18 \cdot 10^{-4}$ (an uplift of 6.7), while the actual observed ADR was $6.74 \cdot 10^{-4}$ (an uplift of 1.4). From this first example, we can learn the following two things: on the one hand, the contrastive regressor does seem to learn the correct positive association between relative discount and resulting ADR. On the other hand, the model did not succeed in computing an accurate forecast in this specific case. This can be partially attributed to the fact that this specific test promotion has an unconventionally low uplift given the specific mechanism and discount. However, it could also indicate that certain omitted variables exist that can explain the low ADR but are not yet included in the feature space.

The second test observation, with contrastive explanation in Table C.2, was underforecasted with a relative error of -93%. In this case, the test promotion is very similar to the five neighbors for all features, hence the differences in ADR are correctly predicted to be small. The resulting final forecasted ADR was $3.19 \cdot 10^{-4}$ (an uplift of 1.9), while the actual observed ADR was $34.08 \cdot 10^{-4}$ (an uplift of 19.8). We can safely state that this specific test promotion has an unconventionally high observed ADR, when compared to the five neighbors. This again indicates that there might be omitted variables that can explain this high ADR, but are not yet included in the feature space.

All in all, the two examples suggest that the pre-training outlier detection method might not be able to detect all outliers, as for both test observations the actual observed ADR is very different from what would be expected based on the nearest neighbors. This large differences in ADR could be caused by one or more omitted variables that have high explanatory power, but are not yet included in the feature space.

5.2.2 Computational load

The results of tracking the model runtimes are also given in Table 5.1. This runtimes correspond to one cycle of training the model on a dataset of $N_{train} = 8369$ observations and predicting the target for a dataset of $N_{test} = 4743$ observations. The regression tree algorithms are all trained with 500 iterations (or 500 estimators for ExtraTrees), learning rate of 0.05 and tree depth of 8.

The contrastive regression models have a runtime of 210 seconds and 437 seconds for inter- and intra-category training, respectively. Direct regression using CatBoost and ExtraTrees takes 345 seconds and 21 seconds, respectively, and Weighted Nearest neighbor Regression (NNR) has a runtime of 364 seconds. Lastly, the naive forecast and manual forecast do not make use of a learning algorithm, hence their running time is negligible. Comparing the two contrastive regressors, we can conclude that the intra-category training approach is more computationally demanding, which could be caused by the fact that we have to train a new CatBoost learner for each category. Comparing the CR-CBintra model with direct regression using CatBoost and weighted NNR, we see that direct regression and weighted NNR are roughly 20% faster than contrastive regression. To try to explain these results, we see two potential causes: first, recall that the number of observations in the contrastive training set used to train the tree algorithm in the contrastive regressor is $N_1 \cdot k_1$ (where N_1 is the number of observations in the training set for direct regression, and k_1 is the number of randomly selected neighbors used to set-up the

contrastive training set). Second, note that the contrastive training set contains twice as many features as the original training set (one set of features for the reference promotion, and one set for the neighbor). These two points could explain the longer runtime for the CR-CBintra model compared to direct regression using CatBoost and weighted NNR. Lastly, direct regression using ExtraTrees shows to be significantly less computationally demanding than direct regression using CatBoost (21 seconds vs. 345 seconds). We expect this to be due to three main reasons: first, ExtraTrees builds decision trees using random split points instead of searching for the best split value at each node, the latter being computationally more intensive. Second, ExtraTrees does not require the process of optimizing sequential trees needed for gradient boosting. Third, CatBoost needs to create multiple random permutations of the data for the ordered target encoding, whereas ExtraTrees doesn't.

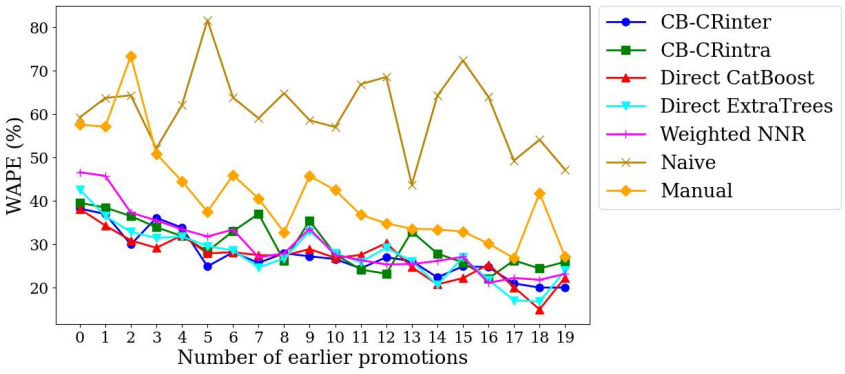
5.2.3 Feature importances in contrastive regressor

Table 5.3 displays the feature importances for the CR-CBinter and CR-CBintra models. For the CR-CBintra model, the table shows the average feature importance from all category-specific trained regression tree, weighted for the number of test observations from each category.

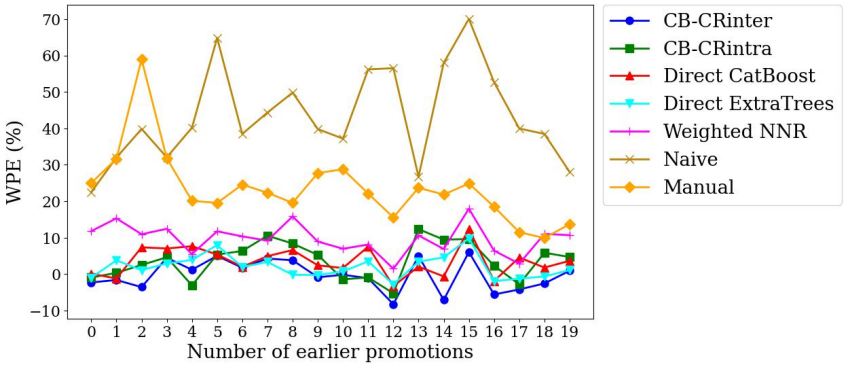
For the inter-category model, the most important features are baseline clean ADR (67.5%), relative disocunt (9.5%), promotion group size (6.9%), article category (5.1%), and regular selling price (3.0%). For the intra-category model, these are baseline clean ADR (60.4%), relative discount (10.4%), promotion group size (8.2%), regular selling price (7.4%), and promotion mechanism (4.9%). Hence, the baseline demand, discount, selling price, and size of the promotion group contribute most to the forecast in both models. Furthermore, the features article content, freshness days, multibuy quantity, superdeal, and promotion month show to be less important, with importance scores ranging between 0.4-2.9%. Lastly, comparing the importance scores of the sets of features belongings to the reference and neighbor shows that both contribute equally to the final forecast.

Feature	Importance score (%)					
	CR-CBinter			CR-CBintra		
	Reference	neighbor	Combined	Reference	neighbor	Combined
Baseline clean ADR	33.9	33.6	67.5	31.9	28.5	60.4
Regular selling price	1.3	1.7	3.0	3.1	4.3	7.4
Relative discount	4.7	4.8	9.5	5.0	5.4	10.4
Promotion group size	3.5	3.4	6.9	4.1	4.1	8.2
Article content	0.7	0.5	1.2	0.7	0.5	1.2
Freshness days	0.5	0.5	1.0	0.9	0.9	1.8
Multibuy quantity	0.2	0.2	0.4	0.5	0.5	1.0
Superdeal	0.7	1.1	1.8	0.7	1.1	1.8
Promotion mechanism	0.9	0.9	1.8	2.1	2.8	4.9
Article category	2.6	2.5	5.1	-	-	-
Promotion month	0.9	0.9	1.8	1.4	1.5	2.9

Table 5.3: Feature importance scores of final contrastive regressor



(a) WAPE



(b) WPE

Figure 5.3: WAPE and WPE for varying number of earlier promotions available

5.2.4 Forecasting performance per level of coldness

Figure 5.3 shows the forecasting performance (both WAPE and WPE) for varying numbers of earlier promotions available (also called the 'coldness', as explained in Section 4.1). An example of how to interpret the plots: suppose that for a certain article, we have already done two promotions in the past. Now we want to forecast the demand for a third one, hence we focus on the model performances for the case where 2 earlier promotions are available (level of coldness equal to 2). We can conclude that the CB-CRinter model and direct CatBoost have the best accuracy, both with a WAPE of 30%. Furthermore, CR-CBinter shows to slightly underforecast demand, while direct CatBoost on average overforecasts in this situation, with WPEs of -1% and 8%, respectively. Note that the plot shows the results for the first 20 promotions of an article, as the subset of articles with more than 20 promotions in the test set is too small to draw reliable conclusions.

Several useful insights can be derived from Figure 5.3. First, Figure 5.3a shows that all models display an increasing accuracy as more earlier promotions become available (except for the naive forecast), with a WAPE of around 40% for pure cold-starts and decreasing to around 20-25% as an article gets promoted more often. This is in accordance with our earlier hypothesis stating that forecasts are more accurate when more useful data is available. Second, the five regression models largely outperform the naive forecast and the manual forecast for all degrees of coldness, which suggests that implementing a regression model (regardless of which one) already shows great potential to improve the forecasting process at Picnic. Third, diving deeper into

the comparison of the different regression models for cold-start forecasting, CR-CBinter and direct CatBoost show to slightly outperform CR-CBintra and direct ExtraTrees for the first three promotions of an article, although these differences in accuracy can be considered small. In addition to this, weighted NNR shows to perform worse than the other four regression models for the first three promotions, but shows comparable performance for forecasts with more earlier promotions available. When an article has had more than three promotions, CR-CBinter and direct CatBoost most consistently retain a WAPE below 30%, however no clear conclusion can be drawn on which model strictly outperforms the other models in terms of overall forecasting accuracy.

Looking at Figure 5.3b, we confirm that the manual forecast, naive forecast, and weighted NNR all on average overforecast demand, with a strictly positive WPE for all levels of coldness. The two contrastive regression and two direct regression models all show the same pattern of relatively stable WPE for all levels of coldness. The CR-CBinter model generally has a WPE slightly below zero, whereas the other three have a WPE around or slightly above zero.

5.2.5 Forecasting performance per article category

Table 5.4 shows the forecasting performance per article category for the CR-CBinter model, CR-CBintra model, weighted nearest neighbor regression (WNNR) and manual forecast. These three regression models are selected and compared to manual forecasting because they provide direct explainability and flagging of the forecasts, hence they are most suitable to replace the current process at Picnic. For each article category and each model, the WAPE and WPE are given as a percentage. Note that the article categories are split into two types: "fresh food" categories, and "non-fresh food or non-food" categories. In general, overforecasting is more problematic for the fresh food categories, because this more often leads to food waste due to lower shelf lives. The goal of this table is to identify which forecasting methods are particularly well suited for which categories by looking at the overall absolute error (WAPE) and the level of over- or underforecasing (WPE).

The WAPE and WPE for the best performing model per category are in bold. We conclude that for 11 out of 21 categories, the CR-CBinter model shows most potential to improve the forecasting accuracy compared to the manual forecast. For 5 other categories, the CR-CBintra model shows to be most promising. WNNR has the best performance for another 3 out of 21 categories, and for the last 2 categories manual forecasting still seems to be the best option. Note that this selection is primarily based on which model has the lowest WAPE, except for the categories "Breakfast" and "Drugstore": although the CR-CBinter model here has the lowest WAPE, WNNR is selected because this method on average overforecasts more than CR-CBinter, which is preferred for this categories. Furthermore, note that the manual process in general heavily overforecast the promotions for non-fresh food or non-food categories: this can be traced back to the "better safe than sorry" strategy of forecasting analysts at Picnic, who generally tend to overforecast promotions from these categories. As this leads to excess stock and congested warehouses, this is also not desirable. Concluding, contrastive regression and weighted nearest neighbor regression show clear potential to improve the forecasting operation at Picnic for a large part of the promotional assortment.

Category type	Category	Forecasting performance							
		CR-CBinter		CR-CBintra		WNNR		Manual	
		WAPE	WPE	WAPE	WPE	WAPE	WPE	WAPE	WPE
Fresh food	Cheese	**23.3**	**2.1**	29.9	9.9	26.7	11.0	43.4	28.7
	Dairy & eggs	24.7	-8.9	**21.4**	**-3.0**	24.7	-2.3	29.0	6.8
	Cold cuts, spreads & tapas	**25.5**	**-0.3**	26.0	11.6	29.1	12.8	26.1	14.2
	Meat & fish	25.4	-6.0	21.3	9.3	23.5	3.2	**18.9**	**-0.2**
	Potatoes & vegetables	24.6	-3.1	**18.6**	**-3.5**	21.9	10.7	25.0	14.8
	Vegetarian & vegan	**24.1**	**12.7**	27.8	22.5	34.4	16.5	37.9	-23.6
	Fruit	**25.5**	**4.3**	36.5	26.9	48.6	45.5	27.9	21.5
	Ready meals	30.6	0.9	32.4	-0.5	31.7	3.2	**28.6**	**-0.2**
	Bakery	**24.3**	**3.3**	36.9	26.2	29.2	16.9	27.1	10.3
Non-fresh food or non-food	Baby & child	**27.1**	**-8.3**	61.9	58.7	106.7	90.2	51.3	31.4
	Pantry	**27.5**	**10.2**	41.9	30.1	34.9	20.1	57.3	38.1
	Beer & appetizers	22.0	-4.9	**19.8**	**-3.7**	22.7	0.5	36.3	22.6
	Coffee & tea	**27.6**	**-7.0**	29.5	-5.7	32.0	2.2	53.5	43.8
	Home & cleaning	34.8	-9.4	**33.7**	**-14.2**	37.9	11.9	57.9	34.7
	Pasta, rice & world foods	**28.1**	**-0.3**	29.9	-0.8	32.4	6.9	36.0	20.4
	Drinks	**17.8**	**-1.7**	22.6	-10.0	29.9	22.1	71.6	66.6
	Cookies, candy & snacks	36.0	-23.5	34.4	-18.7	**34.0**	**-5.6**	47.8	28.4
	Frozen	47.8	17.6	**36.0**	**16.5**	37.4	24.2	69.5	62.4
	Wine	**29.3**	**5.4**	37.7	14.9	35.1	11.3	35.2	-1.4
	Breakfast	22.5	-8.5	26.8	1.4	**23.3**	**3.2**	39.3	31.1
	Drugstore	38.1	-24.1	43.1	-34.2	**43.2**	**-17.3**	81.0	42.4

Table 5.4: Forecasting performance per article category for CR-CBinter, CR-CBintra, WNNR and manual forecast (WAPE and WPE are given in percentages)

Chapter 6

Conclusion

Recalling what was introduced in Chapter 1, this research focused on four main objectives. This chapter reflects on these objectives and summarizes the concluding insights from the results.

The first objective was to develop a model that provides interpretable demand forecasts for cold-start promotions. To accomplish this, a contrastive regression model is adopted that forecasts difference in demand between two promotions and provides post-hoc explanations of forecasts. It makes use of the CatBoost algorithm and a k nearest neighbor search, and was originally designed for intra-category training (i.e., training on and forecasting for solely one article category). This research proposed four extensions to the existing contrastive regression model. First, the model was enriched with a pre-training outlier detection method based on the adjusted boxplot for skewed data. As a result, 4.0% of the data was marked as an outlier and removed, but test results later on show that some observations that could potentially be considered an outlier were still present in the data. Second, a heterogeneous distance measure was successfully introduced that enables the use of both numerical and categorical features. Third, the ExtraTrees algorithm was implemented to replace CatBoost as the internal regressor, but this implementation did not indicate to improve forecasting accuracy. Fourth and last, this research extended the contrastive regression model to allow for inter-category training (i.e., training on and forecasting for multiple categories with one model). Results from model selection show that this extension leads to a decrease in the mean absolute error compared to intra-category training. Furthermore, inter-category training reduced computational load by roughly 50%. Testing the inter- and intra-category contrastive regression models on a dataset of unseen observations shows an overall WAPE of 27.5% and 29.1%, respectively. The relative forecast error is largest for observations with low ADR and and decreases as the ADR increases. Furthermore, the results show that inter- and intra-category training are both superior for a different set of article categories, indicating they are not interchangeable and can both add value in a forecasting operation. Regarding interpretability, the contrastive regression model has the ability to provide contrastive explanations. These explanations accompany each forecast with the feature importance scores, a list of the most similar historical promotions, and their similarity to the forecasted promotion. This offers users an intuitive and easy-to-understand way to check the rationale behind each forecast. Together with the absence of feature engineering and the tree-based structure, the contrastive regression model is an interpretable approach to promotional foracasting. Feature importance scores show that baseline demand, relative discount,

promotion group size, article category, and regular selling price are deemed most important to predict the difference in demand between two promotions. Furthermore, results indicate there might be important explanatory variables that are not yet included in the feature space.

The second objective focused on comparing the contrastive regression model to several baseline methods. First of all, contrastive regression largely outperforms Picnic's current manual forecasting process and a simple naive forecast in terms of overall accuracy. This indicates that the model has great potential to improve the manual process, and that a simple model does not suffice. Furthermore, the overall accuracy of the contrastive regression models is slightly higher than that of weighted nearest neighbor regression, and on par with that of direct regression with CatBoost and ExtraTrees. This suggests that the contrastive regression model is the best choice in this case, offering both accuracry and post-hoc explainability of forecasts. Regarding the computational load, the runtime of contrastive regression is approximately 20% higher than that of direct regression and weighted nearest neighbor regression because it requires training on a contrastive dataset that contains k times more observations than the original dataset.

The third objective was to extend the application area of the model beyond strictly cold-start promotions. Evaluating the forecasting accuracy for varying number of earlier promotions available, we observe that the contrastive regression models can reach an impressive WAPE of less than 40% on average for the first promotion of an article. This error decreases to roughly 20% on average as more historical promotions are available. The contrastive regressor with inter-category training and direct CatBoost model show similar performance, outperforming the other models for the first three promotions of an article. From the fourth promotion onwards, these two models also show the most stable performance with a WAPE below 30% on average, but no clear conclusion can be drawn on which model is consistently superior in terms of overall forecasting accuracy. All in al, this suggests that the added value of the contrastive regression model is not limited to forecasting strictly cold-start promotions, but shows potential to improve the forecasting operation for at least the first three promotions of an article.

The fourth objective was to use the model to provide promotional demand forecasts and additional post-hoc explanations in a real-life business example. In this regard, the research aimed at improving the promotional demand forecasting process of Picnic, a European-wide e-grocery retailer. We conclude that both contrastive regression models together with the weighted nearest neighbor regression model show great potential to achieve this improvement. This conclusion is based on two main reasons concerning accuracy and interpretability: first, the three models altogether show to outperform the manual forecasting process for 19 out of 21 article categories, and are on par with direct regression using established tree-based algorithms. Second, the three models all have the ability to provide a post-hoc explanation for each forecast, greatly increasing its usability in Picnic's operation compared to direct regression. Additionally, replacing the current manual process with a regression model will significantly decrease the workload of forecasting analysts at Picnic: whereas the current forecasting process demands roughly 8 to 12 hours per week distributed between two analysts, running a model and interpreting its results will only take approximately one hour (assuming the model is re-trained and used to forecast roughly 1000 article promotions on a weekly basis). More detailed business insights for Picnic are discussed in the next chapter.

Chapter 7

Business implications for Picnic

This chapter describes the business implications of the conducted research for Picnic. The aim is to extract relevant insights from the results, translate these to tangible implications, and do suggestions that can help improve Picnic's business operations.

7.1 Relevant insights and value for Picnic's forecasting process

As discussed already in Chapter 6, the main contribution of this research to Picnic's business operation is the proposal of a model that provides accurate and explainable demand forecasts for promotions. On top of this, the results from this research bring several insights that can be leveraged by Picnic in the future, and the most important ones are discussed below.

First of all, the data analysis shows that the target variable clean ADR is right skewed. This is caused by a handful of popular fresh food article promotions each week that reach an ADR of 0.10 or higher (meaning that more than 10% of all customers that week buys this specific promotion). Because of this high penetration across the customer base, these promotions can "make or break" a certain week: underforecasting greatly hurts customer satisfaction, while overforecasting causes large amounts of food waste. Fortunately, optimizing a model on the MAE will already bias it towards keeping the relative error for these popular article promotions low. To further decrease the risk of unavailability or food waste, Picnic should make special arrangements with suppliers to deliver extra articles ad-hoc when there is a risk of going out-of-stock, or build in extra alerts that warn if sales are lagging during the week.

Regarding the complexity of the promotional demand forecasting process at Picnic, it should not be ignored that no model in this research achieved an overall WAPE below 27% on average. Analysing the forecasts with a large relative error shows that their actual demand was very different from the demand of their five nearest neighbors, despite these neighbors having nearly identical feature values. Apart from the fact that there might be important omitted variables missing in the feature space, this suggests that promotional sales at Picnic also suffer from a fairly large random component that will always remain hard to predict.

As a next point, recall that the inter-category training set-up yielded a higher forecasting accuracy than intra-category training for 14 out of 21 article categories at Picnic. Apparently, forecasting for these categories generally benefits from pooling the observations in one dataset and training one model. This suggests that the articles from these categories share similarities

in their promotional features, but also that they respond somewhat similarly to promotional periods. On the other hand, we expect articles from categories that benefit from intra-category training to display a more unique and distinctive relationship between promotional features and resulting demand. The clusters of categories that can be formed naturally from this provide Picnic with new learnings about their customer base and buying behaviour.

Diving deeper into the results for the inter-category trained model, we observe that the nearest neighbors mainly come from the same category as the forecasted promotion. The hypothesis posed earlier, stating that many cold-start promotions might actually be most similar to historical promotions from outside its own category, is therefore unlikely to be true. The reason for this is that Picnic's dataset already contains roughly 10,000 promotions across the whole assortment, and this number grows by roughly 1,000 every week. As a result, the model generally has no trouble finding five sufficiently similar historical promotions within its own category.

Further, the error of manual forecasting is found to be significantly worse for cold-starts (i.e., when little promotional data is available for the article): the WAPE for the first three promotions of an article is between 60-75%, whereas for later promotions this number decreases to around 30-40%. This underlines how hard it is for Picnic analysts to find the right data and give a correct interpretation when no earlier promotion is done for the article.

Lastly, it is important to realise that the features baseline ADR, promotion group size, article category, regular selling price, freshness days, article content, and promotion month cannot be directly tuned by the promotion team when configuring individual article promotions. Summing their importance scores shows that these "fixed" features together account for roughly 85% of a forecast's variability. On the other hand, the remaining 15% of the feature importances can be attributed to four features the promotion team can tune freely for each individual promotion: relative discount, promotion mechanism, superdeal, and multibuy quantity. This implies that a rough forecast of promotional demand can be computed by already entering the values for the seven "fixed" features, and adding an initial expected value for relative discount, promo mechanism, superdeal, and multibuy quantity. The latter four can then be detailed in a later stage when the promotion plan is sharpened to produce a more accurate forecast.

7.2 Application of contrastive regressor in practice

In this section, a practical example of the application of the contrastive model will display its explainability and usability for forecasting analysts. Note that quantitatively assessing the interpretability of a model is not straightforward and difficult to do objectively, and this is not the focus of this research. Table 7.1 shows an example of how the output dashboard of an article promotion forecast computed by the contrastive regressor could look like.

The upper part of the table shows, from left to right, the features, importance scores, and feature values for the five nearest neighbors and forecasted promotion. The lower part of the table shows, for each of the neighbors, the actual ADR, forecasted difference in ADR between the neighbor and test promotion, forecasted ADR of the test promotion, and the weight of the neighbor (i.e., the inverse of the distance between the neighbor and the test promotion). Lastly, on the bottom right the final weighted forecasted ADR of the test promotion is given.

Feature	Imp. (%)	NN1	NN2	NN3	NN4	NN5	Test promo
Baseline ADR	67.5	162.8	232.4	247.0	233.2	258.5	236.9
Selling price	3.0	1.43	3.49	1.78	3.49	1.66	1.45
Discount	9.5	0.30	0.25	0.25	0.25	0.25	0.33
Group size	6.9	8	9	8	31	42	7
Article content	1.2	1	1	1	1	1	1
Freshness days	1.0	5	7	21	7	21	5
Multibuy qty.	0.4	1	2	2	2	2	1
Superdeal	1.8	Yes	No	No	No	Yes	Yes
Mechanism	1.8	Abs. price	X for Y	X for Y	X for Y	X for Y	Abs. price
Category	5.1	Dairy	Dairy	Dairy	Dairy	Dairy	Dairy
Month	1.8	Sept	Aug	Jul	Oct	Sept	Nov
Actual observed ADR		888.3	402.1	567.5	407.8	557.2	
Forecasted difference in ADR		139.8	444.0	348.2	510.1	432.7	
Forecasted ADR		1028.1	846.1	915.6	918.0	990.0	
Weight		19.4	18.0	16.3	14.1	13.0	
Weighted forecasted ADR							939.6

Table 7.1: Visualization of output dashboard of a promotion forecast using contrastive regressor

The unique approach of the contrastive regressor enables to not only provide this final forecast, but also additional information about the features and neighbors. This information is highly valuable for a promotion analyst, as it helps determining whether a final forecast can be considered reasonable with respect to similar historical promotionis. This check for reliability using post-hoc explanations cannot be done when a more opaque machine learning algorithm such as direct regression with CatBoost or ExtraTrees is used, as these models solely provide the final forecast and the feature importances. Note that for readability, the example in Table 7.1 now only displays data on the model features and ADR. It is however highly encouraged to also include context such as article name and promotion week.

If deemed necessary by the analyst, the contrastive explanation provides three ways to adjust the forecast. First, the feature importance scores can be redistributed if an analyst thinks that, for this forecast in particular, specific features are more important than others. Redistributing the feature importance scores will affect the weights in the distance calculation, hence it will alter the weights of the neighbors. Second, the weights can be readjusted if an analysts thinks that one of the neighbors is not as similar to the test promotion as the weight represents and should be more or less important in determining the final forecast. Third, the final weighted forecast can be directly adjusted if an analyst simply expects to be able to produce a better forecast manually (for example, if extra information is given that is not available to the model, or if the analyst wants to deliberately over- or underforecast).

At Picnic, checking all roughly 1000 weekly promotional forecasts for reliability manually would still take significant time and effort, and this is not a sustainable process in the longer term. On the other hand, for the contrastive regressor to be implemented in an autonomous forecasting process that automatically places orders at suppliers, it is still highly desirable to have such a control mechanism in place. To facilitate this, one can use the forecast governance step from the contrastive regressor (step 5 from Section 3.1) to identify potentially unreliable forecasts. These flagged forecasts can then be checked accordingly to minimize the risk of large forecasting errors in a labour-efficient manner.

Chapter 8

Discussion

This chapter reflects on the choices made in the Methodology and Data chapters of this report, and the conclusions that were drawn from the Results. First, Section 8.1 discusses all important assumptions and considerations, as well as points out the limitations of this research. Second, Section 8.2 proposes interesting topics that could be the focus of further research.

8.1 Assumptions, considerations and limitations

First of all, the contrastive regression models heavily rely on data regarding baseline demand: the outlier detection method needs it to calculate the promotional uplift, and the feature importance scores of baseline clean ADR are 67.5% and 60.4% for CR-CBinter and CR-CBintra, respectively. In this research, baseline demand is taken from the week in which the promotion was active. In practice however, a promotional forecast at Picnic is computed five weeks ahead, hence baseline demand also needs to be predicted five weeks ahead. This prediction is not error-free, so this indirectly adds uncertainty to the promotional demand forecast. In this research, it was a deliberate choice to use baseline demand from the promotion week instead of predicting this five weeks ahead, so a clean assessment of contrastive regression could be made without the noise from predicting the baseline demand. We assume that this noise will be negligible because baseline demand of an article in general does change much in five weeks. This assumption should be checked before the model is implemented; for example by measuring the decrease in forecasting performance for varying levels of lag in the baseline clean ADR variable.

As to the time structure of the data, the promotion month feature was included in the model to account for yearly seasonality in promotional demand. The hypothesis was that two promotions with equal features but from a different month will have different demand. The feature importance scores of 1.8% and 2.9% suggest this effect is relatively small. However, note that only six months of promotional data is included in this research, so we can not yet draw any reliable conclusions on the seasonality of promotional demand. To investigate further, at least 24 months of observations is needed. Furthermore, no trend variable is included in the models. This choice is substantiated by the results of the preliminary data analysis, which indicate that the trend in clean ADR can be attributed to manual changes in the promotion strategy of Picnic, and these changes are captured by the features. By not including this trend variable in the feature space, we inherently assume that two article promotions with the same promotional features but

from different years can be considered completely equal. In other words, we hereby assume that there are no unobserved variables that change over the years and also influence the promotional demand. This assumption is not waterproof, as it is already known that such unobserved factors do exist: for example, the layout of the promotion page in the Picnic app changes regularly, and macroeconomic changes also influence consumer behaviour and susceptibility to promotions. It should be investigated to what extent such unobserved factors exist, how large their impact is on promotional demand at Picnic, and how they can be included in the feature space.

The outlier detection method proposed in this research uses the discount-normalized lift (DNL) variable to detect outlier promotions. Here the assumption is made that outlier promotions with unconventionally low or high demand given the observed features can be detected by checking for observations that have an extreme DNL. Of course, there might also be promotions that do not have an extremely low or high DNL, but are still an outlier in a different dimension and should actually be excluded. Detecting these outliers as well was not considered within the scope of this research. Moreover, this research assumes that the outlier detection method undoubtedly improves forecasting performance and that no good leverage points are excluded. This assumption should also be tested by comparing the outcome of different models with and without the newly introduced outlier detection method.

Considering the feature selection, a method had to be set up that handles the combination of numerical and categorical features and provides an objective, comparable measure of multicollinearity for each possible feature pair. Work that tackles this problem of uniformly detecting multicollinearity in mixed-type datasets (including suitable thresholds) is limited. Hence this method combines various academic sources, resulting in multicollinearity detection based on both Spearman's correlation, eta-squared, and Cramér's V. The results of the multicollinearity analysis and following feature selection are therefore not claimed to be unambiguously correct, and should be interpreted with the above in mind.

Further, the feature importance scores for CatBoost are calculated as the average relative change in the predicted value caused by a change in the feature value. This method is selected due to its easy interpretability and computational efficiency, but it does introduce the risk that categorical features with many levels (like article category) are overrated compared to ones with little levels (like superdeal). For a more reliable result, the robustness of the importance scores can be investigated by calculating the mean absolute value of the SHapley Additive exPlanation (SHAP) values (Lundberg et al. (2018)) and comparing this to the current importance scores.

Another important decision was to use $k_1 = 5$ random neighbors to build the contrastive training, and $k_2 = 5$ nearest neighbors to compute the final forecast. The number 5 was chosen for computational feasibility and because it yielded the best results in Aguilar-Palacios et al. (2021). Hereby, we inherently assume that $k_1 = 5$ is enough for the CR-CBinter model to capture the complex dependencies between promotions from different categories, and we assume $k_2 = 5$ is enough to collect the most relevant neighbors. These are strong assumptions to make, and we do not rule out that adding more neighbors can further improve performance. Section 8.2 further elaborates on ideas for improvement of this part of the model.

Next, one should be cautious with interpreting the target in this research, which is is customer demand for grocery articles (clean ADR). As customer demand essentially is fictive and

unobservable, this quantity has no real life "observed" value. As is explained in Subsection 3.3.1, clean promotion ADR is calculated from the observed article sales and the number of customers that saw the article unavailable in the app. This means that the model target is an approximation of the article demand during a promotion, and this approximation induces an additional error to the model outcome. One could argue that, as a result, the actual forecasting performance is "only as good as the approximation of article demand". The approximation of clean ADR in Subsection 3.3.1 assumes that no customers withdrew from ordering at Picnic at all due to seeing the article in promotion unavailable in the app. Also, a 75% conversion rate is used for the share of customers that adds an article to their basket and ends up actually ordering it. Although this percentage is based on empirical analysis at Picnic, it still adds an additional source of uncertainty in computing the target variable clean ADR. It should be investigated how large this uncertainty actually is, and whether it is the same across different subsets of the data.

The comparison between the inter- and intra-category trained contrastive regressor also requires nuance. First of all, the results show that the CR-CBinter model assigns a 5.1% importance score to the feature article category. This can be roughly interpreted as saying: on average, 5.1% of the difference in demand between two promotions can be attributed to their article category. This is expected to be caused by the fact that categories can have different levels of baseline and promotional demand in general, but also a different underlying relationship between promotional features and resulting demand. With the intra-category approach, we do not allow the model to leverage inter-category information. Instead, we account for this difference between categories by construction, namely through the training of a seperate learning algorithm for each category. A different way of looking at this is that we add an additional, pre-fixed layer to the regression tree that already distinguishes between the 21 article categories, and construct 21 seperate underlying trees afterwards. The results show that the CR-CBinter model has a higher overall forecasting accuracy than the CR-CBintra model. We differentiate between three potential reasons for this improvement: as a first potential reason, the inter-category set-up could be better because it allows to find nearest neighbors for the forecasted promotion outside its own category. This however is likely not the case, as we observe that the five nearest neighbors often come from the same category. As a second potential reason, it could be that a part of the promotions from a certain article category follows a demand pattern that is different from the rest of the category and more similar to promotions in other categories. By using inter-category training, the model can also learn this demand patterns from other categories, increasing the overall accuracy. As a third potential reason, the underlying pattern between promotional features and demand could simply be the same for different categories, hence the model performs better solely because it is trained on a larger dataset of promotions. Our expectation is that the improved performance of the CR-CBinter model is a combination of the latter two, and that the reason can differ per category. On the one hand for example, the categories "Baby & Child" (148 observations) and "Fruit" (110 observations) show a strong decrease in WAPE after extending from intra- to inter-category training (from 61.9% to 27.1% and from 36.5% to 25.5%, respectively), which might indicate that the separate models in the intra-category approach suffered from having too little training data. On the other hand, the WAPE for the category "Potatoes & vegetables" (346 observations) is notably lower for intra-

category than for inter-category training (18.6% versus 24.6%, respectively) despite having fairly little training data, which shows that extending to inter-category training and increasing the training sample size does not improve performance here.

Lastly, recall that the dataset with article promotions used in this research displays an imbalance in the article category feature: the category "Drugstore" (NL: "Drogist") accounts for roughly one third of the observations, while the other 20 categories all account for between 1 to 8% of the data. The expectation is that at Picnic, this imbalance will remain and possibly grow even larger, as the assortment for the category Drugstore expands rapidly and this category heavily relies on promotions. Despite the imbalance, the inter-category trained contrastive regression model is not unproportionally biased towards producing an accurate forecast for Drugstore promotions. This is confirmed by the results in Table 5.4 of Section 5.2, showing that the WAPE for Drugstore is actually the highest amongst all categories. The reason for this is the following: the model is trained on minimizing the mean absolute error, hence it will primarily focus on reducing the relative error for popular promotions with large demand and focus less on reducing the relative error of promotions with lower demand (Figure 5.2 in Section 5.2 clearly visualises this). Because Drugstore articles generally have low baseline ADR and hence low promotional ADR, the relative error of the CR-CBinter model for this category is high compared to other categories. Interesting to mention is that the CR-CBinter still outperforms the CR-CBintra model for the Drugstore promotions: this shows that training a seperate model solely on data from Drugstore promotions does not increase accuracy, and even suggests that the forecasting performance is improved if we use one overarching training set.

8.2 Topics for further research

Regarding topics for further research, this section distinguishes between research that focuses on improving or extending the contrastive regression model, and research that expands the application area of the model beyond promotional demand forecasting in grocery retail.

8.2.1 Potential improvements to the contrastive regression model

A first point of improvement considers adding more features to the regressor. As mentioned earlier, these include promotions from competitors, additional marketing effort carried out by Picnic, and a more detailed indication of the visibility of the promotion in the store app.

Next, to identify other potential points of improvement or extensions to the contrastive regressor, the steps of the model described in Section 3.1 can be used as guidance. Note that the second step of the outlier detection method can be interpreted as a highly simplified robust regression with relative discount and baseline sales as the explanatory variables and promotional sales as the response variable. The reason for using relative discount and baseline sales is because we assume that these factors are crucial in determining promotional sales. This choice is substantiated by the findings later, displaying a joint feature importance of almost 75% for baseline sales and relative discount. Solely including these two variables was a deliberate choice to maintain high transparency in the way outliers are detected, but including more explanatory variables and performing a proper robust regression might improve the method. Good starting

points are the MM-estimator for robust linear regression introduced by Yohai (1987), or the ROUT-method for robust nonlinear regression introduced by Motulsky and Brown (2006).

An important building block of the contrastive regression model is the internal regressor that computes the feature importances forecasts the difference in clean ADR. In this research, tree-based algorithms are used due to their high accuracy and inherent ability to compute feature importances from the splitting process. As an improvement, the decision tree regressor can be replaced by more advanced regression methods, for example support vector regression (Drucker et al., 1996) or neural network regression (Specht, 1991). Note that both methods do not inherently offer feature importance calculation, but this could be solved by implementing a permutation-based method, for example as introduced by Altmann et al. (2010).

Regarding the feature importance calculation: importance scores are now calculated on an aggregated level during trainig, and hence each nearest neighbor applies the same weights to calculate the distance to the test promotion. An interesting extension would be to explore the use of forecast-specific SHAP-values to weigh the distances between the test promotion and the neighbors. As a result, each distance calculation between a test promotion and neighbor would use a different weights matrix that is based on the SHAP-values from the forecast of the difference in ADR between the two. The potential benefit is that this "local weighing" might improve forecasting accuracy. The expected cost however is that this will severely increase computational effort, as SHAP-values need to be computed for each combination of test promotion and potential nearest neighbor (hence, the tree algorithn needs to forecast the difference in demand between the test promotion and every potential neighbor in the training set).

As another possible improvement, we can explore ways to combine the inter- and intra-category set-ups. Forecasting results underline that some categories benefit from the inter-category training, while others show better forecasting performance when trained intra-category. A simple idea to combine the two approaches is to extend the process of building the contrastive regression set by selecting for example 5 random neighbors from the same category as the reference promotion, and 5 random neighbors from a different category. To allow for even more flexibility in the trainig set-up, an ensemble model can be built that consists of multiple models with different training scopes (intra-category, inter-category, and any option in-between).

Next is the nearest neighbor algorithm that finds the most similar promotions. This research uses a simple brute force approach with a fixed number of neighbors ($k_2 = 5$) and Gower's distance as a similarity measure. First, brute force calculation is computationally heavy as for every new forecast the distance between the forecasted promotion and all historical promotions needs to be calculated. To decrease computational effort for searching the neighbors, a more efficient data structure configuration method such as a K-Dimensional Tree (Bentley, 1975a) or Ball Tree (Omohundro, 1989) can be used. These methods both efficiently structure multidimensional data and hence decrease search time per forecast. K-Dimensional Trees organize the observations in hierarchical axis-aligned rectangles, where Ball Trees organize the observations in hierarchical spheres; the latter is considered computationally more efficient for number of dimensions $d \geq 3$, and therefore preferred for the article promotion dataset. Second, the choice for 5 neighbors is based on the findings by Aguilar-Palacios et al. (2021), but it is not confirmed whether this is also the best choice for the contrastive regressor proposed in this research. Fur-

ther research should evaluate the performance for varying number of nearest neighbors, for example $k = \{3, 4, 5, 6, 7, 8, 9, 10\}$. We also hypothesize that some cold-start promotions have many similar historical promotions, while others might have little to none. This substantiates the implementation of a method that does not fix the number of neighbors, but rather specifies a proximity range in which all nearest neighbors should lie, such as the radius-based neighbor searching techniques first introduced by Bentley (1975b). Third, it should be evaluated whether the use of a different hetereogeneous distance measure than Gower's distance can improve forecasting performance. Possible options are the Heterogeneous Euclidean-Overlap Metric (HEOM) or Heterogeneous Value Difference Metric (HVDM) from Wilson and Martinez (1997).

Lastly, the loss functions used in this research (MSE, WAPE and WPE) are all symmetric, meaning that they equally weigh over- and underforecasts. At Picnic however, overforecasting is sometimes preferred over underforecasting for non-fresh or non-food articles. For these articles, excess stock can be preserved and sold later, while out-of-stocks immediately cause customer dissatisfaction. This substantiates the use of an asymmetric performance metric for (at least) part of the article promotions. Two good options for an asymmetric loss function that are often used are the Weighted Least Squared Error (WLSE) and the Linear Exponential Error (LINEXE). This can be extended to an even more flexible approach where a different loss function is used for each category that fits the freshness of those articles.

8.2.2 Application of contrastive regression beyond demand forecasting

Besides improving the contrastive regression model, there are also interesting opportunities to extend its application area beyond promotional demand forecasting in grocery retail. These opportunities have in common that they involve regression to predict a quantity, suffer from cold-startness or low data availability, and require explainability of the model outcome.

A first example is predicting the performance of a startup company (e.g. annual revenue, or market capitalization after one year) based on a dataset of startups that were founded in the past. Numerical features could then include current revenue, profit margin, or total investments, and categorical features could include the specific market or industry, level of scalability, or type of technology used. In addition to the forecast, the list of similar historical startups provided by the model can be useful to learn from their successes or mistakes, or to identify synergies.

A second example involves predicting the performance of a newly launched book (e.g. copies sold in the first month) based on existing books. Numerical features could include the performance of prequels, the available marketing budget, or the number of pre-orders, and categorical features could include the genre, the reputation of the author, or the platform(s) on which it is released. Additionally, the feature importances provided by the method can help the publisher or film studio determine where to focus on to increase sales.

As a third example, consider predicting the adoption of sustainable regulation by companies based on the adoption of earlier regulation. Numerical features could include the amount of environmental impact (e.g. percentage reduction in carbon emission or energy consumption), the costs of implementation, or the long-term financial yield, and categorical features could include the type of regulation, presence of government incentives, or level of compliance required.

References

Aguilar-Palacios, C., Munoz-Romero, S. & Rojo-Alvarez, J. L. (2020). Cold-start promotional sales forecasting through gradient boosted-based contrastive explanations. *IEEE Access*, *8*, 137574-137586. doi: 10.1109/ACCESS.2020.3012032

Aguilar-Palacios, C., Munoz-Romero, S. & Rojo-Alvarez, J. L. (2021). Causal quantification of cannibalization during promotional sales in grocery retail. *IEEE Access*, *9*, 34078-34089. doi: 10.1109/ACCESS.2021.3062222

Altman, N. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*, 175. doi: 10.2307/2685209

Altmann, A., Toloşi, L., Sander, O. & Lengauer, T. (2010, 5). Permutation importance: a corrected feature importance measure. *Bioinformatics*, *26*, 1340-1347. doi: 10.1093/BIOINFORMATICS/BTQ134

Anderson, E. T. & Fox, E. J. (2019, 1). How price promotions work: A review of practice and theory. , *1*, 497-552. doi: 10.1016/BS.HEM.2019.04.006

Bentley, J. L. (1975a, 9). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, *18*, 509-517. doi: 10.1145/361002.361007

Bentley, J. L. (1975b). *A survey of techniques for fixed radius near neighbor searching* (Tech. Rep.). Stanford, CA: SLAC.

Blattberg, R. C. & Neslin, S. A. (1993, 1). Chapter 12 sales promotion models. *Handbooks in Operations Research and Management Science*, *5*, 553-609. doi: 10.1016/S0927-0507(05)80035-0

Bojer, C. S., Dukovska-Popovska, I., Christensen, F. M. M. & Steger-Jensen, K. (2019). Retail promotion forecasting: A comparison of modern approaches. *IFIP Advances in Information and Communication Technology*, *567*, 575-582. doi: 10.1007/978-3-030-29996

Chase, J., Charles W. (1994, Fall). Customer demand forecasting. *The Journal of Business Forecasting Methods Systems*, *13*(3), 2.

Chauhan, A., Prasad, A., Gupta, P., Reddy, A. P. & Saini, S. K. (2020, 4). Time series forecasting for cold-start items by learning from related items using memory networks. *The Web Conference 2020 - Companion of the World Wide Web Conference, WWW 2020*, 120-121. doi: 10.1145/3366424.3382728

Christensen, F. M. M., Solheim-Bojer, C., Dukovska-Popovska, I. & Steger-Jensen, K. (2021, 3). Developing new forecasting accuracy measure considering product's shelf life: Effect on availability and waste. *Journal of Cleaner Production*, *288*, 125594. doi: 10.1016/J.JCLEPRO.2020.125594

Cohen, J., Cohen, P., West, S. G. & Aiken, L. S. (2013). *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge.

Cooper, L. G., Baron, P., Levy, W., Swisher, M. & Gogos, P. (1999). Promocast ™: A new forecasting method for promotion planning. *Marketing Science*, *18*(3), 301–316.

Dai, Y. & Huang, J. (2021, 4). A sales forecast method for products with no historical data. *2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics, ICCCBDA 2021*, 229-233. doi: 10.1109/ICCCBDA51879.2021.9442603

Donselaar, K. H. V., Peters, J., Jong, A. D. & Broekmeulen, R. A. (2016, 2). Analysis and forecasting of demand during promotions for perishable items. *International Journal of Production Economics*, *172*, 65-75. doi: 10.1016/J.IJPE.2015.10.022

Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., ... Lautenbach, S. (2013, 1). Collinearity: a review of methods to deal with it and a simulation study evaluating their performance. *Ecography*, *36*, 27-46. doi: 10.1111/J.1600-0587.2012.07348.X

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. & Vapnik, V. (1996). Support vector regression machines. In M. Mozer, M. Jordan & T. Petsche (Eds.), *Advances in neural information processing systems* (Vol. 9). MIT Press.

Falatouri, T., Darbanian, F., Brandtner, P. & Udokwu, C. (2022, 1). Predictive analytics for demand forecasting – a comparison of sarima and lstm in retail scm. *Procedia Computer Science*, *200*, 993-1003. doi: 10.1016/J.PROCS.2022.01.298

Fatemi, Z., Huynh, M., Zheleva, E., Syed, Z. & Di, X. (2023). Mitigating cold-start forecasting using cold causal demand forecasting model. *Proceedings of ACM Conference*, *1*.

Fildes, R., Ma, S. & Kolassa, S. (2022, 10). Retail forecasting: Research and practice. *International Journal of Forecasting*, *38*, 1283-1318. doi: 10.1016/J.IJFORECAST.2019.06.004

Geurts, P., Ernst, D. & Wehenkel, L. (2006, 4). Extremely randomized trees. *Machine Learning*, *63*, 3-42. doi: 10.1007/S10994-006-6226-1/METRICS

Gower, J. C. (1971, 12). A general coefficient of similarity and some of its properties. *Biometrics*, *27*, 857. doi: 10.2307/2528823

Hubert, M. & Vandervieren, E. (2008, 8). An adjusted boxplot for skewed distributions. *Computational Statistics Data Analysis*, *52*, 5186-5201. doi: 10.1016/J.CSDA.2007.11.008

Jacovi, A., Swayamdipta, S., Ravfogel, S., Elazar, Y., Choi, Y. & Goldberg, Y. (2021). Contrastive explanations for model interpretability. *CoRR*, *abs/2103.01378*.

Kyu, L. D. (2016). Alternatives to p value: confidence interval and effect size. *Korean J Anesthesiol*, *69*(6), 555-562. doi: 10.4097/kjae.2016.69.6.555

Linardatos, P., Papastefanopoulos, V. & Kotsiantis, S. (2021). Explainable ai: A review of machine learning interpretability methods. *Entropy*, *23*(1). doi: 10.3390/e23010018

Lipton, Z. C. (2016, 6). The mythos of model interpretability. *Communications of the ACM*, *61*, 35-43. doi: 10.1145/3233231

Lundberg, S. M., Erion, G. G. & Lee, S. (2018). Consistent individualized feature attribution for tree ensembles. *CoRR*, *abs/1802.03888*.

Ma, S., Fildes, R. & Huang, T. (2016, 2). Demand forecasting with high dimensional data: The case of sku retail sales forecasting with intra- and inter-category promotional information. *European Journal of Operational Research*, *249*, 245-257. doi:

10.1016/J.EJOR.2015.08.029

Miller, T. (2018, 11). Contrastive explanation: A structural-model approach. *Knowledge Engineering Review*, *36*. doi: 10.1017/S0269888921000102

Motulsky, H. J. & Brown, R. E. (2006, 3). Detecting outliers when fitting data with nonlinear regression - a new method based on robust nonlinear regression and the false discovery rate. *BMC Bioinformatics*, *7*, 1-20. doi: 10.1186/1471-2105-7-123/COMMENTS

Muriana, C. (2017, 10). A focus on the state of the art of food waste/losses issue and suggestions for future researches. *Waste Management*, *68*, 557-570. doi: 10.1016/J.WASMAN.2017.06.047

Nadeau, C. & Bengio, Y. (2003, 9). Inference for the generalization error. *Machine Learning*, *52*, 239-281. doi: 10.1023/A:1024068626366/METRICS

Nicodemus, K. K., Malley, J. D., Strobl, C. & Ziegler, A. (2010, 2). The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC Bioinformatics*, *11*. doi: 10.1186/1471-2105-11-110

Omohundro, S. M. (1989, December). *Five balltree construction algorithms* (Tech. Rep. No. TR-89-063). International Computer Science Institute.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Picard, R. & Cook, R. (1984, September). Cross-validation of regression models. *Journal of the American Statistical Association*, *79*(387), 575–583. doi: 10.1080/01621459.1984.10478083

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V. & Gulin, A. (2017, 6). Catboost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, *2018-December*, 6638-6648.

Rea, L. & Parker, R. (1992). *Designing and conducting survey research: A comprehensive guide.* Jossey-Bass Publishers.

Snoek, J., Larochelle, H. & Adams, R. P. (2012, 6). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, *4*, 2951-2959.

Specht, D. F. (1991). A general regression neural network. *IEEE Transactions on Neural Networks*, *2*, 568-576. doi: 10.1109/72.97934

Tukey, J. W. (1977). *Exploratory data analysis.* Addison-Wesley.

Wen, R., Torkkola, K., Narayanaswamy, B. & Madeka, D. (2017, 11). A multi-horizon quantile recurrent forecaster.

Wilson, D. R. & Martinez, T. R. (1997). Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, *6*, 1-34. doi: 10.1613/jair.346

Xu, C., Wang, X., Hu, B., Zhou, D., Dong, Y., Huo, C. & Ren, W. (2021). Graph attention networks for new product sales forecasting in e-commerce. *Database Systems for Advanced Applications. DASFAA*, *12683 LNCS*, 553-565. doi: 10.1007/978-3-030-73200-$4_3$9

Yohai, V. J. (1987, 6). High breakdown-point and high efficiency robust estimates for regression. *The Annals of Statistics*, *15*, 642-656. doi: 10.1214/AOS/1176350366

Zhang, J. & Wedel, M. (2009, 4). The effectiveness of customized promotions in online and

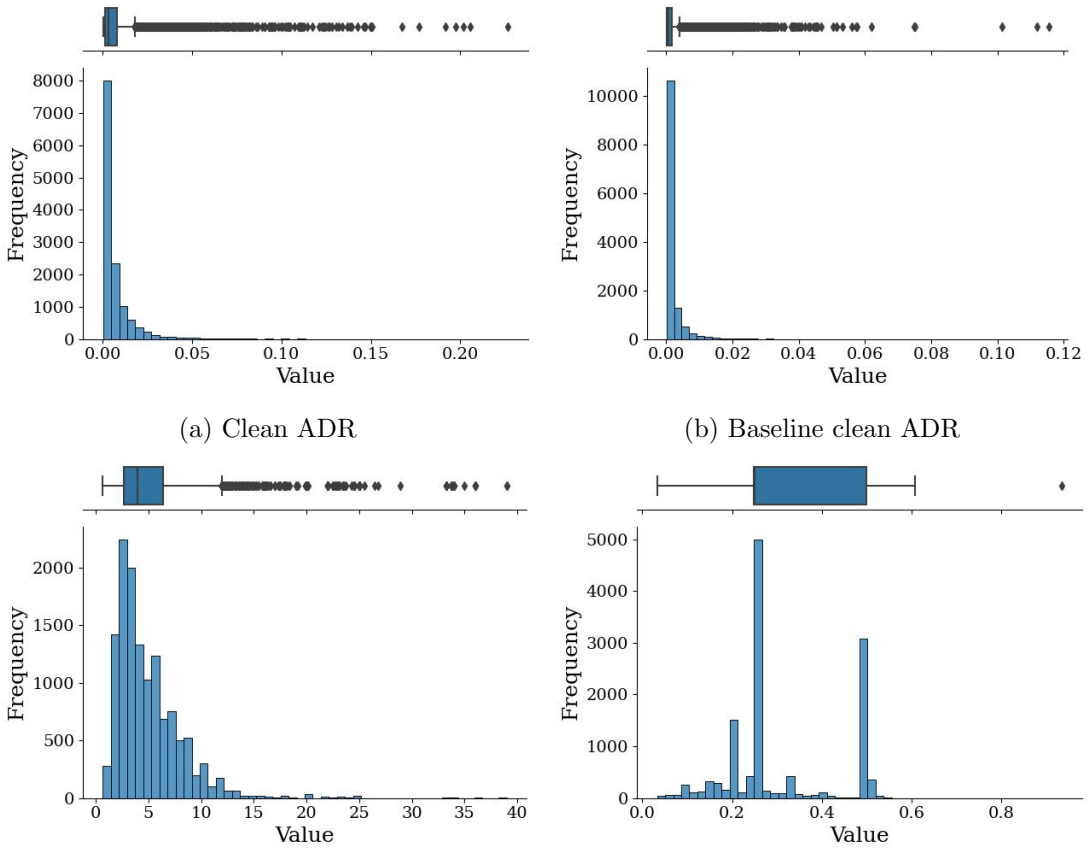
offline stores. *Journal of Marketing Research*, *46*, 190-206. doi: 10.1509/JMKR.46.2.190

Özden Gür Ali, Sayin, S., van Woensel, T. & Fransoo, J. (2009, 12). Sku demand forecasting in the presence of promotions. *Expert Systems with Applications*, *36*, 12340-12348. doi: 10.1016/J.ESWA.2009.04.052

Appendix A

Preliminary data analysis

A.1 Distribution of target and features



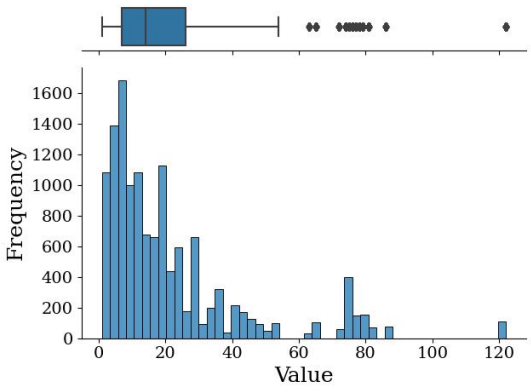
(a) Clean ADR

(b) Baseline clean ADR

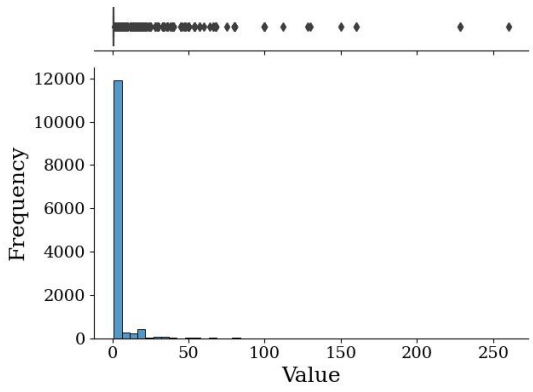
(c) Regular selling price

(d) Relative discount

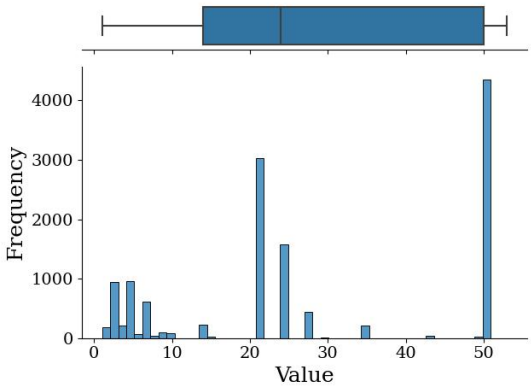
Figure A.1: Full overview of target and feature data distributions



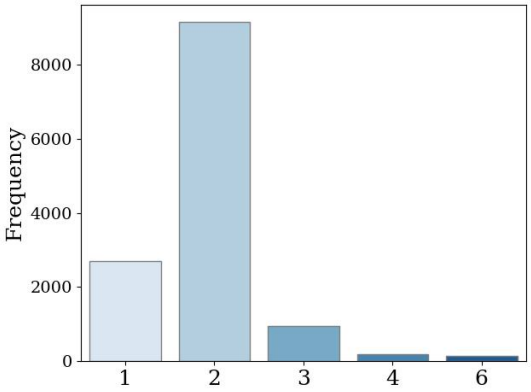
(e) Promotion group size



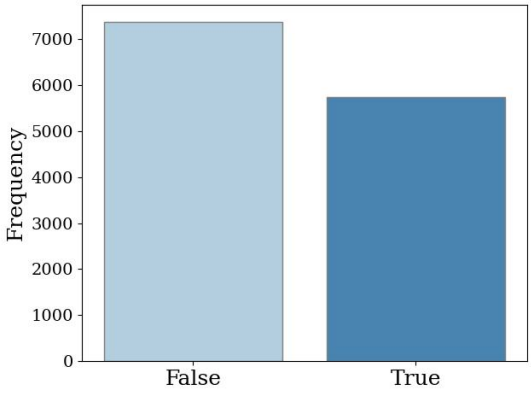
(f) Article content



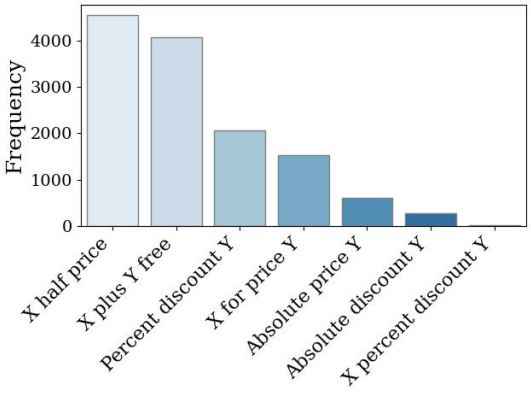
(g) Freshness days



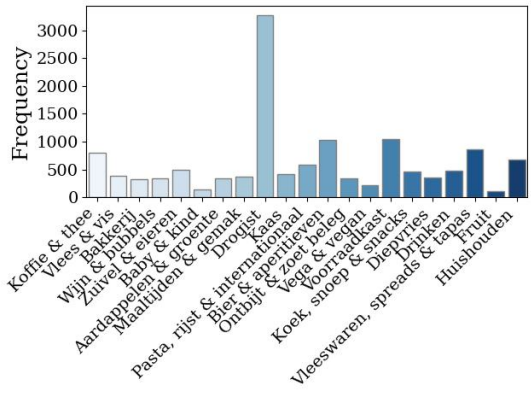
(h) Multibuy quantity



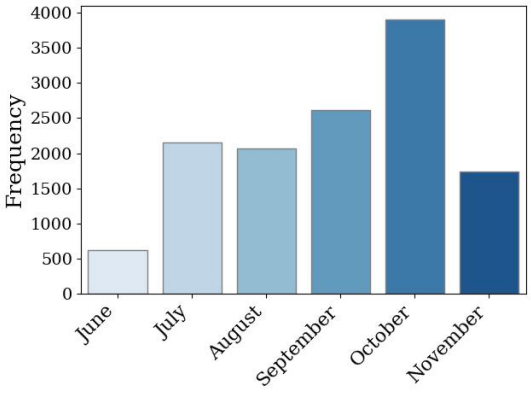
(i) Superdeal



(j) Promotion mechanism



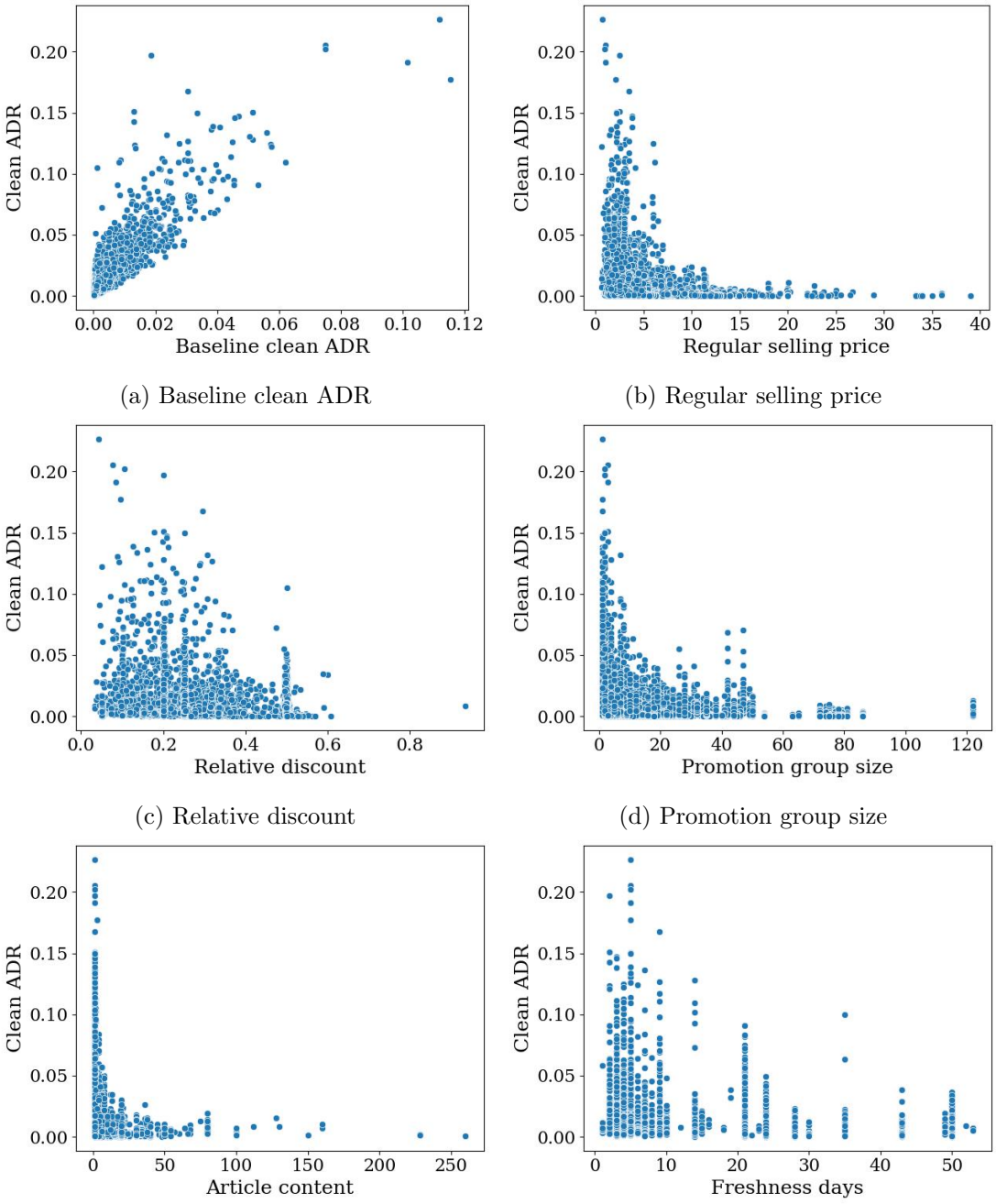
(k) Article category



(l) Promotion month

Figure A.1: Full overview of target and feature data distributions

A.2 Correlation between target and features



(a) Baseline clean ADR

(b) Regular selling price

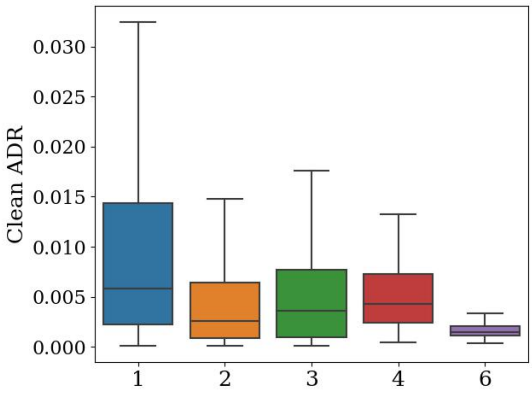
(c) Relative discount

(d) Promotion group size

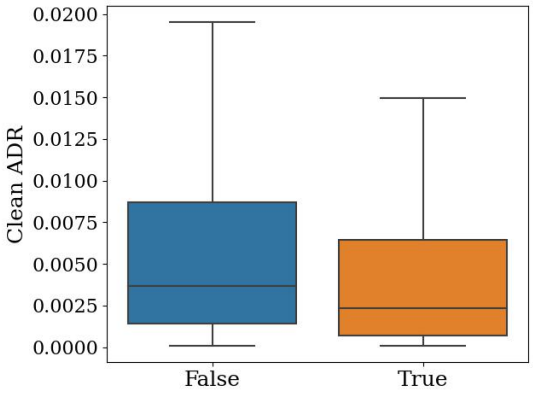
(e) Article content

(f) Freshness days

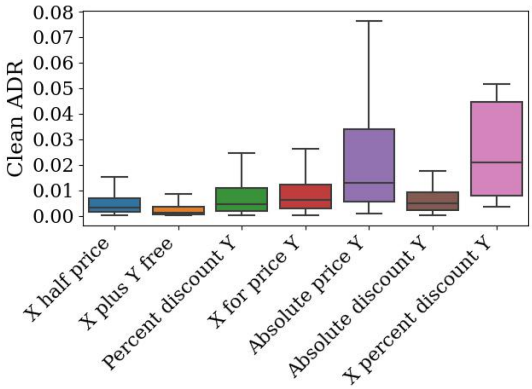
Figure A.2: Scatter plots and box plots of model target vs. features



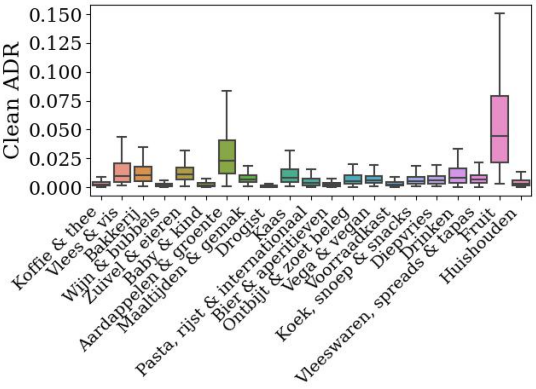
(g) Multibuy quantity



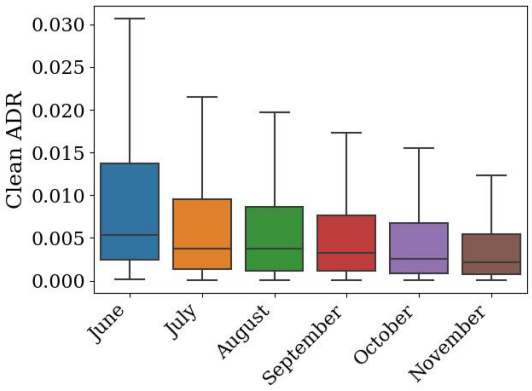
(h) Superdeal



(i) Promotion mechanism



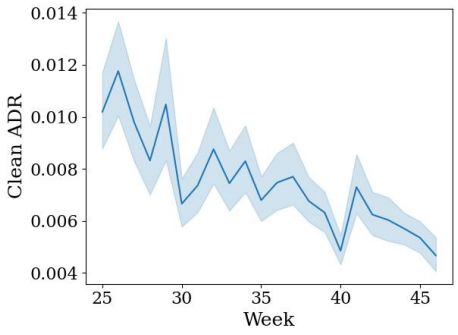
(j) Article category



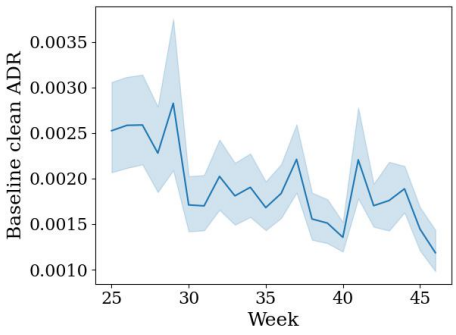
(k) Promotion month

Figure A.2: Scatter plots and box plots of model target vs. features

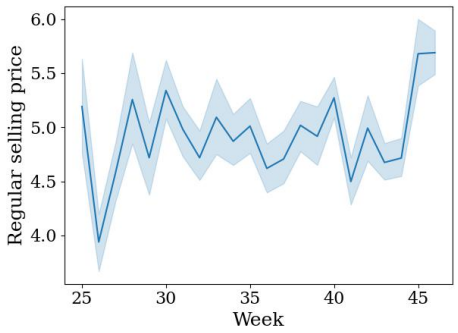
A.3 Behaviour of target and features over time



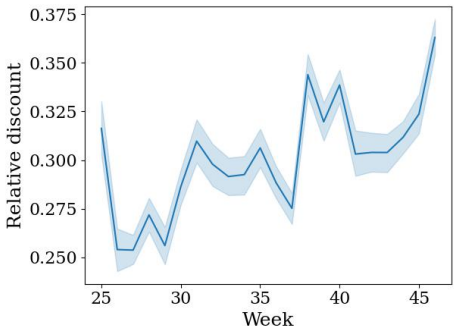
(a) Clean ADR



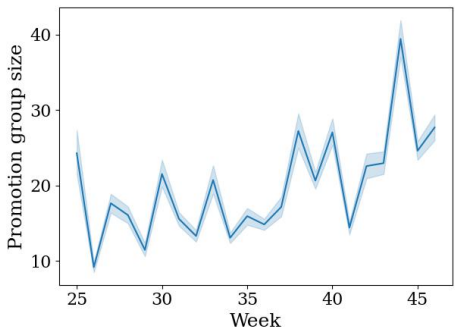
(b) Baseline clean ADR



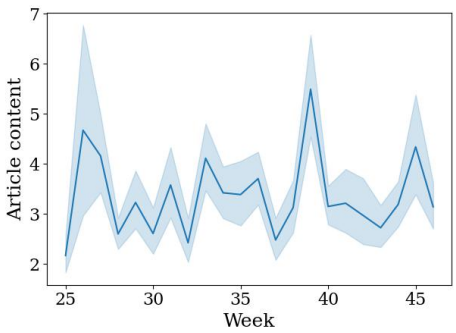
(c) Regular selling price



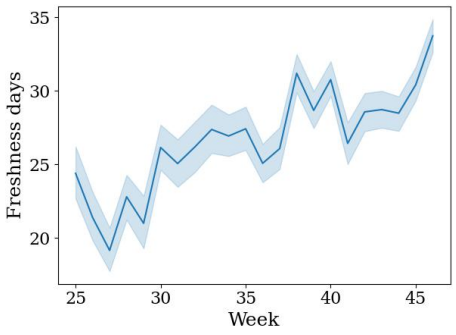
(d) Relative discount



(e) Promotion group size

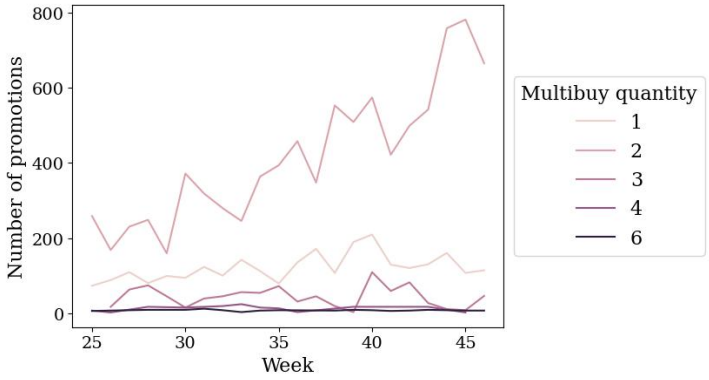


(f) Article content

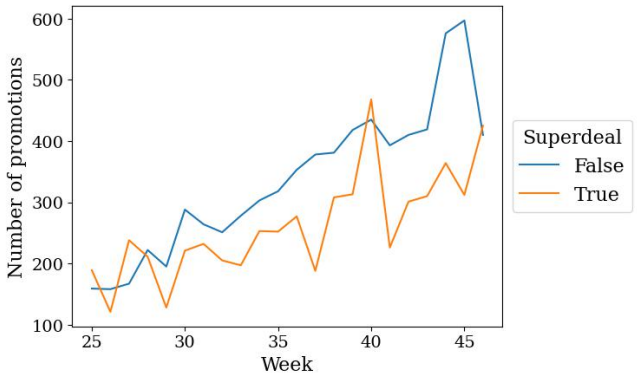


(g) Freshness days

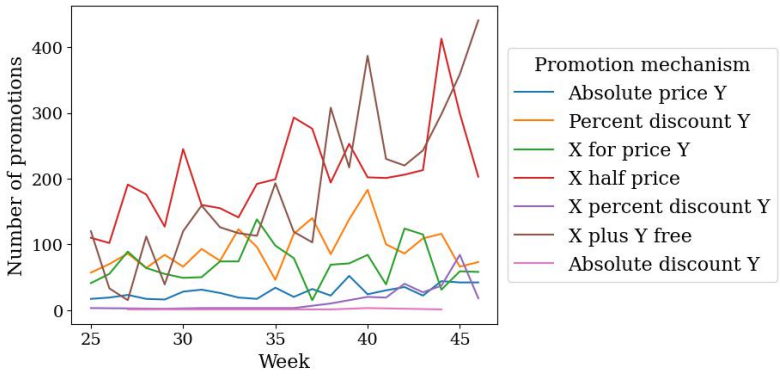
Figure A.3: Behaviour of target and features over time



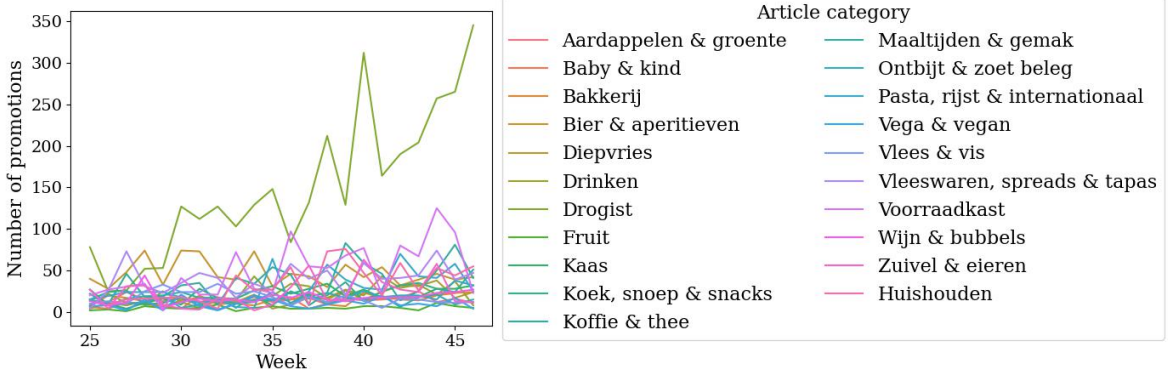
(h) Multibuy quantity



(i) Superdeal



(j) Promotion mechanism



(k) Article category

Figure A.3: Behaviour of target and features over time

Appendix B

Pseudocode for model selection

Algorithm 1 Pseudocode for generalization error estimator ${}^{n_2}_{n_1}\hat{\mu}$

Input: model selection dataset D^{select} (n samples), number of cross-validation rounds J, error metric $L(j, i)$
Output: generalization error estimate ${}^{n_2}_{n_1}\hat{\mu}$

for $j = 1 \rightarrow J$ **do**
  Randomly split D^{select} with ratio 80:20 into D_j (n_1 samples) and D_j^c (n_2 samples)
  Train model and tune hyperparameters with Bayesian optimization using D_j
  for $(X_i, y_i) \in D_j^c$ **do**
    Use X_i as input for trained model to forecast $\hat{y}_i$
    Calculate error between y_i and $\hat{y}_i$ using error metric $L(j, i)$
  end for
  Calculate average error $\hat{\mu}_j = \frac{1}{n_2} \sum_{i \in D_j^c} L(j, i)$
end for
Calculate generalization error estimate ${}^{n_2}_{n_1}\hat{\mu} = \frac{1}{J} \sum_{j=1}^{J} \hat{\mu}_j$

return ${}^{n_2}_{n_1}\hat{\mu}$

Algorithm 2 Pseudocode for variance of generalization error estimator ${}^{n_2}_{n_1'}\hat{\sigma}^2$

Input: model selection dataset D^{select} (n samples), number of rounds M, number of cross-validation rounds J, error metric $L(j, i)$
Output: variance of generalization error estimate ${}^{n_2}_{n_1'}\hat{\sigma}^2$

for $m = 1 \rightarrow M$ **do**
  Randomly split D^{select} with ratio 50:50 into D_m and D_m^c
  for $d = \{D_m, D_m^c\}$ **do**
    for $j = 1 \rightarrow J$ **do**
      Randomly split d with ratio 60:40 into $d_{m,j}$ (n_1' samples) and $d_{m,j}^c$ (n_2 samples)
      Train model and tune hyperparameters with Bayesian optimization using $d_{m,j}$
      for $(X_i, y_i) \in d_{m,j}^c$ **do**
        Use X_i as input for trained model to forecast $\hat{y}_i$
        Calculate error between y_i and $\hat{y}_i$ using error metric $L(j, i)$
      end for
      Calculate average error $\hat{\mu}_{m,j} = \frac{1}{n_2} \sum_{i \in d_{m,j}^c} L(j, i)$
    end for
    if $d = D_m$ **then**
      Calculate generalization error estimate $\hat{\mu}_{(m)} = \frac{1}{J} \sum_{j=1}^{J} \hat{\mu}_{m,j}$
    else if $d = D_m^c$ **then**
      Calculate generalization error estimate $\hat{\mu}_{(m)}^c = \frac{1}{J} \sum_{j=1}^{J} \hat{\mu}_{m,j}$
    end if
  end for
end for
Calculate variance of generalization error estimate ${}^{n_2}_{n_1'}\hat{\sigma}^2 = \frac{1}{2M} \sum_{m=1}^{M} (\hat{\mu}_{(m)} - \hat{\mu}_{(m)}^c)^2$

return ${}^{n_2}_{n_1'}\hat{\sigma}^2$

Appendix C

Example of contrastive explanations

In the two tables below, numbers regarding ADR are scaled by factor 10^4 for readability.

Feature	Importance (%)	NN1	NN2	NN3	NN4	NN5	Test promotion
Baseline ADR	67.5	5.08	5.04	4.49	4.01	4.15	4.95
Selling price	3.0	5.86	5.99	5.49	5.49	6.98	7.19
Relative discount	9.5	0.50	0.50	0.50	0.50	0.50	0.33
Group size	6.9	7	7	8	8	7	8
Article content	1.2	1	1	1	1	1	1
Freshness days	1.0	50	50	50	50	50	50
Multibuy qty.	0.4	2	2	2	2	2	3
Superdeal	1.8	Yes	Yes	Yes	Yes	Yes	Yes
Mechanism	1.8	X+Y	X+Y	X+Y	X+Y	X+Y	X+Y
Category	5.1	Drug	Drug	Drug	Drug	Drug	Drug
Month	1.8	Sept	Sept	Sept	Sept	Sept	Oct
Actual observed ADR		58.74	99.70	59.70	36.12	50.88	
Forecasted difference in ADR		-29.03	-48.69	-26.89	-14.01	-20.62	
Forecasted ADR		29.71	51.01	32.81	22.11	30.26	
Weight		53.4	53.0	53.0	52.7	52.5	
Weighted forecasted ADR							33.18
Actual observed ADR							6.74

Table C.1: Contrastive explanation for forecast with large positive relative error

Feature	Importance (%)	NN1	NN2	NN3	NN4	NN5	Test promo
Baseline ADR	67.5	1.73	1.58	1.54	1.53	1.51	1.72
Selling price	3.0	6.99	6.99	6.95	6.95	6.95	6.99
Relative discount	9.5	0.50	0.50	0.50	0.50	0.50	0.50
Group size	6.9	44	44	44	44	44	44
Article content	1.2	1	1	1	1	1	1
Freshness days	1.0	50	50	50	50	50	50
Multibuy qty.	0.4	2	2	2	2	2	2
Superdeal	1.8	Yes	Yes	Yes	Yes	Yes	Yes
Mechanism	1.8	X+Y	X+Y	X+Y	X+Y	X+Y	X+Y
Category	5.1	Drug	Drug	Drug	Drug	Drug	Drug
Month	1.8	Sept	Sept	Sept	Sept	Sept	Nov
Actual observed ADR		3.86	4.21	2.67	4.30	2.85	
Forecasted difference in ADR		-0.68	-0.35	-0.27	-0.41	-0.27	
Forecasted ADR		3.18	3.86	2.41	3.89	2.59	
Weight		506.8	483.4	476.9	475.6	471.8	
Weighted forecasted ADR							3.19
Actual observed ADR							34.08

Table C.2: Contrastive explanation for forecast with large negative relative error

Appendix D

Additional programming code files

This appendix contains a short description of the six programming code files used in this research.

File 1: contrastiveRegressor.py

Contains the class that is used for the contrastive regression model. Methods are included to train the model, get the feature importances, update the pool of potential neighbours, predict test observations, retrieve the results, and get a post-hoc contrastive explanation.

File 2: weightedNearestNeighbourRegressor.py

Python-file containing the class that is used for the weighted nearest neighbour regression model. Methods are included to train the model, get feature importances, update the pool of potential neighbours, predict test observations, retrieve results, and get a post-hoc explanation.

File 3: promotionOutlierDetection.py

Python-file containing the function that applies the outlier detection method as described in this research to a dataset of article promotions.

File 4: gowerDistanceCyc.py

Python-file containing the functions that calculate Gower's distance between two data instances that contain numerical, categorical, and cyclical variables.

File 5: modelSelection.ipynb

Jupyter Notebook-file containing the model selection procedure as described in this research. This includes data preparation and a loop that trains and tests the four candidate models and saves the forecasting results.

File 6: modelEvaluation.ipynb

Jupyter Notebook-file containing the final model evaluation procedure as described in this research. This includes data preparation and a loop that trains and tests the seven models and saves the forecasting results and feature importances.