ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Ranking Products for Relevancy for Promotions in Terms of Sales Uplift through Bayesian Forecasting

Master Thesis Econometrics and Management Science:
Business Analytics and Quantitative Marketing

Ties Maasdam - 457787MM
Supervisor: Richard Paap
Second assessor: Dennis Fok
Company Supervisor: Georgi Kokotanekov

August 30, 2022

Abstract

With online retail platform holders increasing massively in size during the COVID-19 pandemic, companies are increasingly nudged into putting their products on these marketplaces to keep up with competition. As a result platform holders have an ever increasing amount of data available to estimate consumer demand with.

In this thesis we research the promotion relevance score of products. Given past promotional and selling data, the consumer relevancy of products for promotions are rated through forecasting methods. We implement multiple models; a Bayesian based Product-Specific Tobit Sales Regression model and a Multi-Step Hybrid Regression model, of which the latter is used in a real-world experiment. Both methods are implemented using cloud-computing to deal with the big database and evaluated using measures, such as the Mean Squared Error and computational time.

The results, applications and recommendations are shared for future usage. In brief, we find daily predictions are best suited for the problem given the data and the Bayesian algorithm is recommended when it is computationally efficient implemented.

Keywords: Assortment Improvement, Bayesian modelling, Tobit, Regression, Cloud Computing

Contents

1 Introduction

Bol.com started as a single online retailer but has since then grown to be a platform owner on which thousands of retailers sell their products. This results into an assortment of millions of products, all offered on their online selling platform. The platform consists of a website and corresponding application for phones. Promoted products on the platform have increased visibility for customers through extra sales pages, filters in search results and advertisements over the web. We define Bol.com as the platform owner and the retailer Bol.com as Retailer 1 (R1). R1 used to be the only one able to create promotional campaigns on the platform and partners could join these on approval of R1. This has changed since September 2020, now partner retailers are free to put their own assortment on promotion outside of the campaigns created by Bol.com as long as they fit the requirements of the so called Self-Serviced Promotions (SSP).

Analysis shows that, with about 100.000 different products in SSP by partners in the last quarter of 2020, over 50% did not sell even once during the promotion, and most products promoted have a low uplift. Uplift is defined as the increase of demand during a promotion of the product compared to the period before the promotion, due to visibility increase on and off the platform and the offer price going down. We further define 'irrelevant products' for promotions as products with low demand and no uplift during promotions. On the contrary, products that have high demand or uplift are deemed 'relevant products' for promotions.

Bol.com has 3 pillars on why they want to see more relevant products in SSP. Firstly from a customer perspective, the platform wants its brand image to be 'the store for everyone', but the brand perception may be perceived as out of touch when the majority of products they highlight are not bought at all. Additionally, these irrelevant products in SSP take visibility from more relevant products. Secondly from a partner perspective, Bol.com wants to be the most attractive platform to run promotions on. This is accomplished by creating insights for partners about what products are relevant for promotion. Lastly as a platform, they improve customer loyalty by having new and interesting promotions introduced regularly.

Recently, Bol.com has started to give insights to partners about what products in their assortment are relevant for promotion through a business rule driven model. This has already led to a significant decrease of irrelevant products in SSP and an increase in revenue. The model which creates the insights was not created for promotional relevancy but for marketing

purposes, therefore this model does not use sales uplift at all. In this research, we will create a data-driven model to provide better insights to partners on what products are relevant for promotions where the sales and sales uplift are deemed the dependent variable.

Uplift is seen as the key performance indicator (KPI) of the relevance of products for promotion at Bol.com, therefore we need to accurately estimate the impact of promotions. We note that we cannot take sales directly as demand, since stockouts sometimes occur during promotions, which could lead to underestimating feature importance for sales uplift and in extension the relevancy of the product for promotion. Therefore, we note that the data used in this research consists of the sales data rather than the demand data generating process. This makes it crucial for us to be able to make estimates on the demand to make proper forecasts on promotion performance.

Promotions from R1 fall into two different groups: first, a more classical kind of promotion often used for perishable goods and fashion. For example, in this kind of promotions a high inventory of products at the end of a season leads to promotions. Secondly the promotions most created by R1, these promotions are held to increase visibility for the platform itself. In short, R1 buys extra stock of a product to lead customers to the platform. These promotions are held mostly for seasonal products, such as Christmas products during the holiday period. Additionally, these promotions are also created for other categories of products like electronics and other non-degradable products. Similarly, Bol.com wants promotions from partners to drive customers to the platform. An advantage of the promotions created by partners is that Bol.com does not bear the cost typically seen from stock level inventory management for the assortment of their partners. As the focus of this research is specifically on the relevancy of products of partners, we will not investigate current inventory levels as initiator for promotions. Instead we consider the relevancy of products for promotion from a consumer demand perspective and give insight to partners about consumer relevancy of products for promotion independent of their stock level. As a result, the relevancy of the research is twofold. One is the application of how to deal with censored data for forecasting in a new setting, and we will describe how to translate the results directly into a score driven system to give as insight to managers of the relevance of their assortment for promotions.

In this research we will be answering multiple questions. First and foremost we will delve into the following research question: *"How can we rank consumer relevancy of products for promotion?"*. This can be divided into multiple sub-questions: *How can we use sales data to*

estimate the effect of marketing mix variables on sales?", *"How can we adapt the sales model to deal with stockouts?", "How do we use parameter estimates to make forecasts about relevancy of products for promotions?"* and *"What methods should we use to translate the forecasts of consumer demand into a ranking of products?"*. Here we split the research question up into three parts; First, how to model an approximation on the demand data generating process and what limitations we should set to get feature importance estimates. Secondly, we consider the parameters estimates and use these to create a forecast of consumer demand for the products. Last, we will use the forecasts of consumer demand to model the relevancy of products for promotion as an insight to managers. To answer these research questions we propose the usage of a Bayesian model and a two-step regression model. Of which the latter shall be used as a baseline model and tested through a real-world empirical experiment. We use multiple datasets consisting of the products sold within Bol.com to measure the performance of the forecasting models. Subsequently, the performance of the models is measured based on multiple metrics.

Our finding suggest that Bayesian model outperforms the baseline model on all accounts except computation time by a large margin. The worse performance of the baseline model is caused by the fact that we have too few proper data points to create consistent estimates with. As a result, for the given data daily predictive models are recommended. Furthermore, we recommend the further development of the Bayesian model to make use of more optimised computations and parallel computing, such that, it can power the ranking of products for Bol.com on a daily basis for the whole assortment.

The remainder of this report is structured as follows: We provide an overview of relevant literature in Section 2. In Section 3 we discuss multiple scores used at Bol.com relevant for this research. In Section 4 the data used in this research is presented. The methodology is presented in Section 5. The implementation details and the results of the models are presented in Section 6, we provide conclusions from the results in Section 7. Finally, in Section 8 we discuss the limitations of this research and provide the possibilities for further research.

2 Literature Overview

In this section we discuss the relevant literature for this research. We note there has been done a lot of research in Operational Research that goes into computing the optimal moment for promotional activity based on the stock level of the assortment (Ma et al. (2016)), whereas the focus of this research is the relevancy of products for promotion independent of the stock levels.

Peinkofer et al. (2015) propose a framework to explain the effect of price promotions on the expectations of consumers and reactions to stockout in ecommerce. This framework is based on the expectation-disconfirmation theory. Their findings counterintuitively suggest that consumers are more dissatisfied with non-promoted products having a stockout than price promoted products. From here we note that having a limited stock for a promotion influences the relevancy of the product for promotions to a lesser degree. As we will not incorporate stock levels to determine relevancy of products for promotions into this research, we will assume that stock effects are negligible for the relevancy of products from a consumer perspective.

There are multiple ways to observe the relevancy of products for promotion. One of these is through the impulse response of promotions (Pesaran & Shin, 1998). Here the uplift is taken of the promotional activity as well as the post-promotion dip to find the dynamic effect of price changes. For example, Fok et al. (2006) propose a Hierarchical Bayes Error Correction Model to predict the long term effects of promotions. Similarly, Foekens et al. (1998) propose to use dynamic brand sales models with varying parameter importance to estimate the dynamic effects of sales promotions. Macé & Neslin (2004) looked into variables pre-promotion and post-promotion dip and Nijs et al. (2018) propose to incorporate competitor data for the category demand during promotions to estimate the short and long term effect of promotions. Since most of these models are created for Fast Moving Consumer Goods (FMCG), often the assumption is made that products have no stockouts. However, this assumption is not realistic for this research. Therefore, we just focus on the alternative, the relevancy of products based just on the uplift during a promotion, as this also aligns with the KPIs set by Bol.com.

To create forecasts on the uplift of promotion we make use of the uplift as the additional sales of a product during a promotion compared to when no promotion would have taken place. Here we note that Bol.com has as policy that products have the same price for all customers, thus only estimates on the uplift can be created as we either only have regular sales or promotional sales data for a given period. One way to find estimates of uplift is through a hybrid 2-step model as proposed by Abolghasemi et al. (2020). First, a baseline sales forecasting model is created, afterwards the total sales during promotions are taken minus the predicted baselines sales. This is regressed against variables containing the promotional activity, price and an intercept. This 2-step approach does not account for the uncertainty in the parameters from the first stage in the second stage and the other way around. This leads to underestimation of the uncertainty

of the parameters in the second stage when the sample data is limited. We consider such an approach as a benchmark seen in Section 5.2.

In addition to forecasting the uplift we also consider the impact of stockouts on parameter estimates in a sales model. As shown by Conlon & Mortimer (2013), when one does not take care of stockouts there will be bias in the estimates for demand, presented through a discrete choice model with Expectation Maximisation (EM). To account for the censorship, multiple techniques have been used. Ozhegov & Teterina (2018) propose the usage of machine learning methods to perform an ensemble method to predict censored demand. Ozhegov & Teterina (2018) provide results for the different degrees of censorship for least squares, Ridge and Lasso regression and a Random Forest model. The results show that machine learning algorithms have bias corrected estimates for price elasticity on demand similar to econometric models.

In the frequentist statistic field there have been multiple papers that deal with censored demand data through EM. For a vending machine problem with censored demand, Anupindi et al. (1998) propose to use EM to take care of missing stock data in their periodical inventory data. The assumption is made that products which are substitutable have Poisson demand and this results into their finding that only a part of sales is lost when stockouts occur. Additionally, Vulcano et al. (2012) use a nonhomogeneous Poisson model for arrivals combined with a multinomial logit (MNL) choice model to estimate the demand if all products were always stocked. These estimates are found through an efficient model using EM techniques. Another usage of EM is seen in the use of multivariate Gaussian mixture models with truncated and censored data (Lee & Scott, 2012). For a low amount of product, Stefanescu (2009) propose the usage of a multivariate sales model that uses the interaction between multiple products and deals with stockouts through EM.

Due to our large data set with a high number of products but limited data per product it is logical to look into Bayesian modelling. We formulate the problem at hand within the context of a Tobit censored regression model (Tobin, 1958), here we can create parameter estimators through Gibbs sampling (Chib, 1992). The use of Tobit models has been done in different context before, Cornick et al. (1994) propose a Multivariate Tobit to model milk expenditures. Furthermore, Wei & Tanner (1990) put forth a methodology for a censored regression with generalised log-gamma errors. Jain et al. (2015) propose to use the timing of sales in Bayesian way to forecast demand, they apply it to a multiperiod newsvendor problem with stockouts.

To make use of the sparsity of the data, there has been multiple researches into Bayesian latent factor models (Agarwal & Chen (2009); Bernardo et al. (2003); Bhattacharya & Dunson (2011)).

To allow for the uncertainty in the parameters Hierarchical Bayesian modelling is introduced. Kim et al. (2002) propose the usage of a hierarchical model for household-specific parameters. Within the limitations of the model for pricing policies, assortment, and limited shelf space, they find a subset of varieties of products that can be displayed or purchased. Hierarchical Bayesian has been used to model primary and secondary demand on an individual specific level (Arora et al., 1998); additionally, it has been used to the determine customer arrival rate and choice from sales and stock data (Letham et al., 2016).

3 Scores within Bol.com

In this section we discuss the different scores that are used within Bol.com to rank products in respect to the relevancy of products for promotions and the sales forecasting. In Section 3.1 we present the Relevance Score, the score that is currently implemented as an insight to partners and explain why it needs improvements. In Section 3.2 we discuss the forecasting algorithm which is currently implemented by R1 to make predictions on the demand of products.

3.1 Relevance Score

In this section we discuss the Relevance Score. We explain how the score was created and how it is put together in Section 3.1.1. Furthermore, in Section 3.1.2 we perform an analysis on the score to display where the main points of improvement are to be made for creating a new score for the relevance of products for promotion.

3.1.1 Relevance Score Explanation

The Relevance Score is a score employed by R1 to determine the relevance of products for promotions. The score is currently used in a few settings on the platform. For example, it is used for Search Engine Advertisement bidding, Banner advertising on the platform and product selection for promotions. Additionally, the Relevance Score is currently shown to partners for each product of their own assortment, this is given as an insight for the product relevance of products for promotions. The score was originally introduced as a score for the relevancy of products for marketing applications. After it was repurposed for the relevancy of products for promotion, Bol.com realised that the score was flawed for the set goals. This is due to the fact

that the score was mostly based on the performance of a product at a given moment, but not based on the performance of products *during* promotion.

The Relevance Score is created on a product basis per chunk. A chunk is defined as a product group on the platform. As an example, televisions are a chunk and so are Dutch books. On Bol.com there are about 5000 chunks in total. The chunks vary in size from a dozen products to a several million. As seen in Figure 1 the Relevance Score is created by scoring each product per feature on a scale from 0 to 100. For every chunk, data of the past 28 days is collected. For example, the product with the highest number of views would get the full 100 score and the least seen product a 0. The rest of the products are then given a score based on the visits compared to the other products. This is performed for each of the features: visits to the product page, sales, add to wishlist, the product age, the review score, and the number of returns. Thereafter, the scores are aggregated using a weighting rule for the different features to create the Relevance Score.

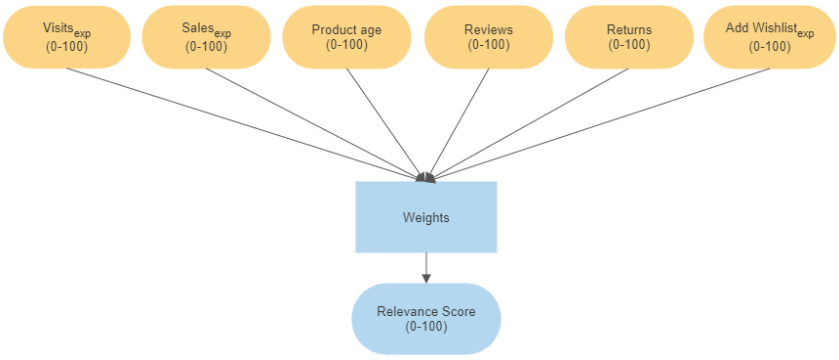

Figure 1: The Relevance score is created using a scaled function of the feature importance of the product given the chunk.

The score is created per chunk, as one scale for the whole assortment does not work well, this is due to characteristics between chunks being vastly different. This has to do with that characteristics per chunk can have different meanings. For example, in the fashion product group products are often bought in multiple sizes and returned when the products do not fit the buyer, whereas in other chunks returns happen less often. Putting the different kinds of products on the same axis would thus not result into consistent scores. The variables *clicks*, *add to wishlist* and *sales* are exponentially weighted, such that, the clicks of yesterday weigh more than the day before. This was done to pick up new patterns sooner. The reviews are only considered when products have 5 reviews or more, otherwise they were not enough data points and get scored 0.

Variable selection is based on the following: the variable *visits* is seen as the best indicators for conversion. As for the other features, based on business rules and necessities the other variables were chosen. Similarly, the feature importance is also based on business logic. There are different features like *add to cart* not used because it was misused by partners to increase their product rating and it is highly correlated with sales. Additionally, customer service cases were not used, as most products do not have customer service cases and the products with customer service cases have a high correlation with sales. Instead they use returns as they consider it a feature with a stronger signal as customers are incentivised to return the products when they are not satisfied, since customers get their money back when they return the product. Similarly there was a demand for the usage of the Customer Lifetime Value (CLV) by the business side of Bol.com, but it was not used as the CLV was not created per chunk, instead it was based on a different score group definition used by the business side of Bol.com which does not overlap well with chunks. The Customer Lifetime Value is explained in more detail in Appendix A.1.

3.1.2 Analysis Relevance Score

To measure the overall performance of the Relevance Score, we need to get a grasp of the current predictive performance of the score. We first note that the Relevance Score currently has 80% of products fall within the 5 to 6 score range and from there the number of products within each score group decreases exponentially, as can be seen in Figure 2 (note the scaling in the y-axis). As a lot of products do not sell during promotions, the distribution aligns with our expectations. Additionally, the highest rated product is 87, thus not the full score range is used.

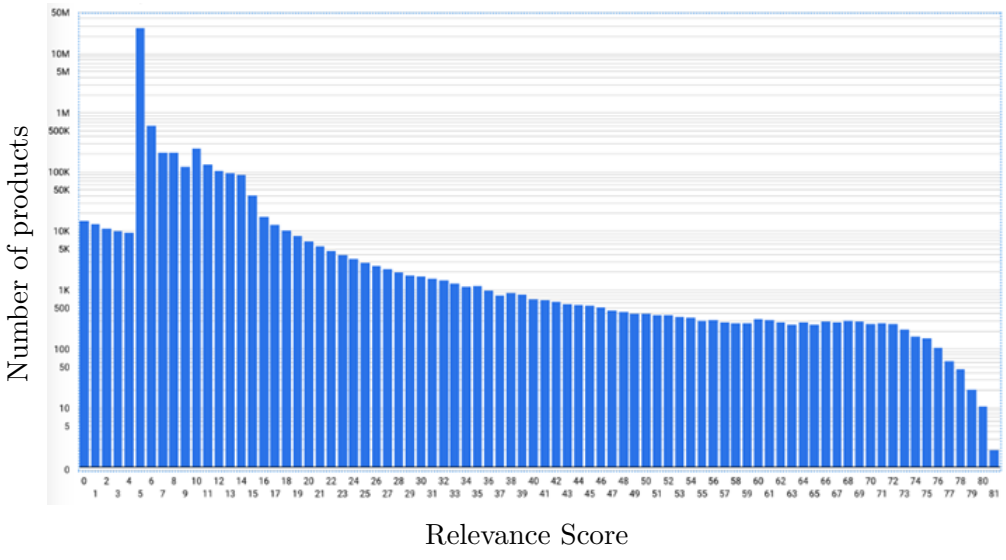

Figure 2: Relevance Score against the number of products in each score group.

In Figure 3 we look into the correlation between the current Relevance score and the average daily sales of a product during SSP. We do not see a clear link between the current Relevance Score and the average sales during SSP's, this is supported by a correlation of 0.28. This is in line with the expectation that the Relevance Score cannot predict the uplift during promotions, thus as products outperform their expected daily sales during promotions over the whole score line, the correlation is lowered.

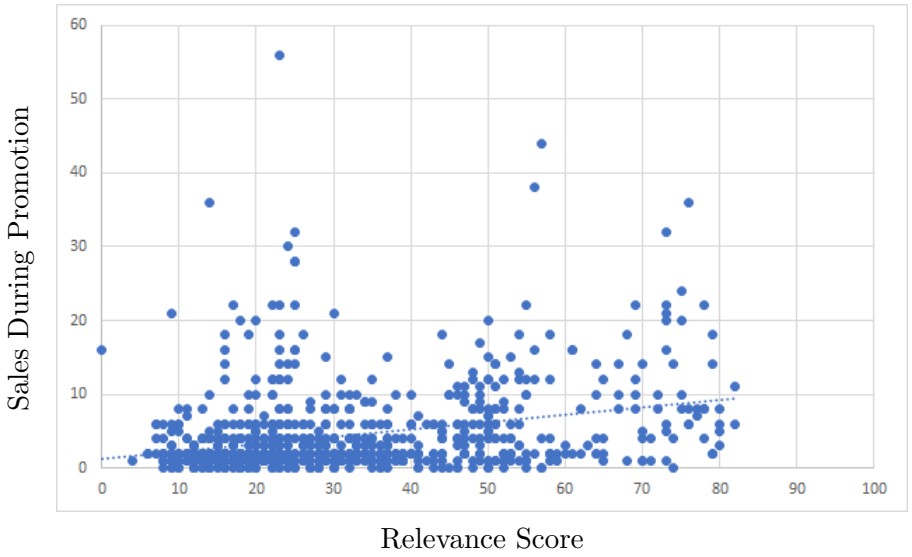

Figure 3: Relevance Score against the average daily sales of SSP during the last quarter of 2020.

Similarly, when considering the median sold of products during SSP, we see that the median sales per product are 0 for all products below products of the 75 score range. This indicates that the Relevance Score does not have great forecasting performances in terms of predicting the relevancy of products for promotion.

Central to the need for this research is the low forecasting performance of the Relevance score. After a survey done over the partners of R1, there was an indication that only less than 10% of partners keep the score in mind for creating promotions. To increase this percentage, R1 needs a new score that can give better predictions for products during promotion.

3.2 Forecasting and Seasonality

To perform forecasting of products, Bol.com uses a LightGBM model (Ke et al., 2017) combined with the forecasts of a Ridge regression model (Marquardt & Snee, 1975). Due to confidentiality reasons of Bol.com we cannot go into detail of the forecasting model. The LightGBM makes tree splits on a gradient boosting method, which avoids overfitting by continuously cycling between

the training and validation data set. The data used for the models consists of the average sales of the product in the last 14, 28 and 56 days, and to make seasonality forecast additional data is used based on the last 3 years of data.

The LightGBM and Ridge model create predictions on a weekly basis based on the sales of the last 14, 28 and 56 days. When a stockout occurs the data is not included within the calculation. The models are optimised to realise the least Mean Squared Error (MSE) for sales. We go more into detail of the MSE in Section 5.5.1. As a result, Bol.com finds that the optimal prediction of sales over the whole assortment realises a tweedie distribution. A tweedie distribution is a special case of the exponential distribution, in this case there is an extra mass point at zero, while there is also a typical exponential curve realised. This distribution falls in line with the expectation that the majority of products do not sell and the products that do sell realise an exponential curve. The predictions made by the models are combined in such a fashion that results in to $\mathit{Prediction} = 0.3\ \mathit{Ridge}\ +\ 0.7\ \mathit{LightGBM}$. The weights for the models are based on empirical results which showed that these ratios were the best. There have already been endeavours to find dynamically optimal weights for product groups. This alternative model improved the predictions minimally, but the gains of this approach did not warrant the costs to run it.

To account for the seasonality in sales over products an additional modifier is used on the *Prediction*, to receive the final predicted demand: B. To compute B, the estimate of the predicted number of sales are taken from the *Prediction*. Next, the *Prediction* is then modified based on the percentage of sales from the current week in comparison to the sales of desired period to be forecast from the years beforehand and the total sales. These seasonality forecasts are done on chunk level. This was a requirement to have sufficient data for consistent estimates. When a chunk has not enough data, due to there not being enough products, a bigger domain is taken. This results into that for specific chunks, a subgroup of chunks can be taken. The forecasts are made up to 12 weeks ahead of time. A downside of forecasting seasonality just based on the sales of previous years is that predictions for product groups can be lousy if seasonality is not the direct cause of fluctuations. For example, products which are weather dependent, such as heaters and swimming pools, are hard to predict based on this method. These products are mostly based on weather conditions, with which seasonality has correlation but not causation. In this research we use the one-week ahead forecasts, B, as one of the main inputs of a benchmark model used in the experiment as seen in Section 5.2.

4 Data

In this section we provide a description of the data used in this research. The empirical data comes forth from the databases provided by Bol.com. Here we consider all products that were available for purchase on the 30th of June 2021 on the platform. Bol.com has seen exponential growth since its inception and the promotional activities have changed with that growth. Additionally, the behaviour of consumers has changed substantially since the introduction of Covid-19 into the Netherlands, making data before March 2020 less representative. Furthermore, in this research we only look at the data of the Netherlands, as behaviour and prices of products differ significantly for Belgium for Bol.com. Additionally we look into the performance of SSPs, thus, we opt to only use data since the introduction of the SSPs on the platform of September 2020. As a result, we research the period from the 1st of September 2020 until the 30th of June 2021.

For this research we collect for every product the sales data, promotional data, and underlying characteristics of products. In order to achieve this, we combine multiple datasets kept by Bol.com through SQL. In this research we have multiple models with different requirements, resulting into two different datasets. In Section 4.1 we describe how we create the dataset for the Bayesian model. The dataset for the 2-step baseline Model is constructed in Section 4.2.

4.1 Dataset Bayesian Model

In this section we elaborate on how the dataset is created, in terms of variables, which is used for the Bayesian model as seen in Section 5.1. The data all comes forth from the database kept by Bol.com. For every product we collect data over a period of 10 months starting from the 1st of September 2020 until the 30th of June 2021. The starting date is chosen as the first day the SSP could be created by partners. For every product we collect the following data: let our dependent variable be daily sales of the product given in *sales*, then let *product_id* be the unique product identifier with specific chunk identifier as *chunk_id*, and *date* provides the date of the entry. Furthermore we have categorical variable *day_of_week*, which signifies the day of the week. The amount of clicks the product page gets on the given and the previous date are contained in *clicks*. The best offer price is contained in *offer*. Here we define the best offer as the offer selected to show to customers in the search engine by Bol.com. Since multiple retailers can sell the same product, the best offer is determined by Bol.com given certain factors, such as, price, review score of the retailer and delivery time, here we note that R1 is not per definition the best offer for a product if they offer it. Furthermore, *promos* contains information on what kind of promotions are hold on the product, in terms of display visibility and discount.

We make transformations to the data to create a dataset which is usable for the model. The dataset up is made up per *product_id*, and we only have the data points for the dates when there is an *offer*. The offer data is recorded in the database at a set time during the night. Any changes to the offers that happened during the day are not recorded in the database. This limitation is due that using the database recording every change concerning offers over the day would have resulted into an increase of a tenfold of terabytes of the data. Which is out of scope of this research. The data points of the Bol.com database concerning offers are recorded at midnight. As a result we could miss a new product and the product is sold within the same day, therefore, we use the selling price of the product as a replacement for the best offer price. However, if the product is introduced after midnight but not sold on the day of introduction, we do not possess data on price, as a result there is no data point recorded on that day, even though there was an offer available over the day. Nevertheless, as the product did not sell over the day, the demand of the product was low, thus the impact should be limited. Furthermore, we note that prices change a lot over a day. A product like a television can go down hundreds of euros over a day between two partners competing for the best offer price and the resulting visibility, as the best offer is updated over the whole day. Similarly, there are flash sale promotions which can be available for only a few hours over the day. Therefore, we opt to use the average selling price of products over a day instead of the best offer price recorded, if the product was sold on a day. This results into the variable *price*.

In this research we consider all *chunk_id*, but we need to modify the *chunk_id* concerning books, as this consists of several million of products, which would exceed the available memory of the used computation processing unit. We split this category in 3 subcategories for Dutch, English and Alternative books. Additionally, we create the variable *stockout*, which signifies the availability of the product during the day. In principle, *stockout* turns true when the product was sold out over the day. However, this boolean is made using logic, this is a result of that stock levels not properly recorded in the database, and especially not for products from partners. This results into that *stockout* becoming true if and only if the following statements are true:

- On the day of selling when there is a potential stockout the product is currently on promotion,
- It is not the final day of the research,
- The day after, the promotion is still going, even if the offer is gone, as promotions are set beforehand even if products run out of stock,

- The day after, the product is not offered anymore at all, or if there is another best offer, it is by a different seller, with a higher price.

In all other cases *stockout* is false. Here the *stockout* boolean can only be true during promotions while a stockout can per definition happen on any day, for the reason we specifically want to elevate the issue of sudden drops of sales during promotions, which could influence the *sales* of a product to drop significantly, although the demand stays the same. Additionally, due to the limitations of the data, using the logic scheme for products which are not specifically in promotion, would lead to a lot of mislabelling. This is due the last constraint which would identify a price increase, as a stockout. For example, since partners compete a lot for price over the day for a lot of products, and bring it back up over the following day. The difference in price increase of the two competitors would be labelled as a stockout.

As we cannot use categorical variables in this research, we transform all data points to either booleans or numerical values. Such that, *day_of_week* is transformed to booleans for every day of the week, therefore, we create the variables: *Monday*, *Tuesday*, *Wednesday*, *Thursday*, *Friday* and *Saturday*. Here we take Sunday as the day for baseline sales. Additionally, we convert the textual data of *promos* into the boolean *price_off*, which is only true for price off promotions. This filters out the display promotions as Bol.com discoved these to not have significant impact on the sales on the platform individually. Furthermore, we consider all the promotions seen during the investigated period, not only the SSP. This decision is driven by that there are a lot of products without any SSP.

Using the aforementioned variables we can create the final dataset. We consider the products that were at least available once during the month of June 2021 and have a minimal of 20 dates where an offer was available over the training period. Here we define the training and validation period as the 1st of September 2020 until 30st of April 2021 and between the 1st of May 2021 until 30th of June 2021, respectively. Wherein, all observations during the given period fall within the training and validation dataset.

For this research we consider data per chunk. At any point there are more than 45 million active products in the Bol.com database, as such we limit to research to products which we had at least 1 active offer during June 2021. Additionally, we only focus on the Netherlands as the prices and promotions of products can differ between the Netherlands and Belgium. Doing the algorithm for both regions is out of scope for this research. This leads to the total dataset

having over 200 billion data point. As this is out of scope due to run-time limitations of this research, we limit ourselves to a data set of 10.000 products. These 10.000 products are selected by using a random subset of chunks which represents the whole database. Significantly reducing the amount of data points to about 40 million.

We note that we have sales data of an online warehouse company, thus there is a certain seasonal pattern to be found in most categories of products, resulting into a sales increase during the holiday period from October to December. But as the period used in this research is short, we do not have enough data points to, for example, take care of these seasonality patterns through extra variables per quarter.

4.2 Dataset 2-step Baseline Model

In this section we provide how the dataset was created used for the live experiment as seen in Section 5.2. For the 2-Step Baseline model we need a dataset which provides data on a weekly basis thus we create a new dataset in a similar fashion as seen to the previous section.

The dataset is mainly based on the same data used in the previous section, as we consider the same period of from the 1st of September 2020 until the 30th of June 2021. Such that, per product we have the identifier *product_id* and *chunk_id*. Furthermore, we collect *week_sales*, which are the number of sales, and the *week_price*, the average selling price, of the product based over the whole week. The week is identified by changing the *date* into the *week* and *year* variables, which are the week number and year identifiers, respectively. Additionally, we use the *promos* data to create the variable *promo_week*. This boolean variable is true when there was a price off promotion during the week and false otherwise. As not all promotions happen during the whole week, this will underestimate the impact of promotions with this method. This is due to the limitations of data in the forecasting model currently in place at Bol.com which is used for this model. The forecast of the forecasting model, B, for predicted demand is stored in *predicted_demand*. We replace the Null values in *predicted_demand* with 0, resulting in *predicted_demand* being nonzero if and only if there has been a sale of the product during the last 56 days. We utilise *week_sales* and *predicted_demand* to create the *uplift* following

$$uplift_{it} = week\_sales_{it} - predicted\_demand_{it}, \tag{1}$$

for product $i = 1, ..., N$, in period $t = 1, ..., T$. Hence, uplift is positive when the sales of the week exceed the predicted demand.

As the algorithm employed has some conditions in which *predicted_demand* is used as the forecast of sales, and we want to test the algorithm specifically for when both steps are performed, we filter out the following data points to reduce the data size:

- If for a product there is no *promo_week* with at least one True value,
- If for a product for all data points *week_sales* equals *predicted_demand*,
- If for a product there are less than 10 data points.

This significantly reduces the amount of data points considered for this research. However, this will impact the results little since only products with little informational value on promotional performance are removed. Here we note that *predicted_demand* is a numerical value with 1 decimal, while *week_sales* is an integer, as a result of the second restriction, only products with zero sales over the whole period are removed from the dataset. Under these restrictions we find a total of 182190 products with close to 7 million data points.

5 Methodology

In this section we provide an overview of all the methods that are used in this research. We describe in what manner these methods are relevant to the research questions. Such that, the Bayesian model is described in Section 5.1. In Section 5.2 we explain the 2-Step Baseline model in detail. Additionally, in Section 5.3 it is explained how the forecasts of the models are applied to create a scoring system as an insight to managers and partners. Then, in Section 5.4 the 2-Step Baseline model is applied in a real life experiment to see the impact of this new provided scoring system. Finally, in Section 5.5 we elaborate on the performance measures used to compare the different models.

5.1 Bayesian Sales Model

The model required for this research needs to be able to predict a product's uplift during promotions. Therefore we require a sales model that appropriately forecasts sales during non-promotion periods and promotion periods. The difference between these forecasts will substitute as the predicted uplift. Additionally, due to the vast catalogue of products and data, we necessitate the model to be scalable and be able to deal with many different data patterns, such as, high selling products versus low selling products and limited data points on promotional performance. Due to individual product level approach and the aforementioned requirements, we propose a Product-Specific Tobit Sales Regression model.

In this section we explain in what way the model is constructed, as well as, by what means the model uses the collected data. Firstly, in Section 5.1.1 we explain how we create a Bayesian Sales model and define estimates for the posterior distribution. Next in Section 5.1.2 it is defined how the acquired estimates are adopted to create forecasts.

5.1.1 Product-Specific Tobit Sales Regression Model

For this model we follow Bayesian assumptions. Bayesian theorem boils down to assume the parameters to be random variables. This is in contrast to the classical Frequentist methods, which assume that parameters have a true value, which is used for the baseline model. Using Bayes theorem, we use our prior believes of the parameters while in conjunction updating these believes by the information of the data to create a posterior distribution of the model. As such, we assume parameters come from a probability distributions. Using these assumptions we derive conditional distributions for each of the parameters in the model, where we define the likelihood function of the data y given the model parameters θ to be $p(y|\theta)$ and prior distribution as $p(\theta)$. Then using Bayes Rule and a kernel function the posterior distribution follows from

$$p(\theta|y) \propto p(\theta)p(y|\theta). \tag{2}$$

First, we consider a Product-Specific Tobit Sales Regression model. Let S_{it} be the observed sales, D_{it} the true demand and O_{it} the stock level of product $i = 1, ..., N$ in period $t = 1, ..., T$. We denote O_{it} is only known when a stockout occurs due to $D_{it} > O_{it}$. Additionally we note that, as we observe sales even when there are more products returned than sold, in other words $D_{it} < 0$, we do not observe negative sales, thus from here it follows that

$$S_{it} = \begin{cases} 0, & \text{if } D_{it} < 0, \\ D_{it} = x'_{it}\beta_i + \varepsilon_{it}, & \text{if } D_{it} > 0 \text{ and } D_{it} < O_{it}, \\ O_{it}, & \text{if } D_{it} > O_{it}, \end{cases} \tag{3}$$

where $\varepsilon_{it} \sim \mathcal{N}(0, \sigma_i^2)$, with $\mathcal{N}$ being a standard normal distribution. Then let x_{it} be a vector of k explanatory variables, where β_i is a k dimensional parameter vector. In the application at hand we denote that x_{it} contains the variables: *price*, *clicks*, *Monday*, *Tuesday*, *Wednesday*, *Thursday*, *Friday*, *Saturday* and *Sunday*, additionally, a constant is added for each product. In the equation we denote that D_{it} contains censored latent data which is not observed. Moreover, there is a lower bound for 0 sales and a varying upper bound depending on the stock level. Let the set of data points for which $S_{it} = 0$ be given by $L = \{(i, t) \ : \ S_{it} = 0\}$, next, let the

data points for which $S_{it} = O_{it}$ be given by $U = \{(i,t) \ : \ S_{it} = O_{it}\}$. Therefore, per definition it follows that $L \cap U = \varnothing$. Given the aforementioned notation, the likelihood function of the model is given by

$$p(S_{it}|\beta_i, \sigma_i^2) = \prod_{(i,t)\in L} \Phi(-x'_{it}\beta_i/\sigma_i) \prod_{(i,t)\notin(L\cup U)} \phi((S_{it} - x'_{it}\beta_i)/\sigma_i) \prod_{(i,t)\in U} \Phi(O_{it} - x'_{it}\beta_i/\sigma_i), \quad (4)$$

with $\phi(\cdot)$ and $\Phi(\cdot)$ being the probability density function (PDF) and cumulative distribution function (CDF) of a standard normal distribution, respectively.

To obtain parameter estimates we follow the proposition of Gelfand & Smith (1990) Markov Chain Monte Carlo (MCMC) sampling through the application of Gibbs sampling. The main advantage of Gibbs sampling is that, although the full joint distribution of the model is unknown, Gibbs sampling retrieves accurate approximates of parameter estimates by iteratively drawing from the conditional distribution for each of the parameters following a Markov chain. Where in each iteration the draws from the previous iteration are used, resulting into eventual convergence of the chain. From then draws come forth from the posterior distribution. Using the aforementioned model, we can derive a conditional distribution of each of the parameters from the given model. We first denote

$$S_{it}^* = \begin{cases} S_{Lit}^*, & \text{if } (i,t) \in L, \\ S_{it}, & \text{if } (i,t) \notin (L \cup U), \\ S_{Uit}^*, & \text{if } (i,t) \in U, \end{cases} \quad (5)$$

and let us define the vectors $S_i^* = (S_{i1}^*, \ldots, S_{iT}^*)$. Then let us assume independent uninformative priors for β and σ due to the many different kind of data in this research. In other words

$$p(\beta_i) \propto 1 \text{ and } p(\sigma_i^2) \propto \sigma_i^{-2}.$$

We follow the derivations from Greenberg (2012) to receive similar full conditional distributions of the parameters. Such that, it follows that for a given product i that

$$\sigma^2|S^*, \beta \sim \mathcal{IG}2((S^* - X\beta)'(S^* - X\beta),\ T), \tag{6}$$

$$\beta|S^*, \sigma^2 \sim \mathcal{N}(\hat{\beta},\ \sigma^2(X'X)^{-1}), \tag{7}$$

$$S_L^*|\beta, \sigma^2 \sim \mathcal{TN}_{(-\infty,0]}(X\beta,\ \sigma^2), \tag{8}$$

$$S_U^*|\beta, \sigma^2 \sim \mathcal{TN}_{[O^*,\infty)}(X\beta,\ \sigma^2), \tag{9}$$

with $\hat{\beta}$ being an initial hyperparameter vector for the mean (in this research the OLS estimate is used), O^* being a vector with known stock level boundary for $i \in U$ and the matrix $X = (x_1, \ldots, x_T))$. Additionally, we denote $\mathcal{IG}2$ as the inverted gamma-2 distribution and $\mathcal{TN}_{(\cdot,\cdot)}$ as the truncated normal distribution with a given interval.

We now perform the Gibbs algorithm to draw from the posterior distribution of the parameters of product i:

B1 Initialise the starting values; set $\hat{\beta}$ and $m = 0$,

B2 Simulate $\sigma^{2(m+1)}$ from Distribution 6,

B3 Simulate $\beta^{(m+1)}$ from Distribution 7,

B4 Simulate $S_t^{*(m+1)}$ for $i \in L$ from Distribution 8,

B5 Simulate $S_t^{*(m+1)}$ for $i \in U$ from Distribution 9,

B6 Set $m = m + 1$,

B7 go to step 2 unless $m > m_{max}$,

where m and m_{max} are defined as the simulation count and maximum number of iterations, respectively. To make sure the algorithm converges there is a set burn-in sample size, m^*, which is the number of iterations before we assume the algorithm to have converged to the posterior distribution. There is not a set way to choose m^*, but after convergence the draws all simulate from stable mean. Although, there will persist a significant autocorrelation between draws from simulation m and $m + 1$, thus, we introduce a thin value n, for which every nth value of m will be recorded. As a result, we can record for every nth the draw from simulation $m > m^*$ the parameter estimates which will be independent draws of the posterior distribution, we denote these draws as $k = 1, \ldots, K$ with $K = \frac{m_{max} - m^*}{n}$.

Given the set of K recorded draws we compute the posterior mean, variance, and the Highest Posterior Density (HPD). Such that, the posterior mean of β_i is given by

$$\frac{1}{K}\sum_{k=1}^{K}\beta_i^{(k)}, \tag{10}$$

similarly, we the posterior variance of β_i is given by

$$\frac{1}{K}\sum_{k=1}^{K}\left(\beta_i^{(k)} - \frac{1}{K}\sum_{k=1}^{K}\beta_i^{(k)}\right)^2. \tag{11}$$

Additionally, we consider the closed 95% HPD, this is the smallest closed interval in which 95% of the K draws of $\beta_i^{(k)}$ are pinpointed. This interval is composed by ordering all draws, subsequently, every 95% closed interval is considered until the smallest is found, which is then denoted as the HPD. Given the HPD of β_i we can deduce if the posterior results have support for the parameters importance if 0 falls outside the interval.

5.1.2 Scenario Forecasting

To compute the relevancy of products for promotion we consider the forecasting possibilities using the posterior distributions of the Sales model. To summarise, for every product we have to compute the expected sales when there is a promotion or not, in other words we want to compute $\mathrm{E}[S_{i,T+1}|x_{i,T+1}, I[promo_{i,T+1}=0]]$ and $\mathrm{E}[S_{i,T+1}|x_{i,T+1}, I[promo_{i,T+1}=1]]$, where $I[promo_{it}=0]$ is an indicator function which is 1 if there is promotional activity for product i at time t and 0 otherwise. Given the model in Equation 3 we simulate one step ahead forecasting of the expected sales for product i through:

C1 Use draws recorded from the Gibbs algorithm Step B2 and Step B3,

C2 Simulate $\varepsilon_{T+1}^{(k)}$ from $\mathcal{N}(0, \sigma^{2(k)})$ for $k = 1, \ldots, K$,

C3 Compute $S_{T+1}^{(k)} = x'_{T+1}\beta_{T-1}^{(k)} + \varepsilon_{T+1}^{(k)}$ given $I[promo_{T+1}=0]$ for $k = 1, \ldots, \frac{K}{2}$,

C4 Compute $S_{T+1}^{(k)} = x'_{T+1}\beta_{T-1}^{(k)} + \varepsilon_{T+1}^{(k)}$, given $I[promo_{T+1}=1]$ for $k = (\frac{K}{2}+1), \ldots, K$,

C5 $\mathrm{E}[S_{T+1}|x_{T+1}, I[promo_{T+1}=0]] \approx \frac{2}{K}\sum_{k=1}^{\frac{K}{2}}\min(S_{T+1}^{(k)}, 0)$

C6 $\mathrm{E}[S_{T+1}|x_{T+1}, I[promo_{T+1}=1]] \approx \frac{2}{K}\sum_{k=\frac{K}{2}+1}^{K}\min(S_{T+1}^{(k)}, 0)$,

where the values for x_{T+1} are based on the different scenarios of whether product i is in promotion or not. Since sales cannot be lower than 0 we use the minimum of computed sales and 0. As a result, the expected value might be slightly biased upwards. Additionally we note that

the usage of different draws for Step C3 and Step C4. This is to make sure the results are individually independent of each other.

The database employed holds only either the promotional price of a product or normal price of the given day, thus to compute uplift we construct one of either scenario, we modify the data as follows depending on the scenario: First, we consider $\mathrm{E}[S_{i,T+1}|x_{i,T+1}, I[promo_{i,T+1} = 0]]$ when $price\_off_{i,T+1} = 1$ in the database. In this case we modify the $clicks_{i,T+1}$ based on the average clicks of the week before the promotional activities started, assuming there is such a period, otherwise $clicks_{i,T+1}$ remains unchanged. Similarly, $offer_{i,T+1}$ is changed based on the average offer price before the promotional activities started or unchanged if the current price is higher. The other data variables do not need to change due to a change in promotional activities. Secondly, we cover $\mathrm{E}[S_{i,T+1}|x_{i,T+1}, I[promo_{i,T+1} = 1]]$ when $price\_off_{i,T+1} = 0$ in the database. $clicks_{i,T+1}$ is based on the max value of either the average clicks during the last 7 days of the previous promotional period or the average clicks of the product during the last week without promotions. The max value of either is taken to take care of an increase in popularity due to promotion or a sudden increase in popularity or due to seasonality effect, respectively. Then $offer_{i,T+1}$ is derived by taking the minimum of either the average price of the product the last promotional period, of which again up to a week is considered or 90% of the current price. In this case we consider that the regular price of a product can have dropped significantly since the last promotional period and 90% of the current price is chosen, as 10% discount of the original price is the minimum promotion price on the platform. In the other cases the data can be directly used for the forecasts.

Using $\mathrm{E}[S_{i,T+1}|x_{i,T+1}, I[promo_{i,T+1} = 0]]$ and $\mathrm{E}[S_{i,T+1}|x_{i,T+1}, I[promo_{i,T+1} = 1]]$, we can now define the expected uplift, $\mathrm{E}[\mathcal{U}]$, which follows from

$$\mathrm{E}[\mathcal{U}_{i,T+1}] = \mathrm{E}[S_{i,T+1}|x_{i,T+1}, I[promo_{i,T+1} = 1]] - \mathrm{E}[S_{i,T+1}|x_{i,T+1}, I[promo_{i,T+1} = 0]]. \quad (12)$$

The $\mathrm{E}[\mathcal{U}]$ is the expected amount a product will sell more if the product is put on promotion. This is of high interest for the relevancy of a product for promotion. As a product that does not sell more due to promotion can be considered irrelevant.

A big advantage of using the Gibbs algorithm is that we can update the posterior distribution as new information comes available. Meaning the algorithm does not need to be rerun from scratch, but rather we add the new data points to the previously converged algorithm and

let it run a new simulation count to converge to the new posterior distribution. As such, the algorithm will converge faster than running from the start, saving time and resources. This is in our interest as in this research we only perform one step ahead forecasts, which need to be tested for accuracy. As a result, we can create a moving window of our training and evaluation dataset, in which the model is initially trained with the training data and gets readjusted using the new data input of the validation set after a forecast of the given date has been created.

5.2 2-Step Baseline model

As a benchmark model we follow the framework of Abolghasemi et al. (2020) to create a 2-Step Sales Regression model. The 2-step approach is as follows: First we compute a baseline sales forecast without promotional data. Secondly, we use these forecasts from the first step in conjunction with promotional data in a regression form to get a 2-step Sales Forecasting model. As baseline sales forecasts, B_{it}, we use the forecast from the team of Bol.com as discussed in Section 3.2. This model cannot use the signals it gets from promotions properly, but it creates baselines sales forecasts, B_{it}, thus it assumes no promotions will happen. Then let the sales model be

$$\mathcal{S}_{it} = B_{it} + I[promo\_week_{it} = 1](\alpha_i + P'_{it}\gamma_i + \epsilon_{it}), \tag{13}$$

with P_{it} and $\mathcal{S}_{it}$ the price and sales, respectively. Let $I[promo\_week_{it} = 1]$ be an indicator function if there was a price off promotion for product $i = 1, ..., N$ at time $t = 1, ..., T$, respectively, here we note that t is in weeks instead of days unlike in Section 5.1, α_i as a constant, γ_i as a parameter, and $\epsilon_{it} \sim \mathcal{N}(0, \sigma_i^2)$. In short, the uplift is regressed when a promotion occurs against the price.

Using the above mentioned model we easily compute uplift forecasts as well. Here we note that we can use $B_{i,(T+1)}$ as baseline sales forecast without promotion. For the forecast of sales during promotions, $\mathrm{E}[\mathcal{S}_{i,T+1}|I[promo\_week_{i,T+1} = 1]]$, can be easily calculated by using the parameter estimates computed for Equation 13. We use the data point of $P_{i,T+1}$ in a similar way as to how the $offer_{i,T+1}$ data was changed in Section 5.1. In other words, we change the data point $P_{i,T+1}$, which is the *week_price*, if $promo\_week_{i,T+1} = 0$. In this case the *week_price* is derived by using the minimum of 90% of the current price or of the average price of the product during the last promotional period, using the same reasoning as in Section 5.1. Additionally, if $\mathrm{E}[\mathcal{S}_{i,T+1}|I[promo\_week_{i,T+1} = 1]$ is lower than $\mathrm{E}[\mathcal{S}_{i,T+1}|I[promo\_week_{i,T+1} = 0]$, which would imply the product sells less when it is on promotion, the former value is replaced by the latter. This results into that for each product the forecasts of the baseline sales and promotional

are computed. Following the other model we create a moving window for the forecasts of the expected sells, such that, the algorithm has to be rerun as new data becomes available. Resulting that we can compare the Bayesian model to the 2-Step Baseline Model as close as possible. By using the moving window the model will quickly use any promotional data as it comes available.

5.3 Score System

To create a score which can be shared with managers, we consider the forecast for the different models $\mathrm{E}[S_{i,T+1}|I[promo_{i,T+1}=1]]$ and $\mathrm{E}[\mathcal{S}_{i,T+1}|I[promo\_week_{i,T+1}=1]]$. For the main part of this research we consider the following scoring system: Let $q=1,\ldots,\mathcal{Q}$ be the list of all *chunk_id*, then we consider per q all products for which $chunk\_id_i=q$, for which this new list has all products with the same *chunk_id*. As a result, we order all order these products based on $\mathrm{E}[S_{i,T+1}|I[promo_{i,T+1}=1]]$ and $\mathrm{E}[\mathcal{S}_{i,T+1}|I[promo\_week_{i,T+1}=1]]$. Then for either algorithm we create a linear scoring system in which the highest predicted selling product, j, for a given q, is scored 100 and the rest of products are scored following:

$$Y = 100\frac{E[S_{i,T+1}|I[promo_{i,T+1}=1]]}{E[S_{j,T+1}|I[promo_{j,T+1}=1]],} \text{ for } chunk\_id_i=q, \tag{14}$$

and similarly,

$$\mathcal{Y} = 100\frac{E[\mathcal{S}_{i,T+1}|I[promo_{i,T+1}=1]]}{E[\mathcal{S}_{j,T+1}|I[promo_{j,T+1}=1]],} \text{ for } chunk\_id_i=q, \tag{15}$$

with Y and $\mathcal{Y}$ as the assigned new Relevance Score for the Bayesian and baseline model, respectively. This is done for all $q=1,\ldots\mathcal{Q}$, until all products are rated at $T+1$. With T being the end of the training dataset. Then we repeat the process for the following T. Here we note that the expected sales per period are created using a moving window, the T will increase by 1 for each period. Meaning that we will still employ the one step ahead forecasts. This way we make score predictions over the whole validation dataset with a variation of the scores over the time.

5.4 Experiment

To determine the impact of a new score on the behaviour of plaza partners we perform an A/B testing experiment in a real-world setting. An A/B test is a test where we have two equal sized groups where each individual is assigned random to either group. Subsequently, each group is assigned one variant, in this case the groups are the partners of Bol.com and the variant is showing a score. We note that currently the Relevance Score is shown to partners of Bol.com for their own assortment as insight as to what products are popular on the platform. One group

get the control variant which is the Relevance Score from Section 3.1, and the other group, the treatment group, gets shown a new variant, the score produced by the 2-Step Baseline model as explained in Section 5.2. The 2-Step Baseline model is used for this experiment as it is operational for the whole database within the time constraints to perform it on a daily basis. After running the test for a period in a real-world environment the test finishes, after which the data collected can be used to see if there is a significant difference between the performance of the two groups. To further elaborate, the goal of the experiment is to test the performance of a new score compared to the old Relevance Score and analyse if partners react to a different score by creating more SSP with relevant products.

The plaza partners are not enforced to use the score to create SSP and only a small percentage of partners look at the score at all, resulting in an extension of the period the experiment has to run to get any significant results. As such, the experiment ran for two weeks from 16th to 29th of June 2021. Additionally, the data is available on what partners see the Relevance Score, but we do not know if the partners have used the score as a decision factor for creating a promotion. Furthermore, the partners do not know the test is happening therefore the score created for the treatment group is normalised to fit over the interval 5 to 87 which partners expect from the Relevance Score.

5.5 Performance Measures

To provide meaningful performance measures for the models at hand, we review a multitude of aspects. First, we introduce the MSE formally in Section 5.5.1. An additional performance measure used for the models is the estimation time as discussed in Section 5.5.2. Lastly, in Section 5.5.3 the measures for the results of the real-world experiment are given.

5.5.1 Mean Squared Error

We follow the standard set in Bol.com of using the MSE to score the performance of the models as a forecasting algorithm. The main difference between MSE and other prominent error scoring measures, such as Mean Absolute Error (MAE), is that with MSE outliers are penalised harder. The decision to use MSE is twofold, first is that outliers in sales will happen mostly during promotional periods, thus penalising big errors in these periods falls in line with the goals of this research, and secondly having big errors in sales forecasts can lead to major supply shortages if demand is much higher than expected. From here it follows that MSE for the Bayesian and

2-Step Baseline model is given by:

$$\text{MSE}_B = \frac{1}{TN}\sum_{i=1}^{N}\sum_{t=1}^{T}|S_{i,t} - \text{E}[S_{i,t}]|^2\,, \tag{16}$$

and

$$\text{MSE}_2 = \frac{1}{TN}\sum_{i=1}^{N}\sum_{t=1}^{T}|S_{i,t} - \text{E}[\mathcal{S}_{i,t}]|^2\,, \tag{17}$$

respectively, where $S_{i,t}$ is the real observed sales, products $i = 1, \ldots, N$ and $t = 1, \ldots, T$ where $t = 1$ is set at the evaluation dataset and T is the final day of the dataset used in this research. $\text{E}[S_{i,t}]$ and $\text{E}[\mathcal{S}_{i,t}]$ are expected sales of the posterior mean of Bayesian model and expected sales of the 2-Step Baseline model, respectively. We note that the expected sales are based on whether there is promotional activity going on or not. The application of MSE will be done in two ways. First, we compare the MSE over the whole assortment between MSE_B and MSE_2. Secondly, the MSE is used in a different setting. To put more emphasis on the forecasting performance of promotions, we specifically look at the MSE during promotions. Such that, we denote that the MSE during promotional activities is specified as:

$$\text{MSE}_B = \frac{1}{TN}\sum_{i=1}^{N}\sum_{t=1}^{T}|S_{i,t}I[promo_{i,t} = 1] - \text{E}[S_{i,t}|I[promo_{i,t} = 1]]I[promo_{i,t} = 1]|^2\,, \tag{18}$$

and

$$\text{MSE}_2 = \frac{1}{TN}\sum_{i=1}^{N}\sum_{t=1}^{T}|S_{i,t}I[promo_{i,t} = 1] - \text{E}[\mathcal{S}_{i,t}|I[promo_{i,t} = 1]]I[promo_{i,t} = 1]|^2\,, \tag{19}$$

for the Bayesian and 2-Step Baseline model, respectively. Where $I[promo_{i,t} = 1]$ is the indicator function that is 0 if there is no promotional activity and 1 otherwise. Such that, for this performance measure only products are considered when promotions activity is happening. Similarly, we create MSE when there is no promotion going on,$I[promo_{i,t} = 0]$, for the Bayesian and 2-Step Baseline model.

5.5.2 Estimation time

Next, we note that the usage of computing power is considerable for a large database of products, it is of high importance to compare the estimation time between models. Cause by a significant increase of predictive power of a model can be outweighed by the costs of running it. This is especially the case within the context of Bol.com, where all forecasting needs to happen within a few hours overnight for the whole database because there are multiple other models dependent

on the results of the forecasting algorithm. As a result, in this research the estimation time of the models is a significant factor in the creation of a new Relevance Score for promotion.

5.5.3 Experiment Performance

To measure the performance of the real-world experiment as described in Section 5.4 we perform a permutation test on the average number of products sold per day (Cobb, 1998). A permutation test boils down to that the test computes whether there is a significant difference between two samples. In other words we specify the following null hypothesis: *"The average number of products sold per day during SSP of the control group is greater than or equal to the treatment group."*, and the alternative of the hypothesis is: *"The average number of products sold per day during SSP of the control group is less than the treatment group."*. In other words it follows that the null hypothesis is: $\mu_{control} \geq \mu_{treatment}$, and the alternative: $\mu_{control} < \mu_{treatment}$, where μ is defined as the average number of products sold per day during SSP, for the control group and treatment group, respectively.

The permutation test is a non-parametric test, as such, the algorithm is as follows:

D1 Compute $TS_e = \mu_{treatment} - \mu_{control}$ and set $m_p = 1$,

D2 Create a permutation, by reassigning all partners to either the control or treatment group,

D3 Compute $TS_p = \mu_{treatment} - \mu_{control}$,

D4 Go to step 2 unless $m_p > m_{p\_max}$

with TS_e and TS_p as the test statistic of the experiment itself and the permutations, respectively. m_p is the iteration count of the permutations and m_{p_max} is the total amount of permutations done. As a result, we use the distribution of TS_p to compute if we can reject the null hypothesis on a significance level of 5%. For this the TS_e has to be in the right tail of the distribution in which less than 5% of the weight of distribution is concentrated.

6 Results

In this section, we discuss how the methods are integrated in combination with the usage of cloud computing. Furthermore the produced results are presented and compared. To start, the implementation and hyperparameter settings are discussed in Section 6.1. Next, in Section 6.2 we present an analysis on the performance and comparison of the methods at hand. Lastly, in Section 6.3 the results of the real-world experiment are analysed.

6.1 Implementation details

To implement the models we make use of the data that is stored on Google BigQuery, this is an online service from Google on which the databases of Bol.com are stored. The data is pre-processed on the BigQuery using SQL as mentioned in Section 4. The framework of the models themselves is implemented using Python (Van Rossum & Drake, 2009), we opt to implement as much ourselves as possible to not use big packages. A limitation of this approach is that our implementation cannot handle the huge database of products due to run time limitations as explained in Section 6.2.4. The framework connects to Google AI Platform, this is another service by Google which allows the computations to be done through to the cloud. This allows us to select a Graphics Processing Unit (GPU) to perform the models on. Specifically, we make use of *NVIDIA V100* GPU in combination with a virtual Central Processing Unit and up to 78 GB of Random Access Memory (RAM). If the computations were done locally it would have to be done on a CPU, because of the limited GPU capabilities in Bol.com hardware. Additionally, the machine would slow down computations considerably due to a limited amount of RAM available. In general, the main advantage of executing models on GPU, is that GPUs are specifically optimised to perform computations through matrix multiplication, which are heavily used during our computations. This significantly speeds up the process of the algorithms. Although the implementation of the algorithms in this research does not use much of the GPU capabilities, the huge step up in RAM removes the bottleneck in memory usage.

After the python framework is connected to AI Platform it retrieves a batch of chunks with data from BigQuery using SQL. Not all chunks can be stored in memory at once due to memory capabilities. Then as mentioned in Section 4.1 the framework splits the data into the training and validation dataset. The training dataset is used to train the models, which iteratively includes data points from the validation dataset after the forecast of the specified point, as described in Section 5.1.2. In the implementation of the Bayesian model we make use of the result that σ^2 is distributed from $\mathcal{IG}2\left((S^* - X\beta)'(S^* - X\beta),\ T\right)$. This leads to simulating a value for σ^2 by incorporating $\frac{(S^*-X\beta)'(S^*-X\beta)}{\sigma^2} \sim \chi^2(T)$, with $\chi^2(T)$ being the Chi-squared distribution with T degrees of freedom. As a result, we sample for σ^2 by simulating a value from $\chi^2(T)$ and dividing that by $(S^* - X\beta)'(S^* - X\beta)$. Moreover, using the inverse CDF technique we can sample from the truncated normal distribution, as seen in Distribution 8 and Distribution 9, using the standard normal distribution and uniform distribution. As such, we

sample $\mathcal{TN}_{[lb,ub]}(\mu, \sigma)$, with lb and ub as the lower and upper bound, respectively, from

$$\mu + \sigma\Phi^{-1}(\Phi((lb-\mu)/\sigma) + u(\Phi((ub-\mu)/\sigma) - \Phi((lb-\mu)/\sigma))),$$

with u as a draw from the uniform distribution, U(0,1), and $\Phi(\cdot)$ the CDF from the standard normal distribution.

Additionally, we note that data of a specific products sometimes do not have full rank when there are variables that never change over the whole set. For example, price or promotional activity can be 0 over the whole dataset of the product, due to a product never going on promotion and the standard price never changing. In this case we remove the variable that does not change over the data from the specific product, so that we can still create predictions and point estimates. The rest of the implementation follows orderly the methodology. The results of the draws over the moving window are stored with the posterior results back into the BigQuery database, from which the forecasting happens through SQL. The framework is created in a similar fashion for the 2-Step Baseline model, but due to the limited number of products it can compute all weights in one batch of data.

For the Bayesian model the hyperparameters, m^*, m_{max} and n have to be set in such a way that the Gibbs sampler converges to the posterior distribution and has sufficient draws recorded to create the posterior results but setting m_{max} too high would make computation time for the algorithm too high. Here we note that since the whole algorithm is based on creating one step ahead forecasts, for every step we have to create new posterior results. Again, here we can use the draw of $\beta_i^{(m)}$ and $\sigma_i^{(m)}$ of the previous step to converge faster as explained in Section 5.1.2. Resulting in maximum amount of draws $m_{max} = 10000$ with burn-in sample $m^* = 5000$ for the first forecast period of product $i = 1, \ldots, N$, after which for all the forecast periods we set $m_{max} = 5000$, $m^* = 2500$. For all periods we set the thin value $n = 10$.

6.2 Performance Results

In this section we present the validity of the models at hand. First, we analyse the performance of the Bayesian and 2-Step Baseline model in Section 6.2.1 and Section 6.2.2, respectively. Subsequently, the models are compared and further analysed through the MSE in Section 6.2.3. Lastly, in Section 6.2.4 we consider the computation time of the models.

6.2.1 Bayesian model

We note that presenting point results of the posterior distribution for all products is not feasible and combining it within one table would not represent the sheer difference between products and results well. As a result, we consider a randomly picked product first over different moments in the validation period. Subsequently, we consider a different kind of product and perform a comparison. Thirdly, performance statistics over the whole data sample are given and analysed.

We present point estimates in Table 1 of a typical well selling seasonal product from the home ventilation product group which sold about 1000 times and had over 50000 clicks during 2020, with an average price of 25 euros. Due to the confidential nature we cannot share specific data of the product. The point estimates of the posterior results are based on the data used for the one-step ahead forecasts for the first of May 2021. To start off, the posterior mean stays consistent over the different variables. Meaning there is an indication that the Gibbs sampler had converged after the burn-in period.

Table 1: Point estimates of the Posterior Distribution for the first 10%, 50% of draws and mean overall, Standard Deviation (SD) and the 95% Highest Posterior Density (HPD) for the First of May 2021 for a seasonal summer product.

β	Mean 10%	Mean 50%	Mean	SD	95% HPD interval	
Constant	-1.26	-1.32	-1.29	1.89	-3.87	1.20
Clicks	$3.28 \cdot 10^{-2}$	$3.28 \cdot 10^{-2}$	$3.27 \cdot 10^{-2}$	$3.29 \cdot 10^{-2}$	$2.87 \cdot 10^{-2}$	$3.59 \cdot 10^{-2}$
Promo	5.02	5.01	5.01	0.49	4.29	7.10
Price	$-4.03 \cdot 10^{-2}$	$-3.85 \cdot 10^{-2}$	$-3.92 \cdot 10^{-2}$	$2.38 \cdot 10^{-3}$	-0.14	$-2.40 \cdot 10^{-2}$
Monday	$-4.22 \cdot 10^{-2}$	$-3.28 \cdot 10^{-2}$	$-2.80 \cdot 10^{-2}$	0.22	-0.87	0.94
Tuesday	0.52	0.50	0.52	0.21	-0.33	1.48
Wednesday	0.59	0.62	0.60	0.21	-0.44	1.45
Thursday	0.33	0.25	0.25	0.23	-0.80	1.11
Friday	0.12	0.13	0.13	0.20	-0.81	0.93
Saturday	$-3.67 \cdot 10^{-2}$	$-3.82 \cdot 10^{-2}$	$-3.63 \cdot 10^{-2}$	0.21	-1.02	0.80

When considering the 95% HPD interval we denote that the variables *clicks*, *promo* and *price* have posterior support a non-zero effect, as these variables have 95% HPD intervals which exclude 0. In other words there is no posterior support for the restriction of these variable to be set to 0. Contrarily, for the other variables, *constant*, *Monday*, *Tuesday*, *Wednesday*, *Friday*, *Thursday*, *Thursday* and *Saturday* do include 0 within the 95% HPD, thus, we find that there is posterior support for these variables to be absent. As for the variables *clicks*, *promo* and *price*, which have posterior results that support a non-zero effect, the positive and negative sign we see such as, the positive impact of clicks and promotional activity and negative impact of price falls within expectation. Even though the impact of price is low, promotional activities increase the sales substantially. This implies that the price elasticity of the product is low in

general but customers are susceptible to promotions. The higher Standard Deviation (SD) for *promo* can be explained by the usage of a boolean variable for promotional activity thus every promotion in considered the same concerning the data, although in reality there is a difference in the kinds of promotions. Therefore, the accuracy of promotional results is lower.

So far, we have considered point estimates for one product for the first day of the validation period. Next, we consider Table 1 in comparison to the Table 2 and Table 3, which consider the same product but the data up to the 1st of June and 30th of June 2021, respectively. In the tables we again notice the point estimates of the mean to stay consistent over the 10%, 50% and whole interval. Moreover, the variables *clicks*, *promo* and *price* have posterior results that support non-zero effects over the whole validation period. Furthermore, these variables have the point estimates of the posterior distribution stay relatively the same over the whole validation period. The exception here is for the point estimates of price which become more negative over June. This likely has to do with a price reduction (not as promotion) of the product in June and high demand for ventilation after the weather warmed up after a relatively cool month of May.

Table 2: Point estimates of the Posterior Distribution for the first 10%, 50% of draws and mean overall, Standard Deviation (SD) and the 95% Highest Posterior Density (HPD) for the 1st of June 2021 for a seasonal summer product.

β	Mean 10%	Mean 50%	Mean	SD	95% HPD interval	
Constant	-2.41	-2.51	-2.43	1.64	-4.58	0.29
Clicks	$3.96 \cdot 10^{-2}$	$3.93 \cdot 10^{-2}$	$3.96 \cdot 10^{-2}$	$1.32 \cdot 10^{-6}$	$3.72 \cdot 10^{-2}$	$4.17 \cdot 10^{-2}$
Promo	5.03	5.03	5.04	0.46	4.34	6.99
Price	$-4.43 \cdot 10^{-2}$	$-3.78 \cdot 10^{-2}$	$-4.17 \cdot 10^{-2}$	$2.11 \cdot 10^{-3}$	-0.12	$-3.93 \cdot 10^{-2}$
Monday	-0.42	-0.38	-0.34	0.22	-1.38	0.43
Tuesday	0.31	0.26	0.24	0.23	-0.64	1.13
Wednesday	0.41	0.43	0.45	0.20	-0.30	1.48
Thursday	0.36	0.17	0.19	0.22	-0.73	1.17
Friday	0.12	0.19	0.17	0.23	-0.75	1.10
Saturday	-0.45	-0.49	-0.47	0.26	-1.35	0.52

Table 3: Point estimates of the Posterior Distribution for the first 10%, 50% of draws and mean overall, Standard Deviation (SD) and the 95% Highest Posterior Density (HPD) for the 30th of June 2021 for a seasonal summer product.

β	Mean 10%	Mean 50%	Mean	SD	95% HPD interval	
Constant	-0.11	-0.59	-0.24	2.22	-3.87	1.63
Clicks	$5.03 \cdot 10^{-2}$	$5.03 \cdot 10^{-2}$	$5.03 \cdot 10^{-2}$	$7.24 \cdot 10^{-7}$	$2.87 \cdot 10^{-2}$	$5.19 \cdot 10^{-2}$
Promo	5.03	5.04	5.04	0.37	4.58	6.98
Price	-0.20	-0.18	-0.19	$9.18 \cdot 10^{-3}$	-0.13	$-7.68 \cdot 10^{-3}$
Monday	1.08	0.80	0.76	0.16	-0.86	1.30
Tuesday	0.69	0.83	0.81	0.14	-0.33	1.25
Wednesday	0.21	0.25	0.14	0.35	-0.44	1.28
Thursday	0.71	0.35	0.37	0.20	-0.80	1.21
Friday	1.07	1.14	1.13	0.25	-0.80	1.66
Saturday	-0.79	-0.75	-0.80	0.12	-1.02	0.72

Since the boolean variables for dates of week have posterior support to be absent over the whole validation period for the above-mentioned product. We also consider the Bayesian Model without these dates of week as variables. The results for the last day of the validation period, 30th of June 2021, are given in Table 4. What is interesting about these results is that the results for the *clicks*, *promo* and *price* stay practically the same for this reduced model compared to those in Table 3. For the *constant* there again is posterior support for it to be absent in this model. Moreover, we note that the SD of the *constant* increases, this is explained by the weekly patterns of products now captured within the *constant* as the other variables remained unchanged. We have not run this model for all of the products due to it being out of scope of this research. This will be further elaborated upon in Section 6.2.4

Table 4: Point estimates of the Posterior Distribution for the first 10%, 50% of draws and mean overall, Standard Deviation (SD) and the 95% Highest Posterior Density (HPD) for the 30th of June 2021 for a seasonal summer product.

β	Mean 10%	Mean 50%	Mean	SD	95% HPD interval	
Constant	$5.48 \cdot 10^{-2}$	0.28	0.35	7.23	-5.30	5.37
Clicks	$5.03 \cdot 10^{-2}$	$5.03 \cdot 10^{-2}$	$5.04 \cdot 10^{-2}$	$8.07 \cdot 10^{-7}$	$4.8 \cdot 10^{-2}$	$5.21 \cdot 10^{-2}$
Promo	5.10	5.11	5.09	0.43	4.47	7.10
Price	-0.19	-0.19	-0.19	$1.02 \cdot 10^{-2}$	-0.39	$-3.14 \cdot 10^{-3}$

The results so far have considered one product to get a feel for the results, for that reason we compare the results to a different product. In Table 5 we consider a product from the dishwasher product group, more specifically it is a non-electrical magnet against limescale, which during 2020 had about 100 sales, 3500 clicks and an average price of about 40 euros. The point estimates of the posterior results are again for the one step ahead forecasts for 30th of June 2021 to make full use of the dataset. We observe that the posterior results are less consistent for this product over the draws. As illustrated, the mean on the first 10% of the draws, first 50% of the draws and over the whole sample change more than the product seen in Table 3. This can be explained by the higher SD of the variables.

Table 5: Point estimates of the Posterior Distribution for the first 10%, 50% of draws and mean overall, Standard Deviation (SD) and the 95% Highest Posterior Density (HPD) for the 30th of June 2021 for a kitchen aid product.

β	Mean 10%	Mean 50%	Mean	SD	95% HPD interval	
Constant	-0.54	-0.40	-0.31	8.75	-7.04	4.94
Clicks	$9.02 \cdot 10^{-2}$	$8.97 \cdot 10^{-2}$	$9.02 \cdot 10^{-2}$	$5.75 \cdot 10^{-7}$	$8.86 \cdot 10^{-2}$	$9.16 \cdot 10^{-2}$
Promo	2.22	2.20	2.17	0.69	0.78	3.16
Price	-0.16	-0.16	-0.16	$1.17 \cdot 10^{-2}$	-0.45	$-2.30 \cdot 10^{-2}$
Monday	1.18	0.93	1.02	0.75	-0.22	3.51
Tuesday	1.28	1.07	1.08	0.55	0.12	2.27
Wednesday	0.59	0.39	0.41	1.27	-1.71	2.85
Thursday	0.63	0.40	0.50	1.23	-1.59	2.32
Friday	1.44	1.31	1.34	1.03	-0.41	2.89
Saturday	-0.21	-0.43	-0.38	1.27	-2.74	1.69

We denote that *clicks*, *promo*, *price* are *Tuesday* have posterior support to have a non-zero effect. Therefore, the results of *clicks*, *promo*, *price* fall in line with the aforementioned product, the latter variable of *Tuesday* having posterior support for a non-zero effect might be caused by dishwashers being ordered from the weekend being delivered on Monday and Tuesday, resulting in that people buy a magnet against the newfound limescale on their dishes more consistently on this day.

We have seen 2 example products of posterior results, next we analyse how often a variable has posterior support for a non-zero effect for the 10000 products considered in this research. For the last day of the validation dataset, the results can be seen in Table 6. As mentioned, the products which did not have full rank in their data had the problematic variables removed, the number of products the variable are considered by can be seen in the second column. Here we notice the *constant*, *clicks*, *promo* and *price* have most often posterior support for a non-zero effect. As mentioned before, 50% of products do not sell during promotions and there are a lot of products that only have a few clicks and sales per year, it is within expectation that this is reflected in how often the posterior results show support for non-zero effect. Note that for the variables of the days of the week, *Friday* and *Saturday* have the least often posterior support for a non-zero effect. This can be rationalised by the 'weekend' effect, as we took Sunday as the baseline day of the week, the number of products sold on Friday and Saturday are the least different compared to the other days of the week.

Table 6: Number of times variable has posterior support a non-zero effect compared with how often the variable has full rank thus has its posterior results computed on the 30th of June 2021 for the sample of 10000 products.

β	Posterior Support for Non-Zero Effect	Data Variable Has Full Rank
Constant	3447	10000
Clicks	7854	10000
Promo	2682	4252
Price	5951	9657
Monday	788	10000
Tuesday	712	10000
Wednesday	745	10000
Thursday	840	10000
Friday	587	10000
Saturday	431	10000

Finally, we consider the newly created score for the Bayesian model or Bayesian Score for short as created in Section 5.3. The distribution of the scores can be seen in Figure 4 over the validation period, this excludes the score 0 which includes over 70% of the observations. We observe that the score realises a tweedie distribution with the point mass on 0, after which an exponential distribution is realised. This is in line with the findings of Bol.com's own prediction model as described in Section 3.2.

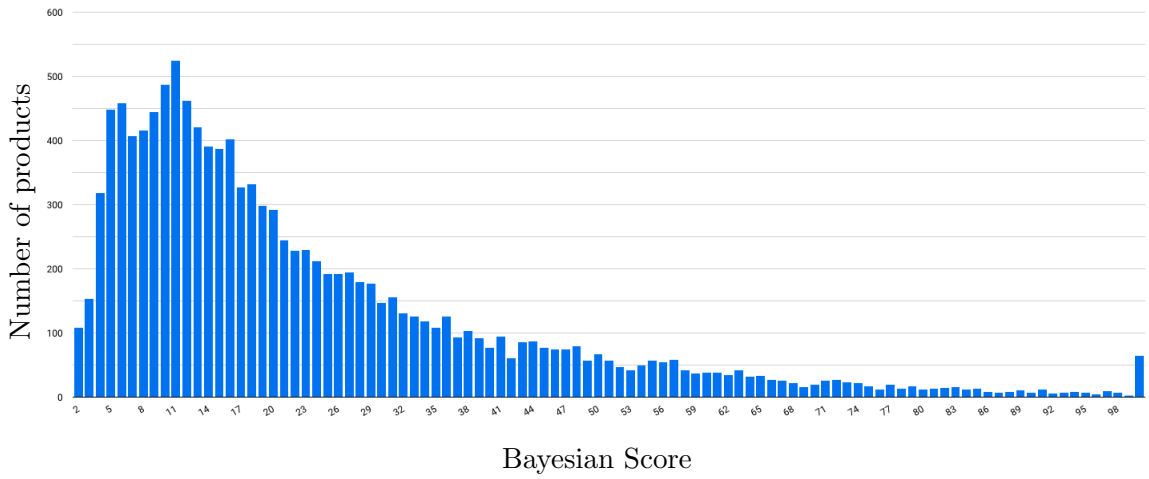

Figure 4: Bayesian Score against the number of products in each score group excluding 0 during the validation period.

Additionally, We have seen in Figure 3 how there is a low correlation between the Relevance Score and the amount of sales during promotion. In the Bayesian Score is compared to the daily sales during promotion from SSP. The correlation found between these is 0.54, which is much higher than the 0.28 seen for the Relevance Score. This indicates a stronger forecasting power than the Relevance Score

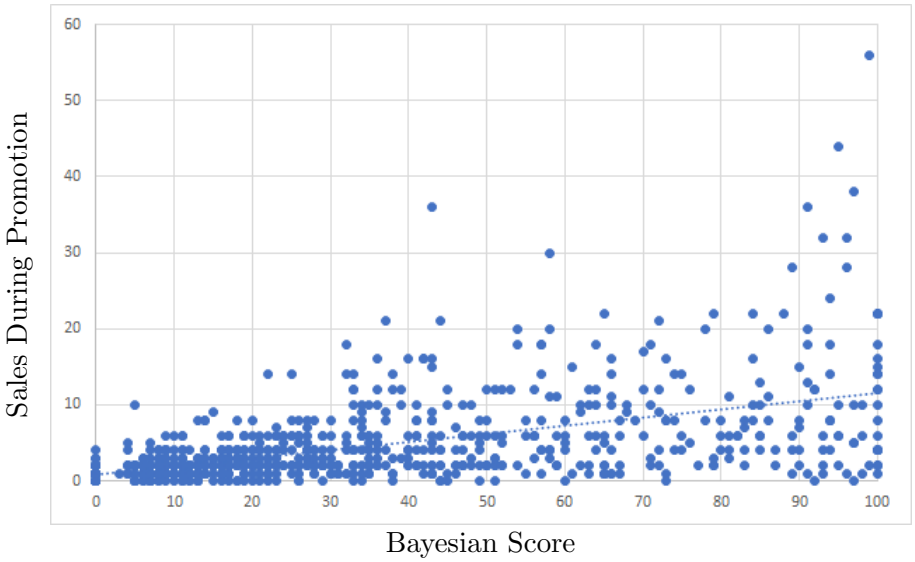

Figure 5: Bayesian Score against the average daily sales of SSP during the validation period.

6.2.2 2-Step Baseline model

Next, we analyse the 2-Step Baseline model. As for this model the database of products is significantly larger, we first consider the results from a subset of products from different *chunk_id*, as seen in Table 7. Here we see the results of the second part of the 2-step regression for a random sample of 10 products in which the parameter estimates for the constant and price are shown. Here we notice that through a t-test that for different products any combination of the variables can be significant for predicting promotional uplift. Therefore we consider Table 8 in which for the whole sample of the 2-Step Baseline model the significant variables and in what combination are given. We immediately note that for 64% of the sample neither variable is significant, which suggests that the model is not a good fit for the data. We argue this might be caused by the nature of the model, as it uses the Bol.com forecast as baseline sales. The actual sales can be lower than the predicted baselines sales resulting in a negative uplift the model tries to predict. Additionally, there is the huge issue of sparsity of promotional data, as all promotions are concatenated to one data point, such that a lot of products only have a few data points to create estimates on. Therefore, there will be underestimation of the uncertainty within the second step with the limited sample size. Thus, it is of no surprise that the model could not get a good fit for a lot of products. Moreover, the high majority of products where the model could not find any significance fall within the 50% of products do not sell anything during promotion.

Table 7: Weights of coefficients for the second step of the 2-step regression on the uplift during promotions between a constant and price for a small sample of products for the last week of the June 2021.

Product	$\beta_{Constant}$	β_{Price}
1	4.35^{*} (0.92)	$-2.31 \cdot 10^{-3}$ $(7.20 \cdot 10^{-3})$
2	0.27 (1.39)	$-7.72 \cdot 10^{-2*}$ $(2.05 \cdot 10^{-2})$
3	0.26 (0.40)	$-6.56 \cdot 10^{-3}$ $(5.30 \cdot 10^{-3})$
4	1.29 (0.66)	$-2.47 \cdot 10^{-2*}$ $(9.28 \cdot 10^{-3})$
5	0.81^{*} (0.20)	$-5.87 \cdot 10^{-3*}$ $(2.75 \cdot 10^{-4})$
6	0.18 (0.13)	$-8.83 \cdot 10^{-4}$ $(1.27 \cdot 10^{-3})$
7	1.26^{*} (0.59)	$-2.12 \cdot 10^{-2*}$ $(8.12 \cdot 10^{-3})$
8	2.76^{*} (1.21)	$-6.42 \cdot 10^{-3}$ $(2.76 \cdot 10^{-2})$
9	1.88^{*} (0.67)	$-2.18 \cdot 10^{-2*}$ $(7.86 \cdot 10^{-3})$
10	0.39 (0.36)	$-9.38 \cdot 10^{-3*}$ $(4.53 \cdot 10^{-3})$

standard error is given in brackets; significance level * $p < 0.05$

Table 8: The number of times a variable is significant and in what combination in the second step for the 2-step regression on the uplift during promotions, between the *constant* and *price* variables for the whole sample for the last week of the June 2021.

Significant Variable	Count
Constant	43726
Price	16397
Both	5466
Neither	116601

Additionally, we use the forecasts from the 2-Step Baseline model framework to create a new score for all products (Baseline Score) as explained in Section 5.3. Figure 6 shows how the Baseline Score is distributed in terms of how many products are in each score group, the mass that is 0 score is not illustrate which accounts for 70% of the products. We notice that distribution is similar to the results found for the Bayesian model.

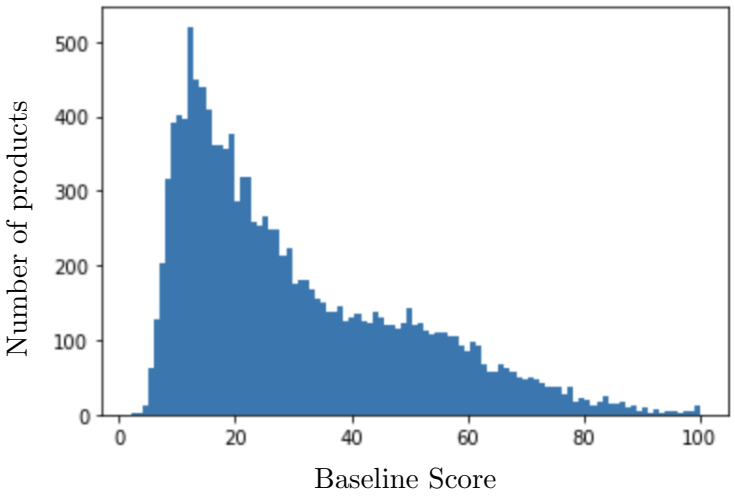

Figure 6: Baseline Score against the number of products in each score group excluding 0 during the validation period.

Next, we compare the Baseline Score against the average daily sales during promotions as can be seen in Table 5. We notice that there does not seem to be a clear correlation between the score and the sales during promotion. These suspicions are further substantiated by a correlation of $5.10 \cdot 10^{-2}$. This is lower than the correlation seen for the Relevance Score of 0.28 and 0.54 for the Bayesian Score. We suggest that the low correlation and loose parameter importance can be accentuated to the Baseline model having a low predictive power, due to a number of reasons. For example, that the restrictions set are too loose on what constitutes as a promotion for the 2-Step Baseline model. As a promotion that happens during 1 day of the week counts as a promotion for the baseline model the data has a lot of noise from the other days of week in terms of impact of promotion on sales. Therefore, the results are biased twice, as the model underestimates parameter importance of price for sales and predictions are made assuming the product is on promotion for the whole week, thus overestimating performance during the week. As a result, the predictive power of the model seems to be compromised.

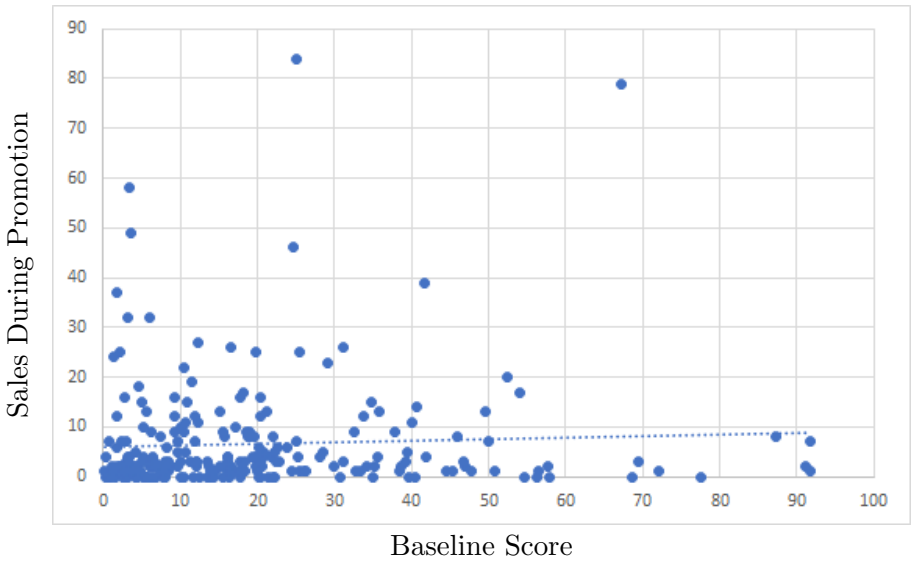

Figure 7: Baseline Score against the average daily sales of SSP during the validation period.

6.2.3 MSE

In this section we consider the MSE of the 2-Step Baseline model and Bayesian model to compare the performance between the models. Due to the Relevance Score not producing a forecast on sales of product we cannot create a MSE for this score. We consider the MSE over the Bayesian and 2-Step Baseline model over all data from the validation set, just over promotions in the validation dataset and only over the data points where no promotion was hold. The results can be found in Table 9. We see that the MSE is a higher for both models when there are promotions going on compared to when there are no promotions. This falls

within expectations as during promotions there are more products sold thus the possibility of getting a higher squared error increase. Not to mention, we denote that all promotions are the same kind, thus the models cannot use the magnitude of the promotion in terms of marketing on and off the Bol.com platform, the models predict for the average performance over past promotions. Thus the likelihood of getting bigger squared error increase.

Table 9: MSE of the Bayesian and 2-Step Baseline model over the whole, promotional, and non-promotional validation dataset.

Model	MSE Whole Period	MSE Promotional Period	MSE Non-promotional Period
Bayesian	1.80	10.46	1.60
2-Step Baseline	74.11	240.97	73.57

Furthermore, the Bayesian model scores better on all fronts compared to 2-Step Baseline model. Here we do consider that due to the difference between daily predictions and weekly predictions and the squared error nature, no real direct comparison can be made. For that reason, we create an additional MSE score for the Bayesian model by using a summation of the daily predictions to compare with weekly sales of products. This leads to a new MSE score for the Bayesian model for weekly predictions of 17.84 over the whole validation dataset. This score is substantially lower than 74.11 MSE produces by the 2-Step Baseline model. This gives a clear indication that the Bayesian model scores better than the 2-Step Baseline model in prediction of sales, granted that we take note that the 2-Step Baseline model has less data to make predictions with per product.

6.2.4 Computation time

Lastly, we look into the computation time of the models. Note that the computation time of the Relevance Score currently used is negligible. As the Relevance Scores just uses BigQuery the score is created within 5 minutes for all active 45 million products. Likewise, after the weighting is created the 2-Step Baseline model can create the predictions and scores for all products within 5 minutes through BigQuery. The matter of the computational load of this model is mostly defined by finding the feature importance. This happens relatively fast as there are few computations making the bottleneck of the algorithm being the retriebal of the data from BigQuery and storing the results back. As a result, the weights of all 182190 products over the whole validation set can be created within 597.23 seconds, which is close to 10 minutes, on a single machine from AI Platform. This boils down to that the computational time per product to create weights with retrieving and storing data takes $3.27 \cdot 10^{-3}$ seconds.

Likewise we can compute the computational time necessary to perform the Bayesian model. Again, using BigQuery we can create forecasts and scores within 5 minutes after the draws from the posterior are known. The bottleneck of this algorithm is that it is computationally expensive to perform all draws over the iterations of the Gibbs sampler and the including matrix multiplications. Performing the one step ahead forecast once for a product takes 2.83 seconds per product. In other words, to perform the algorithm for 10000 products for the 61 dates of the validation dataset with one step ahead forecasts, the current implementation took 480 computational hours. These computational hours were split between 20 GPUs on Ai Platform reducing the waiting time to less than a full day. To perform the algorithm on the whole database of 45 million products the current implementation would take over 2162500 computational hours. Which is hugely out of scope for this research. There are multiple ways to fasten up the implementation currently in place, which will be discussed in Section 8. As such the computational burden of the Bayesian model is high. Such that, even when only performing one step ahead forecasts to create daily forecasts, it would come down to over 36000 computational hours.

6.3 Performance Experiment

In this section we look into the results of the real-world experiment. We saw during the period of the experiment there were 545 partners which saw the scores and set promotions for their assortment, of which 268 were in the treatment group and 277 in the control group. In total there were 35923 unique products put into promotion, those consisted of 27392 in the treatment and 19044 in the control group. We note the number of products put into SSP per partner was not equally distributed, that is to say that the top 10 partners which put fourth the most SSPs, set up to 85% of all promotions. Furthermore, there were 443 partners which set less than 20 products into promotions.

The results of the experiment can be seen in Table 10, where we note that the permutation test has been performed with $m_{p\_max} = 10000$. We immediately notice that $\mu_{control} > \mu_{treatment}$, thus it is certain that treatment group did not outperform the control group on a significant level, but for the formality we perform the permutation test which leads to a significance level of 95%. Therefore, we cannot reject the hypothesis that the average number of products sold per day during SSP of the control group is greater or equal to the treatment group under the current treatment. This result falls in line with the other results seen of the 2-Step Baseline model where the Relevance Score outperforms it.

Table 10: Results of real-world experiment with average sold products per day (μ) for the control and treatment groups, with the test statistic and corresponding significance level of the permutation test on these outcomes.

	Measure
$\mu_{control}$	0.64
$\mu_{treatment}$	0.51
TS_e	-0.13
Significance Level	95%

Nonetheless, we argue that finding a significant level for this test would be hard in consideration of the imbalanced nature of data in terms how many products are put into promotion per partner, as the likelihood that these big promoters used the scores to set relevant products promotions is low. Therefore, we argue if the real-world experiment was hold for a longer time period, so that more partners could have joined or alternatively a possibility to restrict to set promotions of irrelevant products based on the variant shown, more meaningful results could be distilled from the experiment.

7 Conclusion

In online retailing the usage of product stock to determine promotions is often leading, but it is also important to determine the consumer relevancy of products for promotions such that irrelevant products do not take up marketing space. Therefore, the following research question is addressed: *"How can we rank consumer relevancy of products for promotion?"*. The main question is divided into: *How can we use sales data to estimate the effect of marketing mix variables on sales?"*, *"How can we adapt the sales model to deal with stockouts?"*, *"How do we use parameter estimates to make forecasts about relevancy of products for promotions?"* and *"What methods should we use to translate the forecasts of consumer demand into a ranking of products?"*.

To answer the first two sub-questions we describe the modelling of a Product-Specific Tobit Sales Regression Model in Section 5. With the usage of sales data and Bayesian modelling through Gibbs sampling, we deal with stockouts by sampling demand from the posterior distribution during stockouts. Additionally, the draws from the posterior distribution are used to find parameter importance in regard to the demand generating process. Secondly, we come forth with a Multi-Step Regression model that is used as a baseline for the research. The model uses the predictive forecast from Bol.com for non-promotional sales of products to create an uplift forecast for promotions on top of the standard predicted sales, however the model reveals drawbacks in regard to stockouts, due to making use of the weekly sales forecast from Bol.com

it cannot properly handle the input of stockouts. This is caused by the fact that stockouts are originally omitted from the Bol.com forecasting method and the simple regression nature of the model does not deal with it directly. Moreover, it has clear drawbacks in defining what constitutes as a promotion. As such, with the lack of data there is a lot of uncertainty within the creation of feature importance.

In regard to the latter two sub-questions, to create forecasts for the relevancy of products for promotions in Section 5 we come forth with a method to create forecasts through the Bayesian model based on the draws of posterior distribution with one step ahead forecasts. Subsequently, these forecasts are used within a linear ranking algorithm in which each product is scored on the forecast performance within the product group. This results in a new score which shows a higher correlation between ranking and performance during promotion than the currently implemented score by Bol.com. Similarly, the Baseline model is implemented to create one step ahead forecasts, from which the same scoring algorithm is used to score every product per product group. However, this method scored worse in MSE on all accounts compared to the Bayesian model, caused by the predictive power of the Baseline model being low. Consequently, the affiliated score is barely correlated with the average sales per promotion and scores worse than the Relevance Score of Bol.com in terms of finding relevant products for promotion in a real-world experiment.

All in all, to answer the main research question we conclude that the Bayesian model outperforms the two other scores in terms of predictive power to indicate what products are relevant for promotion. Nonetheless, there is a clear drawback to the model in terms of the computational time for practical usage for which we discuss possible remedies in Section 8. Additionally, we find that the data is better suited for daily predictions instead of weekly in this framework. This is due to low sample size, with the difficulty to specify what is a promotion when promotions are not held for the whole week and the difficulty to account for stockouts on a weekly basis leading to low predictive power. To summarise, we recommend the implementation of Bayesian model to create the new ranking of products, although this will require further development to optimise computation time. Afterwards the model can be tested through a new A/B test which is held over a longer period in order to get more partners to join and set promotions, which results in clearer results which are not dominated by few partners. Additionally, this research outlines how the model can be used for calculating uplift, the main KPI of Bol.com.

8 Discussion

In this section we debate the limitations found during the research. We discuss the resulting consequences of said limitations. Finally, based on the limitations we discuss possible directions for further research.

There were many limitations found during the research. As mentioned, during the research we decided that promotions are defined as a boolean which is either on or off. Resulting in that promotions given the price are denoted as all being the same. Whereas, in reality the difference between a promotion of 10% off compared to 40% off can have a big impact on the visibility of the promotion on the platform and the psychological aspect to the customer. Not to mention, these signals become even more lost in the noise of the data for the baseline model due to the weekly nature employed. For the baseline model, we opt to use any promotional data available due to the sparsity of promotional data available in the data set. Therefore, we suggest it is of interest to research the impact of promotions where we keep in mind the nature of the promotional activity. An example of a possible implementation is the usage of weighting coefficient on the data for promotional activity, where bigger promotions are weighted more.

Secondly, due to that the implementation of SSPs for partners of Bol.com was relatively new at the time of research, there is little data available of SSP. Since then more than a year has passed, giving more opportunity to specifically look at products with just the data for SSP. Or alternatively, using the framework of using all promotional data employed for this research, we argue that there is now the opportunity to take better care of the seasonality of products through the usage of quarterly boolean variables.

An additional point of discussion for the baseline model is that it uses the baseline sales forecast from Bol.com, which does not take care of promotions. Therefore, there are products that get an upwards biased baseline sales forecast, due to display promotion being always on in a given period. An example is Christmas trees during the Christmas period. These biased baseline sales are considered a limitation of the framework of using the sales forecast from Bol.com itself. A solution to this issue is to create an alternative baseline sales forecast model. For example, an ARIMA model or Neural Network as employed by Abolghasemi et al. (2020).

In terms of the Bayesian model we note that there are several limitations found during the research. First and foremost, we note that the implementation used in this research has a significant computation time. The long computation time comes forth from the Gibbs sampler, which has to a compute a large number of iterations before the algorithm converges, additionally only after convergence another set of iterations has to be done to create posterior results. This process is repeated for the whole validation period to create a moving window in which one step ahead forecasts are created. This process can be achieved faster by the usage of importance sampling for the moving window (Ritter & Tanner, 1992). Additionally, for daily usage we note that moving window is not required. As such, we have multiple suggestion to remedy the computational time necessary to implement the algorithm for daily usage. A first alternative is to not create daily one-step ahead forecasts but rather change the algorithm to be able to create multiple step ahead forecasts. This way not every day the whole algorithm has to run for the whole database. Furthermore, we suggest the algorithm to be rewritten to make use of multi-threading of the GPU through the likes of using big Python packages (Van Rossum & Drake, 2009). We suggest the usage of CUDA in combination with TensorFlow or Keras (NVIDIA et al. (2020); Abadi et al. (2015); Chollet & Others (2015)), as these packages are built to make use of the GPU, and optimal handling of large volumes of data, respectively.

Furthermore, we acknowledge there is an opportunity to decrease the amount of data, which would in turn decrease the computational time. The data set has a lot of data points which are similar due to that a majority of products do not get any views on most of the days nor any price changes. As a result, we suggest the implementation of weighting coefficients on similar data points. This would lead to a significant decrease in the amount of data, especially for products that have barely any sales. There are a lot of low selling products in the Bol.com database, thus we expect the computation time to decrease significantly through this method as well.

Moreover, the models implemented both deal with products on an individual level. This does not account for the interaction effect multiple products can have on each other by going on promotion, as one product being bought can cannibalise sales of the other product. Therefore for future research we suggest a multivariate regression model as next steps. Alternatively, to allow for the uncertainty in the parameters we suggest the implementation of a Hierarchical Bayesian model as an interesting extension on the current framework.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.*

Abolghasemi, M., Beh, E., Tarr, G., & Gerlach, R. (2020). Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion. *Computers & Industrial Engineering*, *142*, 106380.

Agarwal, D., & Chen, B.-C. (2009). Regression-based latent factor models. In *Proceedings of the 15th acm sigkdd international conference on knowledge discovery and data mining* (pp. 19–28).

Anupindi, R., Dada, M., & Gupta, S. (1998). Estimation of consumer demand with stock-out based substitution: An application to vending machine products. *Marketing Science*, *17*(4), 406–423.

Arora, N., Allenby, G. M., & Ginter, J. L. (1998). A hierarchical bayes model of primary and secondary demand. *Marketing Science*, *17*(1), 29–44.

Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., & West, M. (2003). Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian statistics*, *7*, 733–742.

Bhattacharya, A., & Dunson, D. B. (2011). Sparse bayesian infinite factor models. *Biometrika*, 291–306.

Chib, S. (1992). Bayes inference in the tobit censored regression model. *Journal of Econometrics*, *51*(1-2), 79–99.

Chollet, F., & Others. (2015). *Keras.* GitHub.

Cobb, G. W. (1998). *Introduction to design and analysis of experiments.* Springer.

Conlon, C. T., & Mortimer, J. H. (2013). Demand estimation under incomplete product availability. *American Economic Journal: Microeconomics*, *5*(4), 1–30.

Cornick, J., Cox, T. L., & Gould, B. W. (1994). Fluid milk purchases: a multivariate tobit analysis. *American Journal of Agricultural Economics*, *76*(1), 74–82.

Foekens, E. W., Leeflang, P. S., & Wittink, D. R. (1998). Varying parameter models to accommodate dynamic promotion effects. *Journal of Econometrics*, *89*(1-2), 249–268.

Fok, D., Horváth, C., Paap, R., & Franses, P. H. (2006). A hierarchical bayes error correction model to explain dynamic effects of price changes. *Journal of Marketing Research*, *43*(3), 443–461.

Gelfand, A. E., & Smith, A. F. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, *85*(410), 398–409.

Greenberg, E. (2012). *Introduction to bayesian econometrics*. Cambridge University Press.

Jain, A., Rudi, N., & Wang, T. (2015). Demand estimation and ordering under censoring: Stock-out timing is (almost) all you need. *Operations Research*, *63*(1), 134–150.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*.

Kim, J., Allenby, G. M., & Rossi, P. E. (2002). Modeling consumer demand for variety. *Marketing Science*, *21*(3), 229–250.

Lee, G., & Scott, C. (2012). Em algorithms for multivariate gaussian mixture models with truncated and censored data. *Computational Statistics & Data Analysis*, *56*(9), 2816–2829.

Letham, B., Letham, L. M., & Rudin, C. (2016). Bayesian inference of arrival rate and substitution behavior from sales transaction data with stockouts. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1695–1704).

Ma, S., Fildes, R., & Huang, T. (2016). 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*(1), 245–257.

Macé, S., & Neslin, S. A. (2004). The determinants of pre-and postpromotion dips in sales of frequently purchased goods. *Journal of Marketing Research*, *41*(3), 339–350.

Marquardt, D. W., & Snee, R. D. (1975). Ridge regression in practice. *The American Statistician*, *29*(1), 3–20.

Nijs, V. R., Dekimpe, M. G., Steenkamps, J.-B. E., & Hanssens, D. M. (2018). The category-demand effects of price promotions. In *Long-term impact of marketing: A compendium* (pp. 187–233). World Scientific.

NVIDIA, Vingelmann, P., & Fitzek, F. H. (2020). *Cuda, release: 10.2.89*.

Ozhegov, E., & Teterina, D. (2018). The ensemble method for censored demand prediction. *Higher School of Economics Research Paper No. WP BRP*, *200*.

Peinkofer, S. T., Esper, T. L., Smith, R. J., & Williams, B. D. (2015). Assessing the impact of price promotions on consumer response to online stockouts. *Journal of Business Logistics*, *36*(3), 260–272.

Pesaran, H. H., & Shin, Y. (1998). Generalized impulse response analysis in linear multivariate models. *Economics letters*, *58*(1), 17–29.

Ritter, C., & Tanner, M. A. (1992). Facilitating the gibbs sampler: the gibbs stopper and the griddy-gibbs sampler. *Journal of the American Statistical Association*, *87*(419), 861–868.

Stefanescu, C. (2009). Multivariate customer demand: modeling and estimation from censored sales. *Available at SSRN 1334353*.

Tobin, J. (1958). Estimation of relationships for limited dependent variables. *Econometrica: journal of the Econometric Society*, 24–36.

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. Scotts Valley, CA: CreateSpace.

Vulcano, G., Van Ryzin, G., & Ratliff, R. (2012). Estimating primary demand for substitutable products from sales transaction data. *Operations Research*, *60*(2), 313–334.

Wei, G. C., & Tanner, M. A. (1990). Posterior computations for censored regression data. *Journal of the American Statistical Association*, *85*(411), 829–839.

A Appendix

A.1 Customer Lifetime Value

From the business side of Bol.com there is a need to incorporate the CLV of products on marketing behaviour. Here the CLV is defined as a concept, where the value is implied to be the value the customer will bring to the platform in the future. As such, we would want to capture the causation between customer loyalty and product purchases. As CLV is seen as a driver for the growth of Bol.com. To determine products which classify products that drive the CLV, for each product the Product Loyalty Index (PLI) is determined. The PLI is created by measuring how many more orders are placed on average by customers after buying the given product compared to before, this index is aggregated for the desired level of detail, next it is divided by the average of all products. The PLI is measured on multiple levels, stretching from the specific product to the chunk to the store level.

Figure 8: Customer loyalty before and after purchase for different categories of products, resulting four quadrants: the Loyalty Boosters, Loyalty Stars, Loyalty Laggards and Loyalty Base, respectively.

To better understand the PLI, Figure 8 shows what the PLI captures. On the x-axis the average amount of orders customers make before making a purchase in that category of products is given and on the y-axis the difference in average amount of orders customers do after the purchase compared to before the purchase is given. This creates four quadrants wherein each category of products can be classified. The quadrants are specified as Loyalty Boosters, Loyalty Stars, Loyalty Laggards and Loyalty Base. The Loyalty Boosters and the Loyalty Stars are the most interesting groups as they increase the numbers of products sold over time. When we relate the PLI to the CLV of customers, Bol.com wants to put more importance on products which have a high PLI. As such, we want to use this information to score products with a high PLI with a better score for the relevancy of products for promotion. Although, this is out of scope

for this research. Additionally, there are other known actions customers can take that are highly correlated with a higher CLV, these include customers downloading the app, purchases products from a different cluster of products, registering for the email news service and taking a subscription on Select. Select is the customer loyalty subscription programme of Bol.com, which gives benefits to customers, for example, free deliveries on any purchase price point and free deliveries on Sundays or evenings.