Effective Forecast Enrichments

Driving Factors Behind Enrichment Quality

Eggels, Loek L.

Award date:
2023

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Effective Forecast Enrichments

Driving Factors Behind Enrichment Quality

Operations Management & Logistics

Author:	Eggels L.J. (1233960)
University Supervisors:	van Donselaar K.
	Imdahl C.
Company Supervisors:	van der Staak B.
	Roozemond D.

Eindhoven, July 5, 2023

Abstract

This thesis explores the field of behavioral operations management with a focus on planners' enrichments and accuracy. The literature shows mixed results regarding the planners' ability to add value to forecasts. Even though, a large number of organizations use statistical forecasts with human enrichments to predict upcoming demand. Within this thesis, we utilize a dataset from a company in the process sector with the aim to discover the significant features of effective forecast enrichments to guide planners in making more accurate enrichments. Findings from the literature are confirmed since the real data reveals that planners are more likely to overestimate sales volumes. When planners do make an enrichment that is less than the statistical forecast, they are more likely to improve the forecast accuracy. Planners vary from one and another when it comes to accuracy, but they are consistently less accurate when dealing with products with a high variance We also find that planners are exposed to optimism, overreaction, and anchoring biases. A prediction model is created based on LightGBM with automated feature generation with 14 base features. We find that enrichment size, the statistically forecasted quantity, and the previous forecast value add are the most important indicators of enrichment quality. Lastly, notifications are explored to see if the model could support planners to improve both the forecast accuracy and their forecast value add. Guidelines for the implementation of notifications are discussed.

Management Summary

Introduction

Within the field of behavioral operations management, results regarding human forecast enrichments are mixed. In theory, human forecast enrichments are able to combine the stable statistical forecasts with the human cognitive flexibility. Within this thesis, we investigate the dataset of a company to identify what the feature importances are of effective forecast enrichments and when to notify the planners in order to prevent harmful enrichments. Five research questions will be answered to find the results. First, literature about forecast enrichments is explored. Then, appropriate machine learning models are selected. As the majority of the literature is focused on the pharmaceutical and consumer products industry, it is interesting to see if the results also show up within this dataset in the process sector. Through conversations with service owners within EyeOn, a better understanding of the business context is created. This will improve the interpretability of the results. Then the models will be created, tuned and tested and afterwards, the importance of different features is evaluated to generate actionable insights towards the prevention of harmful enrichments. To achieve these results, we are answering the following research questions:

RQ1: *What patterns do planners exhibit when adjusting system-generated forecasts and how does this influence the overall forecasting accuracy?*

RQ2: *How can tree-based models and neural networks be applied to estimate enrichment quality and how can their results be explained?*

RQ3: *What characteristics from human forecast enrichments that are described in the literature show up in this dataset?*

RQ4: *What features are of great importance to the quality of forecast enrichments?*

RQ5: *In what conditions could planners be notified about their expected enrichment quality?*

Literature Review: Behavioral Operations Management

Within the field of behavioral operations management, the way in which humans operate in forecasting is well researched. Demand planners often anticipate higher sales volumes than actually occur. This is due to their overoptimism bias (Fildes et al., 2009; Eroglu & Croxton, 2010; Syntetos et al., 2009; Trapero et al., 2013). Enrichments that predict less volume than the statistical forecast are often well-founded and will most likely be accurate. However, humans are only inclined to adjust the forecast downwards when they have a clear piece of unmodeled information, but this does not occur frequently. Furthermore, demand planners are more accurate when they are dealing with stable demand patterns compared to very noisy ones (Sanders, 1992) and there is a lot of variance between planners. The propensity of a forecaster to enrich a forecast varies between different sectors, with lower levels in the retail sector (Fildes et al., 2009; Khosrowabadi et al., 2022). To expand upon the overoptimism bias stated previously, human planners have limited cognitive abilities which inhibits them from being rational decision makers. Next to overoptimism, they are also exposed to a range of other biases, like anchoring and overreaction biases. To reduce these biases, one can advice planners in their decision

making. However humans are also hesitant to incorporate advice, especially when this advice comes from an algorithm (Dietvorst et al., 2015).

Literature Review: Machine Learning

There are a large number of machine learning models that can be applied. Within this thesis, we implement a neural network and a LightGBM model. The neural network is selected since it is able to incorporate difficult and intricate non-linear relationships between the predictor and target variables (Chollet et al., 2015). The LightGBM model is an extremely popular model through its high performance and accuracy (Ke et al., 2017).

Exploratory Data Analysis

The company studied within the thesis consists out of 3 BUs, each with their own characteristics. The performance within this company is in line with the literature through its mixed results. In BU 3, planners are improving forecasting accuracy, while they deteriorate the accuracy in the other BUs. Planners within the BUs are also clearly biased when it comes to Anchoring Bias, Overreaction Bias and Optimism Bias. Planners tend to not anchor on a statistical forecast. Instead, they overreact and are overoptimistic about expected demand. This results in over-forecasting. Planners follow the advice to more often adjust high-value, high-variance products (Scholz-Reiter et al., 2012). However, performance also strongly varies between planners and they enrich forecasts better when dealing with more stable demand patterns.

Results Prediction Model

There are a total of 14 basic features, spread across 4 different categories. These are fed into both the LightGBM and Neural Network models under various conditions. The models are tested with data scaling, clipping of the target variable, certain combinations of BUs and, several steps of automated feature engineering. In the end, a LightGBM model, with scaled data, clipped target and a single step of feature engineering provides the best results for a dataset only including BU 1 & 3.

Through the implementation of Shapley values, we are able to rank the importance of several features. In descending order, the Enrichment Size, Statistical forecast, Previous FVA are strong indicators of enrichment quality. Furthermore, we include ABC-XYZ analysis from Scholz-Reiter et al. (2012) to create the product categorization. Certain product categories like BY and CZ are strong indicators of bad enrichment quality. Overreaction bias is the bias that has the strongest presence.

Notifications

The study by Dietvorst et al. (2015) shows people often distrust computer programs when they make mistakes. To avoid false warnings, we need to be careful when alerting planners. Our data shows many adjustments that lower the FVA. Considering our model's behavior, it would not be a good idea to warn every time it predicts a negative FVA. This could result in too many false alarms. To tackle this, we are exploring certain cut-off points. We can use Shapley plots to understand these low predictions and decide a suitable cut-off point based on planners' comfort with predictions and trust in our model. For all explored thresholds, intervening by utilizing the statistical forecasts leads to a baseline of performance that is higher than both enriched and statistical forecasts.

Contents

List of Figures

List of Tables

1 Introduction

To put it bluntly, all forecasts are wrong. But, for companies it is of great importance to create accurate forecasts. Accurate forecasts increase efficiency, reduce waste, and costs (Sanders & Manrodt, 2003; Lin et al., 2014). Often, organizations utilize a statistical forecast as a baseline, which is enriched afterward (i.e., adjusted by a planner) to increase accuracy. Forecast enrichments could allow planners to incorporate unmodeled external events within the forecast. In theory, this method combines the stable, predictable factors of a statistical forecast (e.g., naive forecast, moving average, exponential smoothing), with the flexibility of the mental ability to incorporate factors unknown to the statistical model. However, letting humans tinker with the forecasts does expose the forecasts to the cognitive limitations of the human mind, known as biases (Eroglu & Croxton, 2010; Fildes et al., 2009; Trapero et al., 2013).

1.1 Company Description

The project will be executed at EyeOn. EyeOn is a consultancy firm based in Eindhoven (*EyeOn*, 2022). From this location, the organization has been expanding rapidly since the late 90s and is now based in 6 offices worldwide. The other locations are Antwerp (Belgium), Geneva and Zurich (Switzerland), Dublin (Ireland), Dusseldorf (Germany) & New York (USA). Their team consists of more than 110 experts, from more than 20 countries and with a total of more than 500 years of experience. EyeOn's employees concern a group of forecasting and planning specialists with the mission to realize impactful results that get your business years ahead.
Core challenges are to: i) increase customer service levels, ii) raise forecasting performance, iii) balance inventories, and iv) reduce waste and save costs. Their expertise areas concern: end-to-end transformation, sustainable value chains, future supply chain design, S&OP/IBP, Forecasting & Demand management, supply planning, and inventory optimization.
They are specialized in five different fields: sales forecasting, demand management, sales & operations planning, integral business management, and inventory management. The specialist working at EyeOn provide additional services like interim planning, planning systems, and visual insights, turning data into actionable insights.
EyeOn's client base consists out of large national and multinational companies in the process, life science, consumer products, high-tech, and marine & offshore industries. Examples of such companies are Johnson & Johnson, Heikenen, Jumbo, and Etos.

1.2 Problem Context

The process of forecasting can vary among different organizations. Some frequently used forecasting methods are judgmental forecasts, statistical forecasts, advanced Machine Learning (ML) and Artificial(ly) Intelligent (AI) models, and models in which a human enriches the statistical baseline forecast. Especially the latter is often employed among companies since it theoretically combines both the rigor and consistency of statistical methods while creating flexibility to incorporate unexpected or irregular events. However, human forecast adjustments also include potentially damaging human biases. While some enrichments improve the accuracy, others can deteriorate it, which undermines the rationale for this approach. EyeOn often encounters this phenomenon among customers who want to avoid these bad enrichments. There lies great potential in being able to label forecast enrichments that are expected to reduce the accuracy based on their features. Prevention of damaging forecasts could significantly reduce costs and improve efficiency.

1.3 Literature Gap

There has been a large flow of information created by literature over the last 30 years concerning human forecast enrichments. Different aspects have been evaluated, concerning the accuracy of judgmental forecast enrichments, the accuracy of statistical forecasting methods, and aspects of human interaction with machines and statistics (e.g., algorithm aversion). Most papers purely focus on the way in which humans adjust forecasts without regard for the features of the respective enrichment. They tend to center around the human and not around the enrichment. Some recent papers like Khosrowabadi et al. (2022) have identified specific product characteristics that would drive the enrichment performance, but did not broadly explore specific planner-related features. Furthermore, many papers (Fildes et al., 2009; Trapero et al., 2013; Khosrowabadi et al., 2022) focus on forecast enrichments within the retail sector, while there are other sectors like the process industry. There are differences between these sectors in the fact that the retail sector is more promotion driven, is focused on the B2C market, and has different manufacturing processes. However, forecast enrichments are also prevalent within the process industry, and it is yet unclear if the results are generalizable.

1.4 Scope

Within this thesis, we will focus on a quantitative analysis to accurately predict the forecasting accuracy through enrichments of planners. These enrichments will consist of features that can be expressed quantitatively based on the setting or previous behavior. One could take a qualitative approach in predicting the enrichment accuracy. It is common practice for planners to add comments to their enrichments. These comments give planners the opportunity to express their reasoning. In order to properly assess and incorporate these comments, approaches within the field of Neural Language Processing (NLP) are required. We will not be considering this in this thesis.

1.5 Research Questions

To guide the thesis, the aim is to resolve the following research questions. The first two questions are based on literature research. The third research question verifies the findings from the literature within the dataset used for this thesis. Then, the fourth question explains the results created by the prediction model while question five interprets the features and transforms these into tangible advice for EyeOn and its customers.

RQ1: *What patterns do planners exhibit when adjusting system-generated forecasts and how does this influence the overall forecasting accuracy?*

RQ2: *How can tree-based models and neural networks be applied to estimate enrichment quality and how can their results be explained?*

RQ3: *What characteristics from human forecast enrichments that are described in the literature show up in this dataset?*

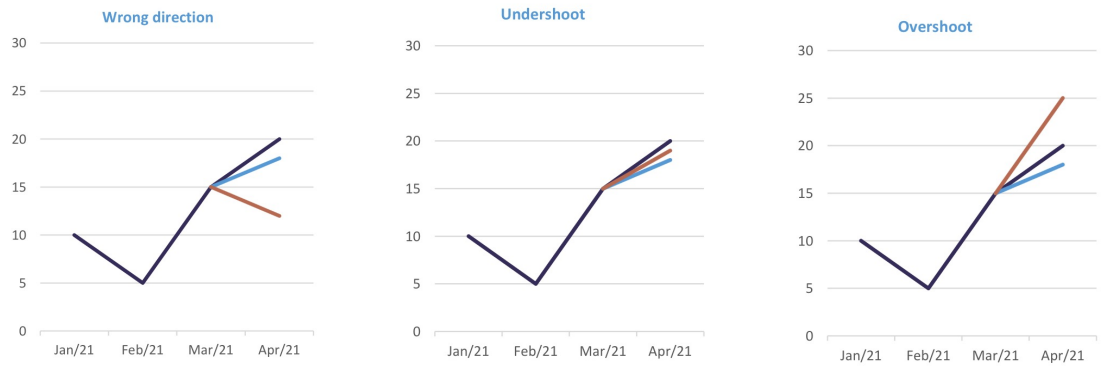
RQ4: *What features are of great importance to the quality of forecast enrichments?*

RQ5: *In what conditions could planners be notified about their expected enrichment quality?*

2 Literature Operations Management

2.1 Forecast Enriching

Petropoulos et al. (2016) defines three different adjustments that a planner can make when adjusting the forecast (wrong direction, undershoot, and overshoot). Within Figure 2.1 the different kinds of adjustments are visualized. The dark blue line indicates the actual demand, the light blue line indicates the statistical forecast and the orange line indicates the final forecast after a human enrichment.



(a) Wrong Direction (b) Undershoot (c) Overshoot

Figure 2.1: Three kinds of forecast enrichments

First, and most detrimental, are wrong direction adjustments. These are adjustments that are in the incorrect direction from the statistical forecast. Such an adjustment always reduces the forecast accuracy of the final forecast and should be avoided at all costs. Unfortunately, small adjustments in the wrong direction are very prevalent in a business environment. The second type of adjustment is an undershoot. Such adjustments occur most frequently and always increase the forecasting accuracy. An undershoot is an adjustment that has been made in the correct direction and reduces the forecast inaccuracy, however, the adjustment has not been large enough to completely negate the error. Small adjustments constitute a total of 57% of all adjustments which show that planners are inclined to only adjust the statistical forecast slightly (Petropoulos et al., 2016). The last kind of adjustment is a so-called overshoot. This is an adjustment in the correct direction but of a too large magnitude. Small overshoots are often beneficial to the overall accuracy since they can be very close to the actual demand. However, large overshoots can be disastrous for the overall accuracy and indicate that something has either gone wrong or at least not as planned.

2.2 Enrichment Quality

Several studies have researched how forecast enrichments are executed and what their results are.

Firstly, the propensity of planners to enrich forecasts is a widely researched topic. It seems that the sector in which a planner operates is the strongest indicator of this propensity. The papers by Fildes et al. (2009); Franses (2013); Baecke et al. (2017); Baets & Harvey (2018) all find that planners are extremely likely to enrich forecasts, with probabilities ranging from 70% up to 98% of all forecasts. Franses (2013) also found that average enrichment sizes were extremely large ranging from 70% to 130% of the statistical forecast.

However, this does not hold true for other sectors. The paper by Fildes et al. (2009) has a dataset that combines data from organizations in different sectors. The data from organizations in the retail sector show a much lower percentage of enrichments, slightly above 10%. The same holds true for the paper by Khosrowabadi et al. (2022), situated in the retail sector, in which only about 5% of forecasts are enriched. Within the retail sector, there are often higher order frequencies and a higher number of SKUs. Thus, there are often too many forecasts to be enriched by hand. Within the retail sector, the propensity to adjust does increase when dealing with perishable/fresh products.

The proposed increase in accuracy by enriching forecasts is not clearly seen. Syntetos et al. (2009) found an improvement in accuracy in 61% of the forecasts, while Petropoulos et al. (2016) and Goodwin et al. (2007) only saw improvements around the 50% mark. The influence on the total forecasting accuracy does differ among studies. For some, the total forecasting accuracy keeps hovering at the same level after enrichments. Some studies report slight decreases in accuracy (Petropoulos et al., 2016) (-2.1%) and others report minor increases (Goodwin et al., 2007) (+0.37%). Broeke et al. (2019) stated that the accuracy of enriched forecasts was in no situation beneficial and Belvedere & Goodwin (2017) reported a 20% decrease in accuracy. Sanders & Ritzman (1995) found an increase in accuracy again by enriching forecasts.

Since enriching forecasts consists of two complementary aspects of both the stability of statistical forecasts with the flexibility of human judgment, the complementary strengths should be utilized correctly. According to Goodwin (2002), in the situations in which humans adjust forecasts, for best accuracy, they must have a clear rationale for the reason why they adjust the forecasts. In general, humans should only adjust the forecast when they obtain contextual information that is not integrated into the system-generated model. One example is the classic 'promotional activity' in which a product is priced at a discounted rate to boost sales for a limited period (Lee et al., 2007). In such a situation, it would be wise to expect a higher sales volume, and thus a higher number of products would be forecasted to satisfy customer demand and increase forecast accuracy. This is proven to indeed be the case by Trapero et al. (2013).

Different studies found uniformly that different adjustment directions and sizes have different effects on forecasting accuracy. To start, adjustments upwards are more prevalent in forecasting compared to downward adjustments (Fildes et al., 2009; Goodwin, 2002; Syntetos et al., 2009; Trapero et al., 2013; Broeke et al., 2019). In addition, these adjustments are also larger which further increases average forecast sizes. When getting closer to the forecasting deadline, the adjustments are more frequent, large in size, and more often in an upwards direction. Planners seem to react with higher levels of volatility when coming closer to the actual moment of sales. Trapero et al. (2013), Fildes et al. (2009), Broeke et al. (2019), Khosrowabadi et al. (2022) and Syntetos et al. (2009) all found that downwards adjustments are more likely to be beneficial to forecast accuracy.

However, given clear indications of an external event like a promotional activity, upwards adjustments can increase the accuracy (Trapero et al., 2013). This study also showed that when a forecast was enriched upwards, it was more likely that it would have a significant negative impact on the overall forecasting accuracy.

This binary classification used in literature was refined by Khosrowabadi et al. (2022) who designated also the sizes of up- and downward adjustments. They found that only

39.8% of large upwards adjustments are valuable, while 66.6% of small upwards adjustments are beneficial. The total accuracy of all upwards adjustments was reduced by a very significant 22.7%. Fildes et al. (2009) found that it does also matter for which industry the forecast is made. Their subset of retailers improved forecast accuracy with each negative adjustment since they were consistently over-forecasting. There are very limited situations in which an upwards adjustment turns out to increase the accuracy. Amongst others, these concern situations in which the statistical forecast expects 0 sales, after which the smallest upwards adjustments lead to large gains, and negative adjustments are not possible.

Syntetos et al. (2009) further confirms that negative adjustments between 50% and 100% can lead to extremely high gains in accuracy. As stated before, the size of the judgmental adjustment also often influences the quality of the improvement. Depending upon the industry in which the forecast is made, most often, the smallest 25% of adjustments lead to a reduced forecast accuracy (Fildes et al., 2009).

From this section, we can see the consensus that planners have a high propensity to enrich forecasts, given they have enough resources to do so. Furthermore, planners are more likely to adjust forecasts upwards. Large upwards adjustments are more often damaging compared to small upward adjustments. However, when planners adjust forecasts downwards, they should aid the accuracy, on the basis that there are no clear external influential factors like promotional events.

2.3 Enrichment Factors

As seen in the previous section, the ability of a planner to effectively enrich forecasts depends upon many situational factors. This section explores what specific factors could influence the quality of forecast enrichments.

The first factor that could influence a planner's enrichment quality is the utilization of causal information. Lim & O'Connor (1996) stated that human planners are very capable to select important variables for enrichment, and can adjust them appropriately. However, on the other hand, Goodwin & Fildes (1999) stated that humans are not up to this task by over-weighting external cues and neglecting the statistical baseline forecast.

Also, Broeke et al. (2019) found that the time lag in a forecast enrichment also influenced planners' behavior, with more frequent and significant changes being made when the time lag was lower.

Another enrichment factor could be the type of product that is being forecast given certain levels of variance and sales volume. Scholz-Reiter et al. (2012) created a framework in which all the products can be placed. The parameters of the framework are shown in Figure 2.2. If a product is in the top 80% of the highest revenue, it would get labeled by an A. Given its coefficient of variation, it would be assigned to a certain column. If the coefficient of variation would be higher than 1, it would be assigned to AZ. One can also see the two columns of NPI (New Product Introduction) and EOL (End Of Life). For these products, there is not sufficient accurate/recent sales data available, and thus are they very hard to forecast using statistical methods. Given the proposition that statistics work well in situations where the variance is low, one would recommend categories in the lower left-hand side (low value, low variance) to be forecasted purely by statistics while products

on the top right (high value, high variance) to be left to human supervision.

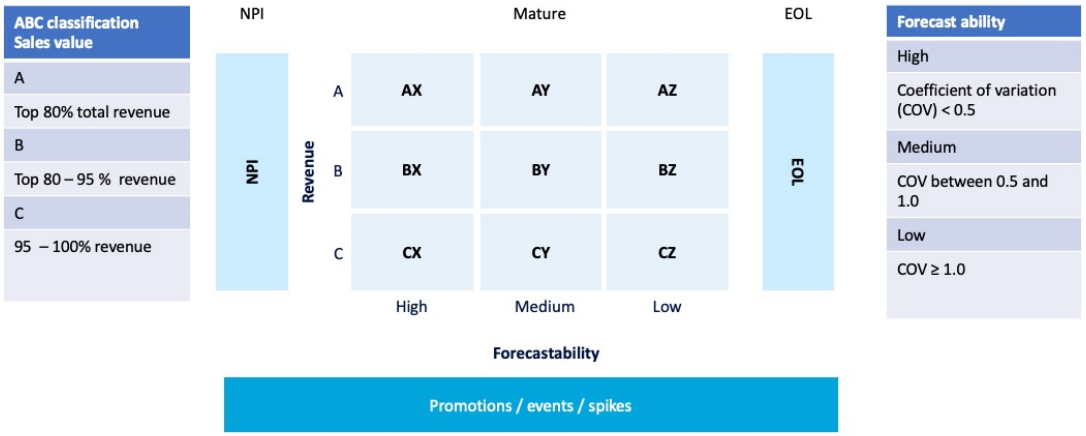


Figure 2.2: Product Category Division

Lastly, Khosrowabadi et al. (2022) investigated what kind of product-specific features influence a forecaster's enrichment ability. These features, in descending order of importance, turn out to be price, perishability, promotions, and weather. An interesting note is the fact that both promotions and weather were already taken into account by the forecasting system in this paper, and thus, planners suffered from a double-counting bias.

2.4 Variance among Planners

Given the factors that influence enrichments, next one could look at the variance between different planners to see if they significantly differ from one another and why.

Eroglu & Croxton (2010) found that basic demographics are not related to the accuracy of forecast enrichments. However, there are clear differences between different forecasters when it comes to quality, with some being able to outperform statistical methods, while others consistently perform significantly worse (Sanders, 1992; O'Connor et al., 1993). The delta between good and bad forecasters did not evolve during the experiments, indicating a limited learning potential (Syntetos et al., 2009; Lim & O'Connor, 1996). Franses (2013) revealed that many planners do not have a clear view of their forecasting behavior, often forecasting more than expected. Emotional product involvement also decreases a forecaster's ability to forecast accurately (Belvedere & Goodwin, 2017).

2.5 Limitations of Planners

As stated before, planners have significant cognitive limitations (Lee et al., 2007; Goodwin, 2002). In the past, it was assumed that people were rational decision-makers but this turned out to be incorrect (Tversky & Kahneman, 1974).

Noisy time series are often extremely difficult to forecast. Humans tend to try and match patterns within the noise which is methodologically incorrect. This often results in reduced forecast enrichment quality (Sanders, 1992; Franses, 2013). Combined with the

product categorization based on the value and variance, it should be interesting to see if human planners are able to enrich forecasts of these product categories in the dataset.

Next to that, people can be stubborn and neglect advice. Both Goodwin & Fildes (1999) and Petropoulos et al. (2016) support this and explore why humans exhibit this behavior. Reasons for this are that planners can be subject to asymmetric loss, either in an upward or downward direction given situational factors. For instance, if management only evaluates planners on the number of stock-outs of products, planners will generally order more to prevent stock-outs. However, this behavior will lead to overstocking many products and increased inventory costs, potentially outweighing the benefits of having no stock-outs. Furthermore, incentives can be misaligned that elicits certain wrong and non-desired behavior. In addition, special events can be integrated into the forecast, but forecasters can neglect advice on the timing. Lastly, planners can make extreme interventions based on misinterpreted information and keep their views even though they are proven to be wrong.

Planners will not only discard advice given by other humans but also the advice that is given by an algorithm. This links to the so-called 'algorithm aversion' that is explored by Dietvorst et al. (2015). Humans seem to be less accepting of advice when it has been provided by either mathematical or AI methods. The main drivers for this behavior are that algorithmic behavior can be harder to explain to laymen (Franses, 2013), people believe in the ability of others to learn while they do not for algorithms, and humans seem to 'punish' algorithms harder than other humans. These behaviors remain after a human has been exposed to the superior performance of the algorithm.

The cognitive limitations to which planners are exposed are known as 'biases'. The most famous research on this topic was executed by Tversky & Kahneman (1974) and further expanded upon by Eroglu & Croxton (2010). The most important biases that were put forward were the so-called optimism bias, overreaction bias, and anchoring bias. A planner exhibits an optimism bias when forecasts are more often adjusted in an upward adjustment, in combination with a large adjustment size. A planner expects to sell an unreasonably high number of products which is incorrect. This bias has been highly prevalent in various datasets (Fildes et al., 2009; Khosrowabadi et al., 2022; Trapero et al., 2013; Sanders & Manrodt, 2003). A second kind of bias is the overreaction bias in which a forecast is adjusted in the correct direction, but the magnitude of the adjustment is significantly too large which increases the forecasting error but in the other direction. Such adjustments are also prevalent in industry (Eroglu & Croxton, 2010). Lastly, there is often an anchoring bias in which humans influence their decision too much based on previous sales, forecasters, or certain types of information. It has been shown several times by Sanders (1992), Baets & Harvey (2018) and Tversky & Kahneman (1974). It is also prevalent after promotional periods, in which forecasters are not able to appropriately adjust the forecasts due to the performance during the promotion.

3 Machine Learning Review

The goal of machine learning (ML) models is to automatically identify meaningful relationships among different variables or identify patterns within a dataset (Bishop, 2006). Machine learning models are trained by iteratively adjusting the applied algorithms based on the training data that it has been fed. Thus, considering good data quality and tuned hyperparameters, the performance of the model should increase given more iterations of

improvement. This allows the model to identify hidden relations in very complex data without it having to be programmed explicitly by a human being (Bishop, 2006). Such models can thus be powerful for modeling intricate and complex tasks with high-dimensional data for classification, regression, and clustering (Janiesch et al., 2021).

3.1 Learning Methods

There are different methods by which a machine learning model can fit itself to the data. Each method has its benefits and drawbacks and can be applied to different situations. In total, there are three kinds: 'supervised', 'unsupervised', and 'reinforcement' methods.

In supervised models, training data needs to consist of an input and a 'label', also known as output or target. Pairs of the input and output data are utilized to calibrate the parameters of the ML model. Once the model has been made, it can be utilized to predict a target variable y based on unseen input data (Janiesch et al., 2021). Such models could be utilized for time series forecasting like done by Khosrowabadi et al. (2022). Based on historical input and output data (e.g., different product features as input and sales quantity as output), one could make a supervised ML model to predict upcoming sales, utilizing unseen input parameters. Within the supervised machine learning models, a division can be made between regression problems (where a numerical value is predicted) and classification problems (where a certain class is predicted).

Unsupervised models are not trained based on a given target, but they are used to identify patterns within the data. Often such models are used for clustering certain groups of data. Reinforcement models learn through a process of trial and error, where they receive feedback in the form of rewards or punishments to maximize their performance in a given environment or task. These two models are not aligned with the objective of this thesis and will not be explored further.

3.2 Tree-based Models

3.2.1 Decision Trees

A decision tree is a method that is the foundation of many other tree-based machine learning methods. It can be used for both regression and classification. The goal of a classification decision tree is to take an unorganized 'bag' of data and sort it into several classes that are unique from each other. A decision tree consists of several separate components. Based on the paper of Quinlan (1986), we start with the nodes. These can either take the shape of a root (node), decision node, or leaf (end node). At each node, a decision is taken based on the properties of the sample of data that is fed to the node. The nodes are connected by branches, which represent the path that one should follow to purify the class. The root node concerns the entire set of inputs originally provided, and the other nodes concern subsets of the original data, specified based upon previously taken decisions. The leaves are ending nodes of the tree in which the classes are identified as 'pure' in a classification tree or as a finalized numerical result in a regression tree. No further adjustments or specifications are made from this point onwards.

The model has to choose questions to split the data as efficiently as possible. There are several metrics used to describe the purity of the class, with a common measure

being entropy. Decision trees that base themselves on entropy are called ID3 (Iterative Dichotomiser 3) trees. In Equation 3.1, the formula for entropy is displayed, with p_i being the probability of a class 'i' in the data.

$$E(S) = \sum_{i=1}^{c} -p_i log_2 p_i \tag{3.1}$$

The scale ranges from 0 to 1, with 1 being a high level of impurity (e.g., highly unsorted data). When the data in a node represents only one class, it can be considered pure and the entropy value is 0. To uncover the decision on which the data will be split within a node, the metric information gain is utilized. Information gain (IG) (displayed in Formula 3.2) measures the reduction in entropy, or the increase of purity, given a new piece of information (i.e., a decision). The goal is to maximize the information gain, which corresponds to a minimization of entropy.

$$IG(Y, X) = E(Y) - E(Y|X) \tag{3.2}$$

When dealing with larger, more complex trees that are fed incomplete or unfiltered data, it might be hard to classify all inputs as cleanly. In such a situation, you need certain 'stopping criteria'. This prevents the tree from overfitting the data and increases general applicability. Stopping criteria could be: a) using a minimum number of records per leaf, b) a minimum number of instances in a node before splitting, or c) setting a maximum tree depth.

Utilizing decision trees has several benefits and drawbacks. To start with the benefits, decision trees can be very simple to comprehend. The methods utilized by them are straightforward, which on the one hand limits its potential, but increases its acceptance by laypeople. Furthermore, decision trees can easily be utilized for both regression and classification problems, making them applicable in many situations. On the other hand, the decision-making mechanism can be simple which leads to incorrect decision making. Furthermore, the decision tree can achieve a high depth which increases its calculation complexity and decreases its understandability (Jijo & Abdulazeez, 2021). As a main takeaway, one should remember that the ultimate challenge in decision tree creation is to strike the balance between flexibility and robustness.

3.2.2 Gradient Boosting Trees

A large number of methods can be utilized to create decision trees. A certain group of these methods is called 'ensemble' methods and they can be very powerful by utilizing several decision trees in a single model. The main two methods are random forests (RF) and gradient boosting (GBDT). Gradient boosting will be explored further, since they can achieve the highest accuracy and performance when tuned properly. Through these advantages, gradient boosting has gained its popularity. Performance is very high which is due to its balance between complexity and applicability. Gradient boosting works by iteratively fitting negative gradients.

To show what this means, we will run through the basic steps a GBDT takes. First, take a labeled dataset and average all labels. This average output value will be the first prediction or a base from which the tree operates. One has to add a new column in the dataset next to the output, which is called 'residual' (or the negative gradient). This residual value is the difference between the actual output and predicted output and indicates how 'wrong' the prediction is. This residual is extremely important since the input factors are

trying to predict this value. So, given a dataset, and an initial prediction which is the average of all outputs, a GBDT tries to predict residual value through a decision tree. The number of leaves in the tree must be smaller than the number of residuals. This will create situations where several instances end up together in a leaf. When there are several instances in a leaf, the average value of the residuals is taken. However, to create well-fitted trees, this prediction is multiplied by a 'learning rate'. This learning rate is utilized to conservatively update the predictions made by the tree. This concerns one cycle, to better fit the tree, this cycle will be repeated several times, and through several decision trees.

However, this process can be very computationally intensive and would not be practical in real life. Thus, algorithms have been created to reduce this computational intensity. Ke et al. (2017) have created a methodology called 'LightGBM'. It utilizes two solutions to reduce the number of data instances.

Gradient-based One-Side Sampling (GOSS) is used as the first solution. Within GOSS, larger gradients, are weighted much more since these are most damaging to the overall result. It takes the highest a% of gradients and randomly selects b% from the remaining data. The sampled data of the small gradients will be multiplied by a constant $(1-a)/b$ to calculate their information gain. Thus, the instances with a large gradient will be focused upon, while the smaller gradients will get compensated for their reduced data instances and retain the original data distribution. Taking into account the values of the gradients, which will greatly outperform the data reduction method through random sampling.

Exclusive Feature Bundling (EFB) is the other proposed solution. While in real datasets, there are often many features available, the feature space is normally quite sparse. In other terms, there are more features reported than features that exist. In turn, if an event occurs, certain features will act in the same manner, while other features will operate in the opposite manner (i.e., they show signs of mutual exclusivity). EFB has the ability to thus reduce these (almost) exclusive features in a sparse feature window and thus increase computational performance.

3.2.3 Hyperparameter Tuning

To set up a LightGBM model, a large number of hyperparameters can be tuned to optimize model performance. According to an article by Bex (2021), there are several categories in which these parameters can be placed: structure & learning, accuracy & speed, and overfitting. The guide created by Microsoft (n.d.) also explains how to tune parameters, what their function is, and what values are advised. Within Section 5.3.2 regarding the methodology, we will take a deeper dive into parameter optimization.

3.3 Neural Networks

Another type of ML model will be described in this section, neural networks. Compared to tree-based models, these are designed in a vastly different ways, have different use cases, and have different strengths and weaknesses. This section of the thesis will explore how these models work, what variants there are, and what their respective (dis)advantages are.

3.3.1 Feedforward Neural Network

To fully understand the concept of neural networks, one should have skills in and knowledge of the fields of mathematics, biology, neurophysiology, cognitive science, physics, and many more. The neural network tries to bridge the gap between standard machine learning approaches and the brain of a living animal. Machines can quickly outperform humans when it comes to highly complex or fast calculations, while they are no match to a human brain in complex perceptual problems. Humans can learn from examples, be adaptive and fault-tolerant, and can be robust in complex tasks. A neural network operates by mimicking the neural networks found in humans (Zou et al., 2018; Jain et al., 1996).

Like a decision tree, a neural network also consists of nodes. However, in a neural network, a node performs a simple mathematical operation on the input that it is provided. In the simplest sense, it would be a single perceptron that performs a weighted sum of its inputs and applies a non-linear activation function to provide its output. Through the non-linear activation function, nodes can add non-linearity to the model which can make it more complex and accurate. Several kinds of functions can be used for activation (Softmax, Sigmoid, ReLU). Examples are shown in Appendix A.1.

The nodes are components of layers, of which there are three types. The first type is the input layer. These nodes receive and pass through the input data to the next layers. The second type is the output layer, in which the output of the neural network is provided. Between the input and the output layers, the hidden layers are situated. These layers are the most interesting since they apply mathematical transformations of their respective input data. Each neural network has a minimum of one hidden layer, but the more layers it has the higher complexity it will exhibit (Jain et al., 1996).

Nodes between the different layers are connected through connectors. In a feed-forward neural network, these connectors only connect nodes between different layers with a downstream flow (i.e., they cannot connect nodes within one layer). The connectors can have different weights. Weights influence the importance of certain nodes, with a higher weight leading to higher importance. Furthermore, bias can be introduced in the connectors. This bias is added to the input before the mathematical transformation is executed within the node. The goal of including this bias is to better fit the data. It does this by moving the prediction on the axes (Jain et al., 1996).

The basic process of the neural network is visualized in Figure 3.1. This shows a small neural network, with a singular input and output. In between, there are two hidden layers. We can see input X1 traveling across the connector, being adjusted with the respective weight and bias. The output of this connector is called x1' since it has been adjusted from the original x1 value. This is fed into the activation function, which will provide a y value. This y value will be sent across the three connectors toward the next hidden layer. In the nodes in hidden layer 2, multiple input connectors come together. For them, x' values from the connectors are summed together before being put into the activation function. This also holds true in the output node, where three inputs are being summed.

Neural networks have to be trained in order to achieve good results. In the learning process, the weights and biases are tuned based on their gradients. This will be done through the process of backpropagation. Backpropagation consists out of two phases, the forward phase and the backward phase. In the forward phase, data inputs are passed through the network in order to calculate the node outputs. In the backward phase,

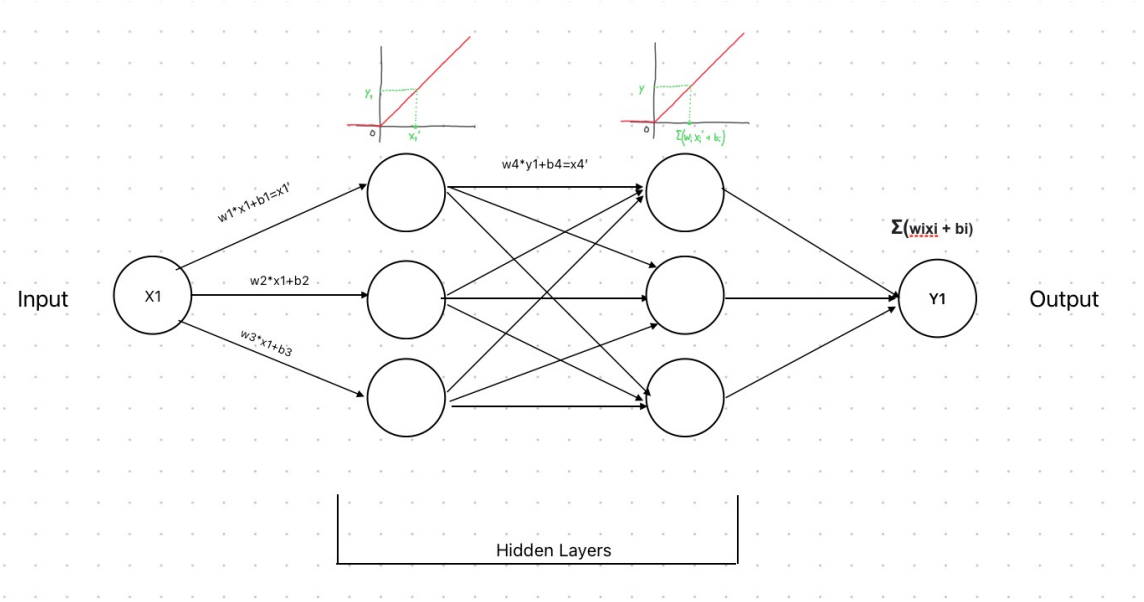


Figure 3.1: Concept Neural Network

gradients of the loss function compared to the weights of the network are computed.

The two main ideas of backpropagation are: i) to use the chain rule in order to calculate derivatives and ii) plugging these derivatives into gradient descent to optimize parameters. For a deeper dive into the mathematics, Appendix A.2 explains the backpropagation process.

3.3.2 Hyperparameter Tuning

Like LightGBM, neural networks offer a large number of methods by which they can be adjusted. This section will be based on the article by Radhakrishnan (2017). The first important hyperparameters are the number of nodes and layers that the neural networks encompass. Both the number of hidden layers and the number of nodes will influence the level of over- or underfitting. Increasing this will increase precision at the risk of overfitting. To combat this, one could use the dropout technique in which neurons are canceled from the network, making the model more general. Rates between 20% and 50% are recommended. In addition, the weights in the network are usually initialized according to a uniform distribution, giving them the same weight. However, it is possible to adjust this given different activation functions on different layers of the network. The learning rate is another highly important parameter in the system. This rate indicates the speed at which a neural network updates its parameters. A low learning rate leads to smooth convergence of the network, while a larger learning rate could increase the speed of learning, at the risk that the network will never converge since it overshoots constantly. The number of epochs represents the times that the full training data is shown to the neural network in the training phase. A higher number of times could increase precision at the risk of overfitting.

3.4 Fitting of Machine Learning Models

Previously, we stated that the hyperparameter tuning should create a model that does not over- or underfit the data. The goal of our model is to perform well on unseen

data based on relationships and patterns it finds in the training set. However, there is a trade-off since the training data often includes noise (i.e., patterns that are not actually there). When you include this noise, the model will perform very well on the training set, but worse on unseen data. On the other hand, underfitting occurs when the model is too simplistic and is unable to extract the nuances of the training data. It is unable to capture relevant patterns which will lead to poor performance on both training and testing data. Ideally, one would strike a balance between over- and underfitting.

To test whether a hyperparameter configuration shows over- or underfitting, one can check the error measure for predictions on the training and testing sets. When the training data performs much better than the testing data, we see overfitting. When the training data performs about as badly as the testing data, we see underfitting. In Figure 3.2 we can see a visual representation of the concept of over- and underfitting.

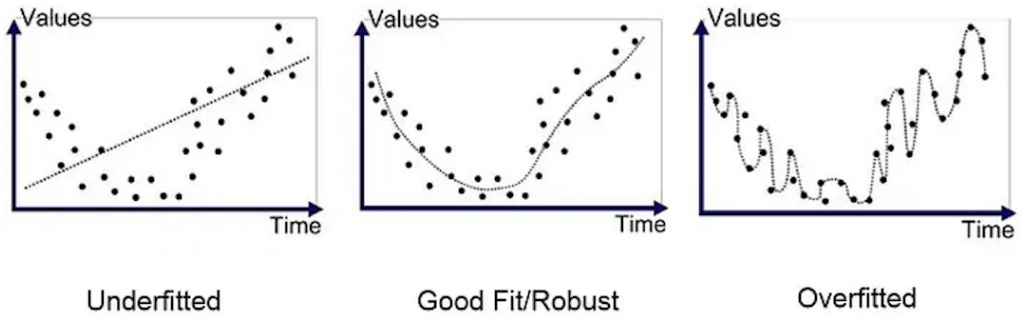


Figure 3.2: Example of Fitting

3.5 SHAP: SHapley Additive exPlanations

As discussed before, ML models are capable to create decisions to efficiently classify data. ML models are often referred to as 'black boxes', to which one can only provide input, and only receives one output. The steps in between can be hard to fathom for an ordinary user. Furthermore, such models can incorporate systematic errors. A classic example of which could be an ML model that tries to classify if the animal in the image is either a wolf or a dog. Often, the model was able to correctly classify wolves as wolves and dogs as dogs. However, it stumbled when it was provided with the picture of a dog in the snow (Figure 3.3). It turned out that wolves are more often seen in snowy environments. Thus, the model was just very good at classifying snow. If an ML model is not closely monitored or understood, it can very easily develop these systematic errors (Ribeiro et al., 2016). This is something that should also be prevented within the context of this thesis.

(a) Input Image (b) Explanation of Decision

Figure 3.3: XAI Example

A popular method in the field of explainable AI (XAI) is SHAP (SHapley Additive exPlanations) based on the work of Shapley (1952). Although his work is based in the field of cooperative game theory, it found new uses within AI. In SHAP, each feature can be interpreted as a player in a cooperative game. The contributions of each feature to the overall outcome can be regarded as its own importance. It shows the default prediction that the ML model makes, together with all the features that made the model alter its prediction. Thus, the user will be able to understand the rationale behind a single decision.

4 Setting and Data

Within this section, we will look at the dataset that is used for all analysis and research in this thesis. Since the dataset concerns real company data, we will also explore the business environment from which the data is extracted. This context will improve the clarity of results later in the thesis. Furthermore, the data cleaning will be explained in this section.

4.1 Research Setting

The dataset provided for this thesis originates from a large multinational company in the process industry. The company will be referred to as 'Company A' in the remainder of this thesis. Within Company A, data from 3 Business Units (BU) (1, 2 & 3) is utilized. Since these BUs all stem from the same company, they sell roughly the same products, however, they are aimed at different industries and therefore show different characteristics. The BUs operate like separate entities.

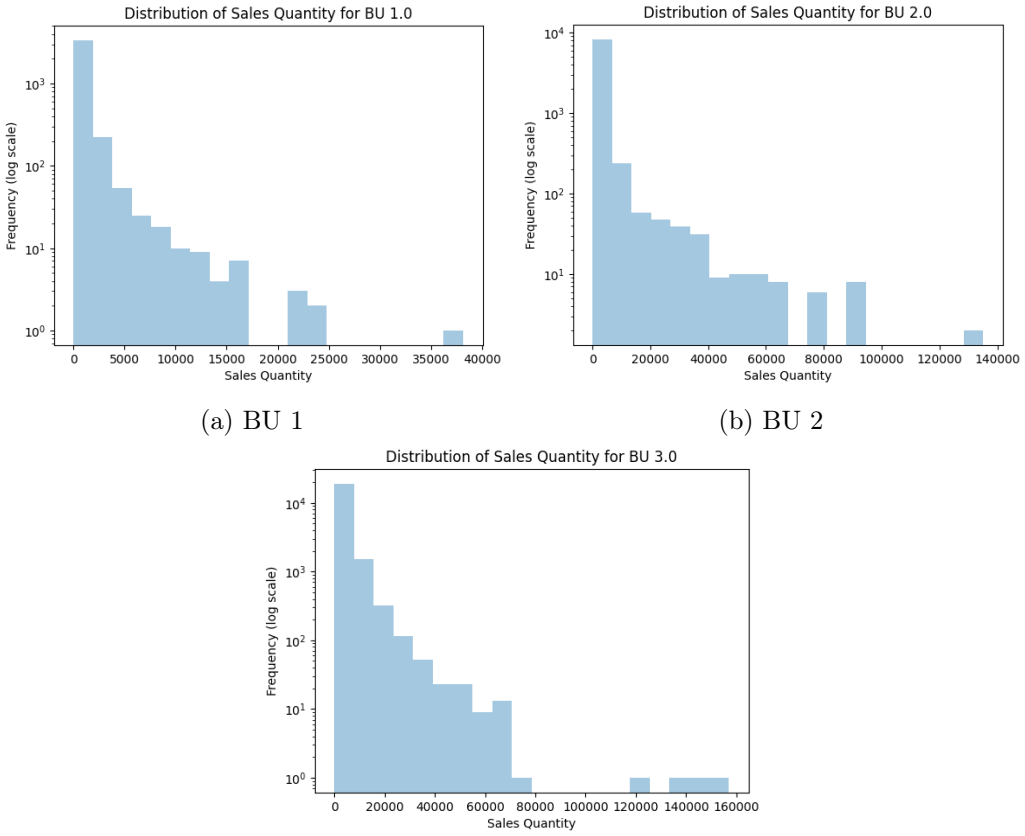
In total, Company A serves 1204 customers with 5,888 products, further specified in Table 4.1. To show the broad range and skew of the data, it also shows the mean, median, minimum, and maximum sales volumes per BU. Within this table, one can clearly see that each BU is skewed to the left, with a long tail to the right. This means that there are a lot of low sales volumes, but the average becomes large due to some very large sales. The sales quantities are also plotted in Figure 4.1. These figures use a logarithmic scale to show the long tail.

BU	Products	Customers	Mean	Median	Minimum	Maximum
1	936	230	776.82	236.55	1.18	38,140.00
2	1923	106	2,170.78	350.00	1.31	135,080.00
3	2788	847	2,993.12	1180.00	2.00	156,975.00

Table 4.1: Descriptives per BU

Company A uses EyeOn's services to help them in forecasting. Every month, EyeOn provides statistical forecasts and business insights to help them improve their forecasting performance. After the planners receive the forecast, they can adjust the forecasts in software called Jedox. Planners are linked to a BU and only enrich forecasts within this specific BU. Before looking further into the data, we should set up some assumptions about the planners on which this thesis and the results of it are based:

- Planners strive towards creating forecasts with the highest possible accuracy



(a) BU 1

(b) BU 2

(c) BU 3

Figure 4.1: Sales Quantity Distribution per BU

- Forecasts that are created are unconstrained (if the planner at Company A knows they are unable to deliver 300 units of product 1, they will still order 300 if they deem this to be the right quantity)
- Planners also have no gain to stock up warehouses for future demand and thus they do not do this.

The first assumption recognizes the underlying goal of planners. The second assumption indicates that planners have the freedom to create the forecast without external constraints or limitations, which is essential for the importance of their ability to judge the situation. The last assumption indicates that planners have no incentive or motivation to build stock. Planners are thus purely focused on creating the best possible enrichment regarding accuracy.

4.2 Data Description

Given we are working with company data, the creation of a clean and useful dataset can be challenging. Each BU has separate datasets for its master data, categorization of products, and performance. All enrichments are stored in a different database. Joins between these databases have to be made to incorporate data from each BU. After merging and stacking all data, we can connect enrichment values, statistical forecasts, sales quantities, planner IDs, and many more variables.

In total, there is data based on 237,996 enrichments (Table 4.2). However, a large number of these cannot be used for analysis. Firstly, some products have a missing sales quantity (34,117). Sales quantities can be missing for a variety of reasons, among which are the discontinuation of a product, or a change in hierarchy. Furthermore, if Company A decides that certain branches (e.g., a certain region or customer group) do not need the statistical forecasts from EyeOn, their sales information is no longer shared, while an enrichment created previously could still be reported within the database. Such enrichments are deleted since they cannot be evaluated based on their accuracy. Furthermore, there are a large number of products (24,958) that are part of the product categories 'EOL' or 'NPI'. For these products, there is too little data to accurately tune a statistical forecast, necessary to predict an accurate sales quantity. Next, there are also a lot of sales quantities (65,354) that are less than 1. In most cases, the sales quantity will be 0. However, sometimes, the sales are only slightly above 0. Given that errors for these enrichments can be exponentially large, and will misrepresent the overall performance of a forecaster, these are also removed. Next, there are several duplicate rows (70,952) in the database due to many merges and different product categorization levels. These categorizations levels can change over time and thus it will report all categorizations it has ever found. Only the latest categorizations are selected, corresponding to the moment of the last enrichment for a certain month. Some planners also only make a small number of adjustments (less than 50). Planners with such low statistics only adjust forecasts incidentally and are removed from the dataset (1,765). Lastly, the dataset is corrected for outliers (310). Data points are selected to be outliers when they are more than 1.5 standard deviation higher or lower than the mean.

Total enrichments	237,996
Missing Sales	34,117
EOL/NPI	24,958
Sales quantity <1	65,354
Final Forecast = 0	8,147
Duplicates	70,952
Deleted Planners	1,765
Outlier Correction	310
Final	32,393

Table 4.2: Data Cleaning

4.3 Performance Overview

Diving deeper, we will look into the actual performance measures. It can be measured in several different ways and from different perspectives. Within Table 4.3, all abbreviations and variables are labeled.

We will briefly explain potential error measures, before selecting one for further analysis. Firstly, we will investigate FA based on % MAE, which is defined by the Formula 4.1. This measure checks the absolute error or deviation ($e^i_{t,k}$) in units between the sales quantity and the forecasted quantity. This is then divided by the sales quantity (A^i_t). FA can be calculated for each i which represents the perspective from which the accuracy is measured ($i \in \{Planner, BU, Category\}$). Furthermore, FA can also be measured for different kinds of forecasts which are represented by k ($k \in \{Enriched, Statistical\}$).

This method is robust because smaller errors on small sales quantities are summed together with larger errors on larger sales quantities. Thus, one identifies the total absolute error of the forecast and then divides them based on the total sold quantity. This ratio only defines the 'error' of the data. To transform this to 'accuracy', one has to start at one and subtract this error ratio. In turn, when a forecast is perfectly accurate, it would result in a value of 1. However, if the error ratio is larger than 1, the FA can become negative. This method is also utilized by EyeOn to measure and evaluate forecast accuracy.

$$FA_k^i = 1 - MAE_k^i\% = 1 - \frac{\sum_{t=1}^{n} |F_{t,k}^i - A_{t,k}^i|}{\sum_{t=1}^{n} |A_t^i|} \tag{4.1}$$

$$FVA^i = FA_{Enr}^i - FA_{Stat}^i \tag{4.2}$$

Another important measure is the FVA, which is also known as the 'Forecast Value Add'. This metric will compare the accuracies of the statistical and enriched forecasts. This metric has its center at 0 and a positive value will indicate an increase in accuracy, while a negative value will show the opposite.

Next to FA, there are a lot of other measures that could be used with different properties. Three other metrics will be described below. Compared to previously, the metrics measure error and not accuracy. In turn, a lower value will indicate a better result for any of these three measures.

$$RMSD_k^i = \sqrt{\frac{\sum_{t=1}^{n}(A_t^i - F_{t,k}^i)^2}{n}} \tag{4.3}$$

The first alternative is the RMSD (Root Mean Square Deviation) represented by Equation 4.3. This measure squares the errors of each forecast and divides the sum of these by the number of enrichments to get the average value. The RMSD is proportional to the size of the error, and thus very sensitive to large outliers. It will also indicate a non-negative value, with a value of 0 indicating perfect accuracy. Another point of note is the fact that the RMSD is scale-dependent and thus difficult to compare with other metrics.

$$sMAPE_k^i = \frac{1}{n} \sum_{t=1}^{n} \frac{\left|F_{t,k}^i - A_t^i\right|}{(A_t^i + F_{t,k}^i)/2} \tag{4.4}$$

The second alternative would be the sMAPE (symmetrical Mean Absolute Percentage Error). It is seen as an alternative to the normal MAPE but improved towards the MAPE's largest shortcoming, its asymmetric property. The sMAPE limits errors for over-forecasting to prevent this asymmetric property from showing. However. sMAPE is still sensitive to lower sales quantities and errors are disproportionally large. Furthermore, the

sMAPE does not penalize large over-forecasts effectively, and thus promotes over-forecasting.

$$MAPE_k^i = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t^i - F_{t,k}^i}{A_t^i}\right| \tag{4.5}$$

The last alternative is the most familiar and most often used: MAPE (Mean Absolute Percentage Error). MAPE has been known for its limitations and is often not the best method to measure errors. Just like sMAPE, it is sensitive to low sales quantities. Furthermore, its asymmetric property makes it so over-forecasting can exceed values of 100% while under-forecasting cannot.

$$Bias_k^i = \frac{\sum_{t=1}^{n}(F_{t,k}^i - A_t^i)}{\sum_{t=1}^{n} A_t^i} \tag{4.6}$$

Next to error measures, it is also important to define how we measure the bias of a forecast. The bias shows if a forecasting method is either over- or under-forecasting. In Formula 4.6, the bias definition is shown. The bias measure ranges from -1 in a situation where no products are forecasted, up to infinity. The bias measure is thus also inherently asymmetric. When the bias value approaches zero, we can consider the forecast to be unbiased.

Metric	Definition
i	Perspective: Planner, BU, Product Category
k	Type of Forecast: Enriched or Statistical
t	Period of time for which the metric is calculated
n	Number of observations
FA_k^i	Forecast Accuracy for Perspective i, using Forecast k
$MAE_k^{i\,\%}$	Percentage Mean Absolute Error
FVA^i	Forecast Value Add for Perspective i
$RMSD_k^i$	Root Mean Square Deviation for Perspective i, using Forecast k
$MAPE_k^i$	Mean Absolute Percentage Error for Perspective i, using Forecast k
$sMAPE_k^i$	Symmetrical Mean Absolute Percentage Error for Perspective i, using Forecast k
$F_{t,k}^i$	Forecasted Quantity for Perspective i, using Forecast k, at time t
A_t^i	Actual Sold Quantity for Perspective i, at time t
$e_{t,k}^i$	Forecasting Error in Units for Perspective i, using Forecast k, at time t
$Bias_k^i$	Bias for Perspective i, using Forecast k

Table 4.3: Summary of Metrics

4.3.1 BU - Perspective

Since the BUs operate separately from each other, this section will also report the performance per BU.

Within Table 4.4 the FVA levels are reported on a BU level. We can see one BU (2) that exhibits very different behavior from the other two. BU 2 has been known to be difficult to forecast due to the nature of the product. Customers of this BU often work on project basis, with large projects starting unpredictably. This unpredictability relates to

BU	Statistical Forecast	Enriched Forecast
1	0.37	0.24
2	-0.59	-2.82
3	0.50	0.51

Table 4.4: FA: Comparison between Statistical and Enriched Forecasts per BU

both the time and location of the project. Once the job starts, there needs to be definitely enough products in order to finish the project. Therefore, enrichments within BU 2 do not contribute positively to the FA compared to the statistical forecast. Later, we will look into this topic further. BU 1 starts off with a more accurate statistical forecast, but also experiences a decrease in accuracy through forecast enrichments. However, this reduction is much less dramatic. Last, BU 3 has the best benchmark for their statistical forecast at 0.50. Their accuracy increases slightly through enrichments to 0.51.

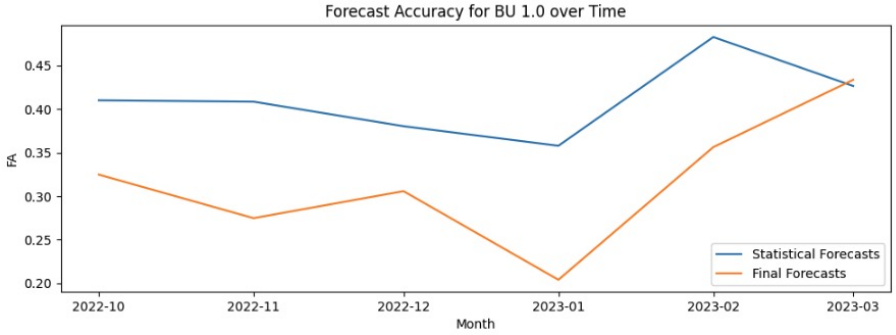
	RMSD		sMAPE		MAPE		MAE%	
BU	Stat Fcst	Enr Fcst	Stat Fcst	Enr Fcst	Stat Fcst	Enr Fcst	Stat Fcst	Enr Fcst
1	1,381.55	1,781.92	0.75	0.69	1.63	1.96	0.63	0.76
2	12,789.90	34,752.47	1.05	1.15	12.93	31.70	1.59	3.82
3	3,948.22	4,041.99	0.65	0.56	2.60	3.34	0.50	0.49

Table 4.5: Comparison Error Measures

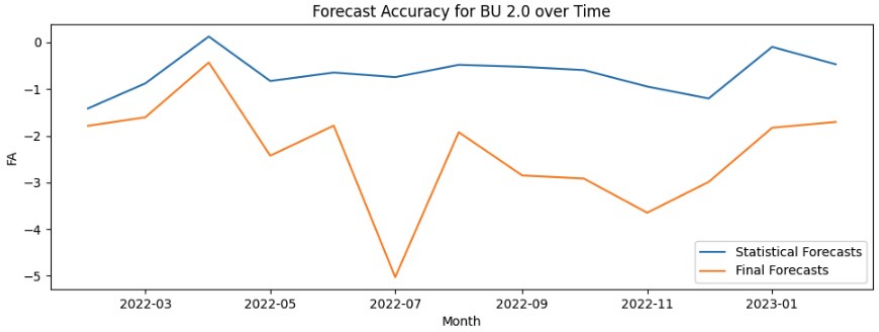
Within Table 4.5, the statistics for the alternative measuring methods are reported. It is interesting to see how results and in turn conclusions would change based on the use of a certain error measure. Under RMSD, each BU would decrease the accuracy of their forecasts through their enrichments. This situation is also shown under the MAPE. However, within the sMAPE, something interesting happens. Both BU 1 and 3 seem to be able to improve their forecasts through enrichments, while BU 2 only reduces their accuracy by a slight margin. Since we know that the sMAPE is not punishing over-forecasting as harshly, together with the improved enriched forecasts, we can see that all BUs have a tendency to over-forecast. Especially for BU 2, their error is a bit under 3 times the size in other measures, while in the sMAPE it is only about 10%. For the remainder of this thesis, we will continue to employ the FA as defined in Equation 4.1.

Looking further, it would be interesting to see how the FA is affected over time. Figure 4.2 compares the FA over time for each BU for both forecasts. For each BU, the length of the dataset varies. Some BUs have data for about a year, while BU 1 only stores results for the last 6 months, with older performance being discarded. If we plot the graphs for both statistical and enriched methods, we can see two BUs where the FA is decreased consistently over the time-series, namely BU 1 and 2. Especially in BU 2, the performance of the enrichments is rather erratic. During July 2022, BU 2 shows a very low performance in their enriched forecasts. This is due to some very bad enrichments combined with a low sales quantity. For BU 3, the FA of enriched forecasts hovers around the statistical forecasts. In some months, the statistical forecast performs better, in other other months, the enrichments perform better.

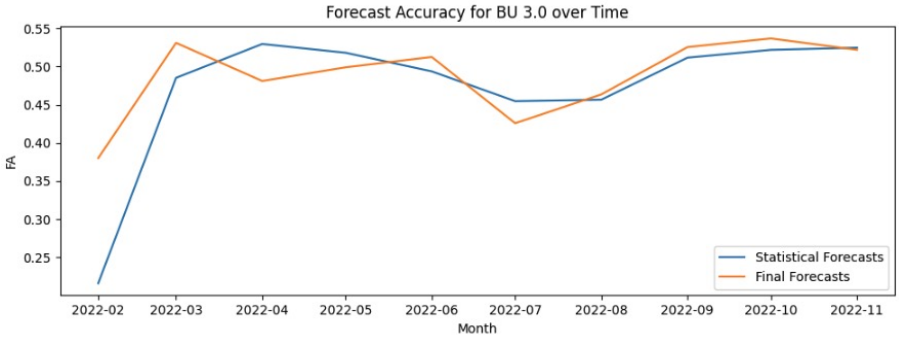
Now we have seen the performance of each respective BU, we can look if the forecasts are biased. Both statistical and enriched forecasts can be biased in different ways. A



(a) FA Comparison BU 1



(b) FA Comparison BU 2

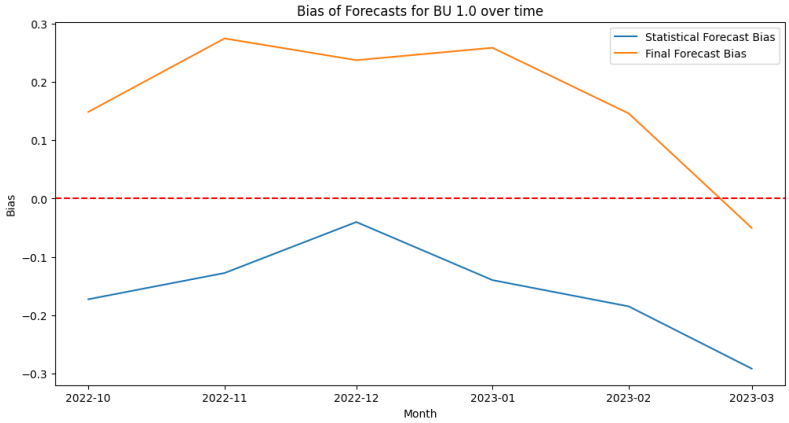


(c) FA Comparison BU 3

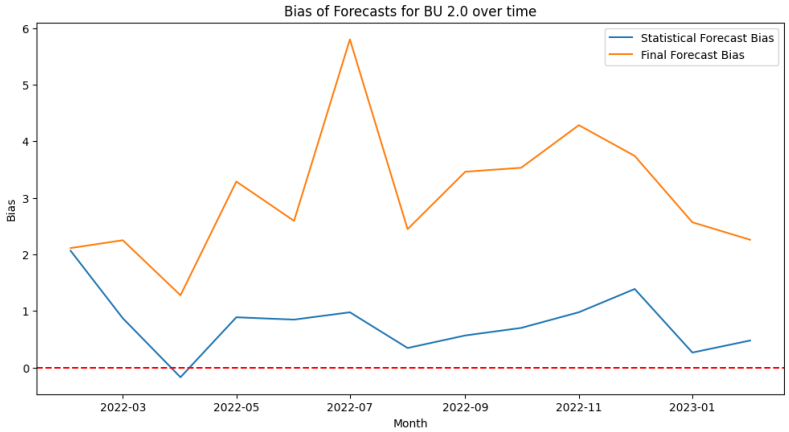
Figure 4.2: Comparison of FA in BU

statistical forecast can be considered to be biased if it constantly predicts either too much or too few products. If this is the case, the parameters behind the statistical forecasts might be inaccurately tuned, causing this behavior. Enriched forecasts can be biased since they are exposed to human behavior and in turn to human cognitive limitations. In Figure 4.3 we will look at the bias per BU.

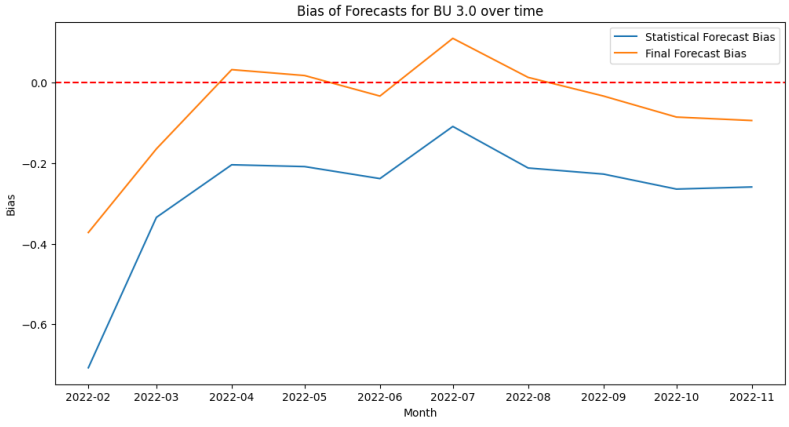
In Figure 4.3, we can observe several things. Firstly, the statistical forecasts for BU 1 and 3 seem to be slightly pessimistic. For these BUs, the statistical forecast predicts fewer sales than occur. After these forecasts have been enriched their bias seems to reduce. Especially the enriched forecasts of BU 3 hover around 0, indicating almost no bias. BU 1 becomes slightly optimistic. On the other hand, BU 2's performance could be explained by



(a) Bias Comparison BU 1



(b) Bias Comparison BU 2



(c) Bias Comparison BU 3

Figure 4.3: Comparison of FA in BU

its bias. The statistical forecast for this BU also consistently over-forecasts and this is only amplified after enriching the forecasts. Human planners thus adjust the forecasts in the wrong direction, which causes a higher error.

The performance of BU 2 seems to be incredibly low (Figure 4.2b) and strongly biased (Figure 4.3b), which could raise some questions. One could wonder if the data is correct if something major happened (e.g., product failure, pandemic, production issues) or if something went wrong within the reporting. We can see this BU over-forecasting excessively, over every month, and with large dips in performance for certain months. The BU works on a project basis, at a large scale, and with a tight schedule. Thus customers, require a large sales volume at very short notice. Such projects can also not afford to be delayed once they have been started. It would be a financial disaster if the customer starts the project, without enough quantity to finish it. Projects can often also be canceled last minute. This results in a sales quantity of 0 at the given customer. However, we cleaned the data for this as reported in Section 4.2. Enrichments with a final sales quantity of zero are removed from the dataset since one cannot calculate error terms for this. Conversations with the responsible person at EyeOn (i.e., the service owner) cleared up why these statistics could very well be correct. Within Jedox, planners are able to enrich products at several hierarchy levels. At the highest level, products can be enriched for an entire region, while at the lowest level, an enrichment concerns a specific customer key. Higher level enrichments are proportionally disaggregated to end up at customer-product level. The aggregation level between the forecast and the reporting differs for some arbitrary reason. Thus, the reported data within the dataset looks at the data at a higher level than just product-customer. Once combined with this uncertain project-based nature of sales, this can lead to such large errors. To give an example: imagine you sell your product (called 'product 1') within Region A. Region A consists of 10 customers that can demand this product. Customer 1 orders 2 units of product 1 every month while the remaining customers normally demand 0. For the upcoming month, we expect there to be a large project at customer 4, expected to sell 100 products. Thus, the total demand for this region would be 102 for the upcoming month. However, last minute, this project is canceled while the products have already been forecasted. The sales quantity now drops to 2 for the region and the forecasted amount remains at 102. This would lead to a FA of -51. This phenomenon also occurs in BU 2 creating these large errors.

4.3.2 Category - Perspective

As seen in Section 2.1, products are divided into 1 of 9 categories based on their value and variance. Products that are high in variance and high in value are recommended to be carefully examined by hand before adjustments are made. It is common advice to not enrich products that have both a low variance and a low value, since statistical methods should suffice. We see that in general, planners are indeed more inclined to adjust forecasts or products that are more valuable and variable, as is advised. The added value to the forecast differs greatly between different BUs. As seen before, the performance of BU 2 is significantly lower compared to the other two. For BU 1 and 3, the FVA is very close to zero for pretty much all product categories. BU 1 shows good results for categories CX and AZ. It does seem that planners' expertise improves the FVA in this area. BU 3 improves the FVA in CX and AY categories. Their improvements in CX are very good, however, there are only a very small number of adjustments (35). It is interesting to see that their performance for CZ products is particularly low. Given its low value, planners might not spend a lot of effort on forecasting this highly variable category of products.

All BU	X	Y	Z
A	-1.22	-0.76	-0.44
B	-1.12	-2.53	-1.11
C	0.17	-2.19	-0.99

Table 4.6: FVA Categorization of all BUs

All BU	X	Y	Z
A	4029	11644	3491
B	422	5154	3412
C	38	677	1414

Table 4.7: # Adjustments Categorization of all BUs

BU: 2	X	Y	Z
A	-3.61	-2.16	-1.39
B	-3.30	-7.35	-3.26
C	-	-6.42	-2.00

Table 4.8: FVA Categorization BU 2

BU: 2	X	Y	Z
A	349	1774	1422
B	24	472	902
C		159	490

Table 4.9: # Adjustments Categorization BU 2

BU: 1	X	Y	Z
A	-0.03	-0.15	0.10
B	-0.01	-0.17	-0.05
C	0.11	-0.04	-0.09

Table 4.10: FVA Categorization BU 1

BU: 1	X	Y	Z
A	452	1180	588
B	120	524	504
C	3	91	247

Table 4.11: # Adjustments Categorization BU 1

BU: 3	X	Y	Z
A	-0.01	0.02	-0.02
B	-0.04	-0.07	-0.00
C	0.39	-0.11	-0.87

Table 4.12: FVA Categorization BU 3

BU: 3	X	Y	Z
A	3228	8690	1481
B	278	4158	2006
C	35	427	377

Table 4.13: # Adjustments Categorization BU 3

4.3.3 Planner - Perspective

Planners and their enrichment behaviors are at the core of this master thesis and are thus a very valuable 'asset' that should be investigated further. In this section, we will zoom in on the actual contribution each individual planner has toward the FVA. Given the results in the previous section, one could ask themselves if planners actually add value at all.

Within Table 4.14, each planner is noted in descending order based on their FVA. Within this table, we can identify some interesting points of notice. Firstly, out of a total of 23 planners, only 7 managed to add value on average. However, even those planners that managed to achieve a positive FVA, only improved the FVA by a small amount.

Furthermore, these planners only operated within BU 1 or 3. The best planner in BU 2 reduced the FA by 0.199. The number of enrichments a planner has made does not seem to be an accurate indicator of their FVA since we can see planners with both high and low numbers of enrichments operate on both ends of the spectrum. It does seem that planners that are being given an inaccurate statistical forecast are more likely to reduce FVA.

User	BU	Number of Enrichments	FA Statistic	FA Enrichment	FVA
13	1	259	0.25	0.35	0.10
15	1	57	0.54	0.61	0.06
7	1	990	0.37	0.43	0.06
27	3	316	0.29	0.35	0.06
26	3	633	0.37	0.39	0.02
25	3	3,692	0.50	0.51	0.01
10	1	362	0.37	0.37	0.00
24	3	16,394	0.52	0.52	-0.00
22	1	60	0.72	0.65	-0.07
6	1	594	0.44	0.33	-0.10
1	1	1,056	0.46	0.28	-0.18
4	2	1,115	0.09	-0.11	-0.20
11	2	333	-0.70	-1.11	-0.41
14	1	174	-2.92	-3.74	-0.81
19	2	59	0.14	-0.75	-0.89
2	2	1,744	-0.02	-1.06	-1.05
12	1	442	-0.31	-1.51	-1.21
18	2	53	0.26	-1.16	-1.42
3	2	1,381	-0.04	-1.47	-1.44
9	2	1,046	-0.66	-2.35	-1.69
5	2	668	-0.82	-2.83	-2.01
16	2	113	-0.05	-2.73	-2.68
8	2	852	-2.71	-8.24	-5.53

Table 4.14: Planner Performance, ranked by FVA

Looking at the enrichment descriptives (Table 4.15) for each planner, we can see that in most situations, planners do not improve the FVA, as indicated by the column '% Corrected Adjusted'. The information from this column is quite worrying given that most are far below 50% and in some cases even drop down into single digits. In addition, by looking at the difference between the mean and median values, we can get more information about their behavior. In almost all situations, the mean value seems to deviate more from 0 than the median. This indicates that in most cases, planners only add or subtract a smaller amount of value, while there are a few adjustments with a very large size that influence the mean heavily. When comparing the positive and the negative adjustments to each other, the median negative value deviates further from zero compared to the median positive value for each planner. For almost all planners, the worst negative adjustment is also much larger than their best positive adjustment. Given the high values here, it does seem that forecasters' best enrichments are unable to compensate for their worst ones.

Planner	% Correct Adjusted	Mean Positive	Median Positive	Best Positive	Mean Negative	Median Negative	Worst Negative
1	35%	0.47	0.22	18.21	-1.93	-0.47	-198.13
2	18%	1.20	0.42	33.11	-12.60	-3.88	-297.00
3	20%	4.02	0.37	211.25	-19.83	-4.39	-680.54
4	7%	0.55	0.35	3.80	-17.73	-2.03	-642.69
5	25%	8.62	0.33	243.26	-28.96	-5.87	-1,346.61
6	7%	0.45	0.24	6.88	-0.59	-0.35	-1.68
7	23%	0.34	0.18	7.68	-2.16	-0.51	-39.63
8	16%	30.41	0.49	1,637.80	-137.34	-15.60	-3,754.51
9	28%	2.68	0.26	439.68	-36.05	-3.84	-1,153.95
10	5%	8.67	0.26	47.53	-1.32	-0.72	-7.76
11	24%	29.12	0.41	817.93	-20.72	-2.91	-876.92
12	22%	1.84	0.33	43.12	-20.07	-2.84	-783.75
13	67%	0.38	0.32	1.99	-2.31	-0.66	-31.30
14	27%	2.74	1.01	20.69	-3.62	-2.13	-86.42
15	47%	0.38	0.30	1.10	-1.33	-0.94	-3.80
16	37%	0.84	0.43	9.50	-96.26	-11.67	-984.50
18	36%	1.97	0.53	27.65	-12.58	-0.58	-287.95
19	25%	19.00	0.43	274.71	-52.00	-5.40	-614.26
22	37%	0.12	0.00	0.66	-0.32	-0.01	-1.54
24	51%	0.41	0.18	81.73	-2.56	-0.31	-1,816.18
25	45%	1.97	0.20	1,187.15	-2.12	-0.28	-272.48
26	61%	0.43	0.27	29.60	-2.72	-0.37	-111.40
27	54%	0.38	0.29	1.87	-4.80	-0.35	-180.16

Table 4.15: In-depth Statistics FVA Descriptives per Planner

4.4 Correlations

As a part of the exploratory data analysis, we will also investigate the correlations of the variables. Correlations can tell us a lot about the feature space of the dataset. We will investigate both the correlations between the independent variables and the correlations between the independent variables and the FVA.

When independent variables are correlated strongly with each other, several variables predict the same phenomenon and this can lead to biased predictions. Effectively, you will have a single phenomenon that is double or triple-counted. However, depending on the business context, it is sometimes inevitable. An example of this would be taking into account both the statistical forecast and the size of the enrichment. Most likely, if there is a large statistical forecast, there could also be a larger adjustment. However, these are two separate pieces of information. On the other hand, in Table 4.16, you will find that higher correlations are actually preferred. Higher correlations indicate that there is some relation between the independent variable and the target, and this would be a good sign for the quality of the prediction model. However, one must always remember that correlation is not the same as causation. If there is a high correlation, this might not be caused by a relationship between the variables, but it could be caused by an external factor that affects both variables.

Looking closer at the correlations between the features and the FVA in Table 4.16, there do not seem to be very large variables. However, the largest ones are 'Optimism Bias', 'Enrichment Size' and 'Number of Adjustments' & 'Overreaction Bias'. Negative

correlations indicate that when the value of, for example, Optimism Bias is high, the FVA is low, and thus that the forecast is likely to be less accurate.

Feature	**Correlation**
Optimism Bias	-0.26
Enrichment Size	-0.19
Number of Adjustments	-0.17
Overreaction Bias	-0.17
Previous Forecast	-0.13
Hour of the Day (cos)	-0.06
Hour of the Day (sin)	-0.03
Timelag	-0.03
Day of the Month (sin)	-0.02
Statistical Forecast	-0.02
Day of the Week (sin)	-0.01
Day of the Week (cos)	-0.01
Anchoring Bias	0.03
Day of the Month (cos)	0.06
Previous FVA	0.15
Hierarchy Level	0.15

Table 4.16: Correlations between Features and FVA

Correlations between the different independent variables are also calculated. Table 4.17 has been created to show the most strongly related independent variables. Within this table, correlations above 0.5 have been marked in bold. For reference, a table with all correlations is present in Appendix E.1. There are a few that seem to be strongly related. Any features that are directly related to the forecasted quantity; i.e., statistical forecast, previous forecast, and enrichment size have higher correlations. This is especially true for the relations regarding the previously forecasted quantity. Between the statistical forecast and the enrichment size, the correlation is much lower.

In addition, there are strong correlations between the number of adjustments, hierarchy levels, and the various planner-related biases. An interesting statistic here is the correlation between the hierarchy level and the overreaction bias. It states that when the hierarchy level is low, the overreaction bias is expected to be high. This makes sense, as when one enriches forecasts on a higher hierarchy level, their enrichment encompasses a larger number of products and customers and it is expected that they are less precise. Combined with the tendency to over-forecast, the results are not surprising. Furthermore, there is a large negative correlation between the hierarchy level and the number of adjustments. Indeed, if a planner enriches a forecast for a single customer, this enrichment is generally less important than when a planner enriches a forecast for a large number of customers in one go. Thus, for a higher hierarchy level, it is logical that planners spend more time adjusting the forecast, or in other words, adjusting the forecast several times before coming to a final enriched forecast.

	Statistical Forecast	Previous Forecast	Hierarchy Level	Number of Adjustments	Optimism Bias	Anchoring Bias	Overreaction Bias	Enrichment Size
Statistical Forecast	1.00							
Previous forecast	**0.78**	1.00						
Hierarchy Level	-0.08	-0.14	1.00					
Number of Adjustments	0.24	0.25	**-0.65**	1.00				
Optimism Bias	0.22	0.26	**-0.60**	**0.57**	1.00			
Anchoring Bias	0.06	not sig	0.40	-0.21	-0.10	1.00		
Overreaction Bias	0.12	0.17	**-0.73**	**0.58**	**0.68**	0.13	1.00	
Enrichment Size	0.33	**0.62**	-0.15	0.19	0.22	-0.03	0.16	1.00

Table 4.17: Correlation Table (Subset)

5 Methodology

In this section of the thesis, we will explain what methods have been used in order to create the models. Analysis has been conducted within Dataiku. Within Dataiku one will build a 'flow' with several recipes that transform the data to the desired output. Many recipes are standardized functions, however, for this thesis, Jupyter notebooks have been added as recipes in many cases.

5.1 Variables

5.1.1 Dependent Variable

As a reminder, within this thesis, the goal is to accurately predict the accuracy of forecast enrichments based on a number of features. Given certain feature importances, one would be able to predict which enrichments might improve or harm the forecasting accuracy. Thus, we will create a prediction model that will predict the FVA of a specific enrichment. The FVA is calculated by subtracting the FA_s (Statistical Forecasting Accuracy) from the FA_a (Actual Forecasting Accuracy). The FAs can be calculated using Formula 4.1 and the FVA by Formula 4.2.

As explained before, this variable is corrected for outliers. However, values can still range from -3,044.96 to 1,637.80. This large spectrum makes it extremely hard to predict accurately. Prediction models will become very erratic and volatile while trying to fit themselves into the training dataset and it is also not necessary to be able to identify good or bad enrichments. Therefore, the FVA has been capped at a maximum of 5 and a minimum of -5. This might seem like a narrow range, but there are only a very small number of enrichments that have such dramatic high values and these do not add too much information at the end of the project. For BU 1 and 3, 97.52%, and 97.40% of values will fall within the capped range. Only BU 2's volatility will be significantly affected since only 72.62% falls within this range. Values that fall outside the range will thus be capped and not deleted, since these can still inform the machine-learning model about the characteristics of extreme enrichments.

5.1.2 Independent Variables

To accurately predict the FVA, one needs a number of independent variables on which the predictions will be based. There are 14 base features that can be assigned to four different groups: enrichment, time, product, or planner.

- **Forecast-Related**
  These concern factors regarding the size of the original forecast, the size of the adjustment, and the direction of the adjustment. One interesting variable is the number of adjustments. Within the dataset, one can identify when a forecast for a specific product has been adjusted multiple times. A large number of adjustments could indicate that the forecast is heavily investigated and could be a predictor of higher accuracy.

- **Time-Related**
  The second group of features is related to the time at which an enrichment is made. As seen in the paper of Broeke et al. (2019), planners can become more volatile in their adjustments when nearing the moment of sale. Furthermore, it could be interesting to see if planners are affected by the time of day or the day of the month at which they make enrichments. Being pressured by a deadline, planners might act with less care and this could affect the FVA.

- **Product-Related**
  The number of features in the third group is rather limited. These features concern a specific product and are categorical in nature. In this dataset, we incorporate the business unit in which the product is sold and its category based on the theory of Scholz-Reiter et al. (2012).

- **Planner-Related**
  The last group of features is related to the planners and their biases.

In total, 3 biases are calculated: Optimism Bias (B_o^i), Anchoring Bias (B_a^i), and Overreaction Bias (B_r^i). The calculations follow the formulas defined by Eroglu & Croxton (2010). The formulas for bias are based on the percentage error of the statistical forecast $p_{t,stat}^i$ (Equation 5.1) and the percentage error of the enriched forecast $p_{t,enr}^i$ (Equation 5.2). As expected by the name, these formulas check the error per forecast and divide them by the sales quantity. Positive values indicate an over-forecast and negative values indicate an under-forecast.

$$\text{Percentage error of statistical forecast } p_{t,stat}^i = 100\% * \left(\frac{F_{t,stat}^i - A_t^i}{A_t^i} \right) \tag{5.1}$$

$$\text{Percentage error of enriched forecast } p_{t,enr}^i = 100\% * \left(\frac{F_{t,enr}^i - A_t^i}{A_t^i} \right) \tag{5.2}$$

To identify optimism bias, Equation 5.3 is used. One only has to look at the percentage error of the enriched forecasts and the number of enrichments made (n). If a planner

shows a tendency to enrich optimistically, the value will be larger than 0, while it will be lower than 0 if the planner is frugal with their adjustments. This value is thus tied to an individual planner. To give an example, if a planner forecasts 150 units and the actual demand is 100, $p^i_{t,enr}$ is 50%. In turn, optimism bias B^i_o will also be 50%. When there are more enrichments, one takes the average of all percentage errors. In order to test if the bias is significant, a statistical test has to be identified. When one wants to identify if a sample deviates from a mean of 0, and is dealing with a continuous variable, a studentized t-test can be executed with a significance level of .05. This test is further explained in Appendix C.4.

$$B^i_o = \frac{1}{n}\sum_{i=1}^{n} p^i_{enr} \tag{5.3}$$

The second bias is the anchoring bias (Equation 5.4), which indicates the tendency of a planner to anchor their enrichments on the statistical forecast. For this measure, two binary variables x^i and y^i must be calculated. Variable x^i indicates if the planner undershoots in their forecast. The undershoot is represented by the fact that the absolute error of the enriched forecast is lower than the error of the statistical forecast, but multiplying the errors should indicate a value that is larger than 0. This happens in situations where either both errors are positive or both are negative. If the planner overshoots the error would be below 0 since a positive error is multiplied by a negative error. y^i only checks if the absolute forecast error of the final forecast is smaller than that of the statistical one. To give an example, imagine that p^i_{enr} is +30% and p^i_{stat} is -50%. y^i will be reported as 1 since the absolute error is reduced, but x^i will report as 0 since the product of the error terms will be negative. This metric is centered in 0.5, meaning that in half of the cases in which they improve the accuracy they overshoot, and in the other half they will undershoot. If the metric reports below 0.5, they overshoot in more instances than they undershoot.

In order to test this statistic, we use a binomial test that is centered around 0.5. This test is chosen since we are dealing with a metric based on two binary variables. The significance level is also 0.05 (Appendix C.1).

$$B^i_a = \frac{\sum_{i=1}^{n} x^i}{\sum_{i=1}^{n} y^i} \tag{5.4}$$

$$x^i = \begin{cases} 1 \text{ if } \left|p^i_{enr}\right| < \left|p^i_{stat}\right| \text{ and } p^i_{enr}p^i_{stat} > 0 \\ 0 \text{ otherwise} \end{cases} \tag{5.5}$$

$$y^i = \begin{cases} 1 \text{ if } \left|p^i_{enr}\right| < \left|p^i_{stat}\right| \\ 0 \text{ otherwise} \end{cases} \tag{5.6}$$

The last bias is the overreaction bias. Overreaction bias shows if a planner is prone to adjust forecasts erratically, which leads to a larger forecasting error in the opposite

direction. If the statistical forecast error would be 20% below the actual sales quantity, then overreaction bias would show if the planner enriches the forecast by such an amount that the final forecast error is more than 20% over the sales quantity. In order to check this, an additional binary variable z^i has to be calculated. An overreaction is identified when the absolute final forecast error is larger than the absolute statistical forecast error and their multiplication is below zero. As seen before, the multiplication is below zero when one forecasting method undershoots while the other overshoots. Overreaction bias can range from 0 to 1, and its significance is tested using a Wilcoxon Signed Rank test (Appendix C.5) since we are dealing with a continuous outcome and one categorical predictor variable.

$$B_r^i = \frac{1}{n} \sum_{i=1}^{n} z^i \tag{5.7}$$

$$z^i = \begin{cases} 1 \text{ if } \left|p_{enr}^i\right| > \left|p_{stat}^i\right| \text{ and } p_{enr}^i p_{stat}^i < 0 \\ 0 \text{ otherwise} \end{cases} \tag{5.8}$$

Variable	Type	Description
Forecast-Related:		
Enrichment Size	Numerical	Size of the enrichment (either negative or positive)
Statistical Forecast	Numerical	Number of products statistically forecasted by EyeOn
Previous Forecast	Numerical	Forecast of last month
Number of Adjustments	Numerical	Number of times a single product-customer combination is adjusted
Time-Related:		
Timelag	Numerical	Time in months from enrichment to forecasted period
Hour of the Day	Numerical	Hour of the day at which enrichment is created
Day of the Week	Numerical	Day of the week at which the enrichment is created
Day of the Month	Numerical	Day of the month at which the enrichment is created
Previous FVA	Numerical	Represents the last reported FVA for this product-customer combination at the time of adjustment
Product-Related:		
Business Unit	Categorical	Business Unit: 1,2,3
Product Category	Categorical	ABC-XYZ product designation
Planner-Related:		
Optimism Bias	Numerical	Numerical representation of the optimism bias of a planner
Anchoring Bias	Numerical	Numerical representation of the anchoring bias of a planner
Overreaction Bias	Numerical	Numerical representation of the overreaction bias of a planner

Table 5.1: Independent Variables Overview

5.2 Scaling & Encoding

Within our models, we will be dealing with both numerical data as well as categorical data. The categorical data will be one-hot encoded in order for the models to properly incorporate this data. This concerns the columns regarding the Business Unit and the Product Category. We have opted for the one-hot encoding compared to the numerical encoding for its ease of use. This method is namely supported by the LightGBM model, Neural Network, and the Automatic Feature Engineering package 'AutoFeat'. Furthermore, time-related features are cyclically encoded. Certain features like the hours of a day or days in a week are cyclical, e.g., Monday is the day after Sunday, or the first day comes after the last day. Humans can understand this through context, but machine learning models only understand this by this encoding. One can transform these features by dividing them and transforming them to a sine and a cosine value. This encoding can improve the accuracy of a ML model (Bescond, 2020).

The features are also scaled using a Standard Scaler (Pedregosa et al., 2011). The features encompass different phenomena and different ranges of values are used. Features with a large spectrum of potential values could mislead the model into thinking that they are more important than the other features. Therefore, scaling transforms all features to roughly the same range of values (Brownlee, 2020). The transformation is shown in Equation 5.9.

$$x_{scaled} = \frac{x - \mu}{\sigma} \tag{5.9}$$

5.3 Models

In total, three types of models have been created: Linear Regression, LightGBM, and Neural Networks. All models are regression models which predict the expected FVA. Each model has their respective strengths and weaknesses which will be explored in the section below.

5.3.1 Linear Regression

As a baseline, we will try to predict the FVA through the use of Multiple Linear Regression. This baseline utilizes the principles of simple linear regression while incorporating multiple independent input variables predicting one single dependent output variable. Linear regression is a modeling technique that assumes a linear relationship between an independent input variable and a dependent output/target variable. The assumption of linearity makes such a model very simple to create and interpret. Furthermore, it only requires limited computational resources and is robust against overfitting, especially for small datasets. Within this dataset, we have more than one input variable, which means that Multiple Linear Regression should be executed. The premise is the exact same, however, n input features are used which leads to a projection to a $n+1$ dimensional space. Such an extension of the original model can make it more complex to interpret, but it is still built of simple mathematical principles. From this baseline, we can also investigate what additional value machine learning models would add. The Multiple Linear Regression model will be created through the Scikit-learn package in Python (Pedregosa et al., 2011).

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \epsilon \tag{5.10}$$

Equation 5.10 shows the mathematical representation of multiple linear regression. Y is the predicted variable, β_0 is a constant and the remaining βs are multipliers for the

values of each independent feature X_i. The ϵ at the end represents the error term that encapsulates various factors that are not modeled into the model.

5.3.2 LightGBM

As explained before, a LightGBM model is a gradient-boosting decision tree that is built iteratively (Ke et al., 2017). In the model, the data is first split into a training and a testing set. The split is randomly sampled within the data and 20% of the data is utilized to verify and measure the model performance, stratified per the sizes of the different BUs. Then we can create the LightGBM model. We do not have to specify the categorical variables since these are one-hot encoded. Next, the data is scaled. Even though it should not be too important for a tree-based model, it is still recommended to scale the numerical values. The data is scaled through the StandardScaler within the Sci-kit learn package in Python.

Hyperparameter	**Type**	**Range**
'boosting_type'	Fixed	'goss'
'metric'	Fixed	'mae'
'max_depth'	Integer	5, 7, 9, 11
'num_leaves'	Integer	32, 128, 512, 2048
'num_iterations'	Integer	100, 200, 300, 400, 500
'learning_rate'	Float	0.05, 0.10, 0.15, 0.20, 0.25
'colsample_bytree'	Float	0.50, 0.75, 1.00
'min_child_samples'	Integer	25, 50, 75
'reg_lambda'	Integer	25, 50, 75, 100

Table 5.2: Hyperparameter Tuning LightGBM

One of the most crucial steps in any machine learning model is hyperparameter tuning. The performance of the model can vary greatly based on the values of the hyperparameters. Identifying what the best hyperparameters are, is a computationally intensive task. Especially since in this thesis, we will be using a grid search. In a grid search, a machine learning model is built for each combination of hyperparameter values (i.e. 14400), and the best option is chosen based on the testing MAE. In order to get good results, balanced with a good runtime, a two-step process of hyperparameter tuning is proposed. Firstly, a wide grid search is executed based on the parameter values in Table 5.2. This provides an initial estimate from which will be searched further. In the second step, a narrower grid search is defined, a so-called local search. To identify the search space of the local grid, one has to extract the hyperparameters. Then, one calculates the averages for a higher range and a lower range. To give an example: the global grid search states that the optimal number of iterations is 400. Then the local grid search would investigate values 350, 400, and 450. When the global grid search identifies parameters that are either a maximum or a minimum of a hyperparameter value, you only check the value below. For instance, the global search finds the value of lambda to be 100, in the local grid search only 87 and 100 are tested.

5.3.3 Neural Network

The neural network operates very differently compared to the LightGBM model. Keras has been utilized as the package at the base of the network (Chollet et al., 2015). The type of neural network is a feedforward neural network. Within the model, one first converts all

the categorical variables to one-hot encoded equivalents. Then, it uses the same scaler and train-test splits as before. Just like the LightGBM model, a grid search is employed to test all the various options for the hyperparameters. These hyperparameters are also shown in Table 5.3. The neural network has a higher runtime compared to the LightGBM model, and thus the hyperparameter optimization is limited by reducing the number of parameter values.

The neural network follows the following best practices (Ranjan, 2019):

- 2 hidden layers, with a dropout percentage and activation function 'relu' (An extra layer is tested as well).
- 1 singular output layer, with the 'linear' activation function, of size 1
- Number of nodes per layer as a power of 2, converging to a lower number further down the network.

The hyperparameter search will however test if an additional third hidden layer would increase the accuracy of the model. The 'num_neurons' parameter only applies to the first hidden layer. The input layer is, of course, equal to the number of features provided to the model. The next layer is thus equal to the selected value for the hyperparameter. The following layers all divide the number of neurons by 2, making a converging network. The last layer is indeed a singular, linear output layer.

Hyperparameter	**Type**	**Range**
'learning_rate'	Float	0.001, 0.05, 0.010
'num_epoch'	Integer	50, 100, 200
'batch_size'	Integer	8, 32, 64
'num_neurons'	Integer	32, 64, 128
'num_hidden_layers'	Integer	2,3
'kernel_regularizer'	Categorical	L2(0.01)
'dropout'	Float	0.2
'activation'	Categorical	'relu'

Table 5.3: Gridsearch Neural Network

5.3.4 Feature Engineering

Machine Learning models can be run using the basic set of features. However, in order to improve accuracy, one could explore changing or adding features that better predict the dependent variable. Adding features might be done through the use of domain knowledge, or finding additional data sources from which features can be generated. An example could be to add weather information when predicting the number of ice cream sales. Changing features concerns the exploration of mathematical transformations that can be done on the current features. Examples of this could be to take the log, square (root), or a ratio of the current features. Then, one can test if this adjusted feature predicts the dependent variable better.

Feature engineering can be done in several steps, with each succeeding step increasing the number of features and interaction effects. In the first step, only the mathematical transformations are executed. In the second step, interaction effects between different features are also incorporated. As an example, it could create a feature called '(Number of

adjustment)**2 × (log(1/anchoring bias))'. As one can imagine, comparing all features in this manner, the number of features will grow exponentially. For the adjustment of features, the open-source Python package 'AutoFeat' is utilized (Horn et al., 2019). Since manual feature transformations can be very labor intensive, this package is designed to remove this burden from the researcher. AutoFeat can automatically generate new features, and select features that it deems useful for the model accuracy. One can identify the number of steps that they deem necessary. Within the results, feature engineering is also explored in the varying machine learning models.

6 Results

Within this section, the methodology will be executed and its results will be reported. We will check if biases are present among the planners, what the characteristics of the dataset are, and how the various prediction models function together with their validity.

6.1 Bias

In Section 5.1.2, we described the ways in which planners can be biased. Results from the literature indicate that planners are often biased and we will apply the formulas by Eroglu & Croxton (2010) to verify the biases among the planners in the dataset. For each planner, the biases that they exhibit in their enrichment behavior are noted in Table 6.1.

User	BU	Number of Adjustments	Optimism	Anchoring	Overreaction
1	1	1,056	2.25	0.33	0.18
2	2	1,744	12.58	0.19	0.31
3	2	1,381	22.44	0.18	0.30
4	2	1,115	3.34	0.03	0.04
5	2	668	38.27	0.42	0.38
6	1	594	0.12	0.02	not significant
7	1	990	0.73	0.04	0.03
8	2	852	133.80	0.20	0.39
9	2	1,046	41.35	0.24	0.23
10	1	362	0.97	0.05	0.02
11	2	333	19.45	0.28	0.23
12	2	442	23.76	0.16	0.35
13	1	259	0.76	0.12	0.07
14	1	174	8.46	0.34	0.24
15	1	57	0.44	0.11	0.11
16	2	113	72.13	0.28	0.35
18	2	53	12.35	not significant	0.26
19	2	59	52.61	not significant	0.36
22	1	60	0.52	0.28	not significant
24	3	16,394	2.78	0.29	0.12
25	3	3,692	4.10	0.27	0.10
26	3	633	1.39	0.18	0.07
27	3	316	not significant	0.25	0.10

Table 6.1: Bias Table per Planner

It is clear that optimism bias is very prevalent among planners and that their values are very high. Accordingly, the spectrum ranges from 0.12 up to 133.80. Every single planner (with the exception of planner 27) has a significant optimism bias. On average, they all forecast more than the sales quantity. The values for planners in BU 2 are extremely high. For anchoring bias, almost all forecasters show a significant bias. This metric is centered at 0.5, and values below 0.5 indicate that when a planner reduces the forecasting error through their enrichment, they are more inclined to overshoot. Just like with optimism bias, no planners (with a significant bias value) show clear anchoring on the statistical forecast. Lastly, the overreaction bias is also prevalent across this set of planners.

From Table 6.1, it seems that there could be a relation between optimism and overreaction bias. Therefore, the correlations are calculated between the biases. Table 6.2 shows that there indeed is a strong correlation between optimism and overreaction biases. From this correlation, we can deduce that planners who have a high optimism bias, are also likely to overreact. Figure 6.1 shows also that planners from BU 2 are especially prevalent to be exposed to these biases. Anchoring bias also has significant correlations with the other two, but these values are much smaller.

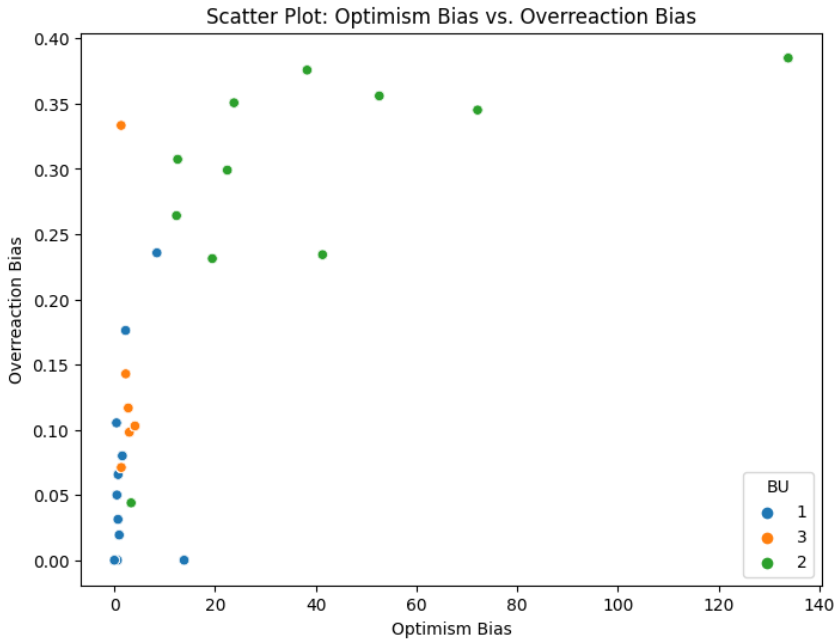


Figure 6.1: Optimism Bias versus Overreaction Bias

	Optimism	**Anchoring**	**Overreaction**
Optimism	1.00		
Anchoring	-0.10	1.00	
Overreaction	0.68	0.13	1.00

Table 6.2: Bias Correlations

6.2 Prediction Models

Within this section, we will investigate the models that were proposed in Section 5.3. In particular, we will evaluate their performance, both in terms of accuracy and speed, their validity, and their behavior. The results of the most important models can be found in Table 6.5. For conciseness, this table only represents a subset of all models that have been created and the full table can be found in Appendix G.

6.2.1 Linear Regression

The first prediction method will be linear regression. It is a simple method that can be utilized as a baseline to compare the machine learning models. We find that a large number of assumptions are violated in Appendix F (Linearity, Normality, Homoscedasticity). Since these assumptions are not required for a prediction model, we will still utilize it as a baseline. However, it does show us that linear regression might not be a good fit for the data. In Appendix G, a table is shown with the results and information of all models. Given the low performance for both training and testing sets, combined with the violated assumptions, we decide to not utilize linear regression for the final model.

6.2.2 LightGBM

For the second prediction model, a LightGBM algorithm is used. Compared to the linear regression model, there are no clear assumptions, but there are a few things to keep in mind. To create a good LightGBM model, its hyperparameters have to be tuned correctly. This ensures convergence of the model and in turn, good results. Additionally, one has to keep a close eye on the behavior of the model. Certain configurations of hyperparameters can cause the model to either under- or overfit which reduces the overall quality. In theory, the data does not have to be scaled in order to produce correct results, but models are created both with and without scaling to check this. Lastly, in the best models, feature engineering will be explored to further improve the prediction quality.

The basic models, without scaling and clipping, seem to easily surpass the quality of their respective linear counterparts. Scaling does indeed not seem to improve the accuracy of the model. However, clipping does improve the quality of the model. Just like with the linear regression model, the model operates better when only using data from BUs 1 & 3. For the clipped and scaled model, feature engineering is also implemented. By using one-step feature engineering (i.e., only mathematical transformations of the features), a handful of new features are created. These new features seem to have a significant effect on model quality by reducing the MAE of the test set from 0.507 to 0.435 (BUs 1 & 3) and from 0.669 to 0.594 (all BUs). However, the second step of feature engineering does not seem to add any additional value. For some subsets of BUs, the testing MAE is improved slightly through this second step, but for others, it slightly deteriorates the performance. Due to feature engineering, it seems that models are searching for patterns in the noise of the data (i.e. overfitting) since their training MAE is quite a bit lower compared to the testing MAE.

Without feature engineering, the models are largely unbiased. Through feature engineering, it seems that the models are more inclined to predict pessimistically. In other words, it seems that the models predict a more negative FVA on average, especially when

focusing purely on BUs 1 & 3.

Compared to the linear regression model, the runtime is significantly longer. This is mainly due to the large number of models it has to train to optimize the hyperparameters. However, increasing the steps in the feature engineering also increases the runtime, since the dataset is larger through the additional features. The chosen hyperparameter configuration leads the model to converge on both the training and testing sets. There is not a very strong 'elbow' visible in the training set. That seems to improve iteratively within the predefined parameter set. The testing set does show a clear 'elbow' and remains roughly equal while the training set reduces error. This means that the model has a tendency to overfit. A visualization of the convergence is shown in Figure 6.2.

Hyperparameter	**Parameter Value**
'boosting_type'	'goss'
'metric'	'mae'
'max_depth'	10
'num_leaves'	512
'num_iterations'	500
'learning_rate'	0.125
'colsample_bytree'	0.63
'min_child_samples'	13
'reg_lambda'	87

Table 6.3: Hyperparameter Tuning LightGBM

For the LightGBM model, the recommendation for this dataset would be to only use a single step of feature engineering and only focus on BUs 1 and 3. Single-step feature engineering has a lower runtime and similar accuracy compared to a 2-step feature engineering configuration. Furthermore, using fewer features makes it easier to explain the results to a 'layperson'. BU 2 is also very dissimilar to BU 1 and 3. This reduces the accuracy of the model significantly. Combined with the information we have about BU 2 (Section 4.3.1), we will only focus on BU 1 and 3 in the remainder of this thesis. This leads to the hyperparameter configuration shown in Table 6.3. On the testing data, it has the ability to predict the FVA with an MAE of 0.435. However, the model forecasts pessimistically so in most cases the LightGBM model will predict a FVA that is less than the actual FVA.

6.2.3 Neural Network

The last explored prediction method is the feed-forward neural network. Within Section 5.3.3, the design of the model and the hyperparameter tuning are described. Just like the LightGBM model, certain combinations of the NNs are tested and the results can be seen in both Table 6.5 (Subset) and Appendix G.

Unlike the LightGBM model, the NN does seem to benefit from the 2-step feature engineering. The testing MAE remains roughly equal between no (0.495) and one-step (0.497) feature engineering but manages to go down a bit further after the second step (0.486). The NN seems to predict more pessimistically compared to the LightGBM model and in turn, also more pessimistic than the actual results. Additionally, the highest testing

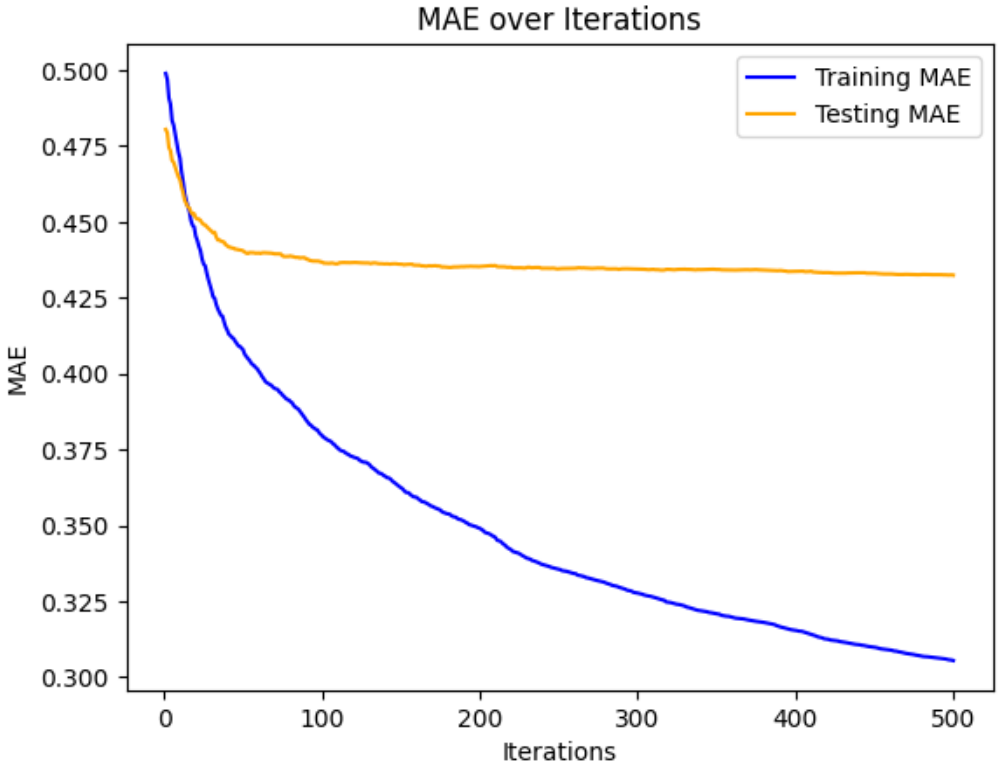


Figure 6.2: LightGBM Convergence

MAE is also slightly higher (0.486 for NN compared to 0.435 for LightGBM). The most striking difference is the increase in the time required for hyperparameter tuning, which is roughly double. One also has to take into account the significantly smaller grid space in which the hyperparameter tuning is executed.

After hyperparameter tuning, Table 6.4 shows the best parameters for the model together with two-step feature engineering. With this parameter configuration, the model converges quickly as visualized in Figure 6.3.

Hyperparameter	**Parameter Value**
'activation'	'relu'
'batch_size'	64
'dropout_rate'	0.2
'epochs'	200
'kernel_regularizer'	L2(0.01)
'learning_rate'	0.001
'num_hidden_layers'	3
'num_neurons'	128

Table 6.4: Hyperparameter Tuning Neural Network

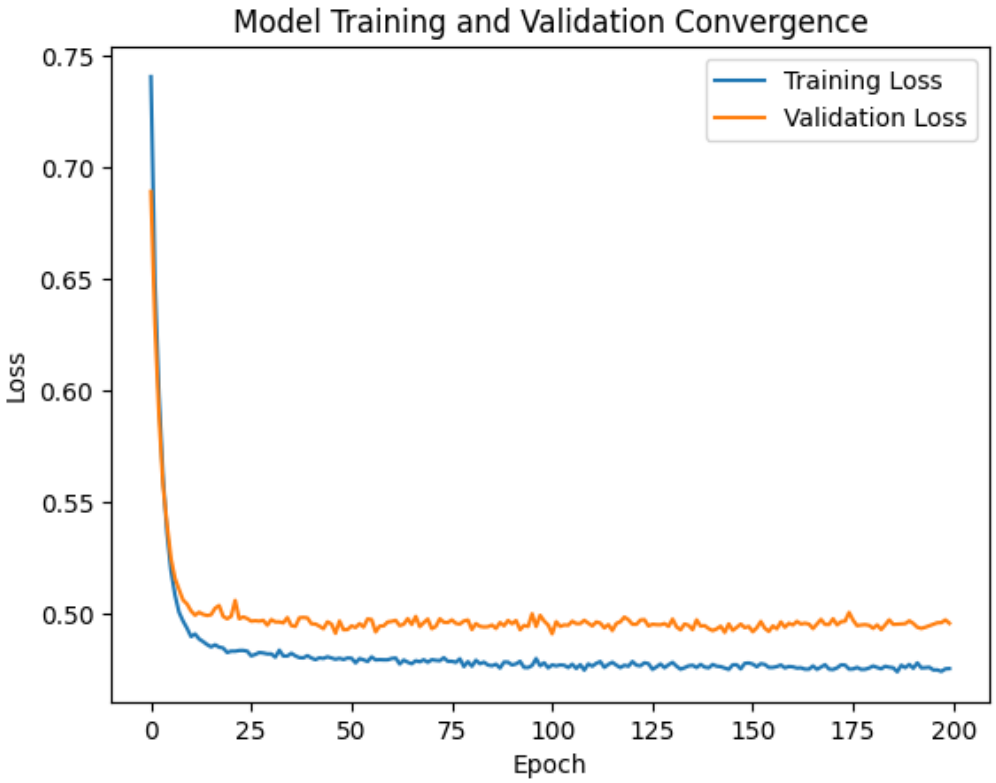


Figure 6.3: NN Convergence

#	Model	BU	Scaled	Clip	FE	# Feat	Training MAE	Testing MAE	Training Bias	Testing Bias	Runtime (hh:mm:ss)
7	LR	All	Yes	Yes	No	29	0.801	0.812	-0.000	0.003	00:00:00
8	LR	1 & 3	Yes	Yes	No	28	0.556	0.553	0.000	0.083	00:00:00
16	LGBM	All	Yes	Yes	No	29	0.494	0.669	-0.001	-0.024	03:09:10
17	LGBM	1 & 3	Yes	Yes	No	28	0.489	0.507	0.018	-0.033	02:07:26
19	LGBM	All	Yes	Yes	1	35	0.491	0.594	-0.172	-0.201	02:56:44
20	LGBM	1 & 3	Yes	Yes	1	33	0.334	0.435	-0.695	-0.697	04:02:14
22	LGBM	All	Yes	Yes	2	104	0.436	0.592	-0.166	-0.188	06:03:11
23	LGBM	1 & 3	Yes	Yes	2	102	0.324	0.437	-0.670	-0.694	04:52:25
31	NN	All	Yes	Yes	No	28	0.627	0.649	-0.272	-0.294	14:16:02
32	NN	1 & 3	Yes	Yes	No	27	0.474	0.495	-0.945	-0.953	10:42:31
34	NN	All	Yes	Yes	1	37	0.622	0.649	-0.376	-0.390	13:56:15
35	NN	1 & 3	Yes	Yes	1	32	0.472	0.497	-0.936	-0.930	10:45:24
37	NN	All	Yes	Yes	2	112	0.599	0.652	-0.283	-0.290	14:32:10
38	NN	1 & 3	Yes	Yes	2	97	0.456	0.486	-0.767	-0.759	10:52:21

Table 6.5: Results Prediction Models (Subset)

6.2.4 Prediction Behavior

Next to just a numerical measure to check the accuracy of a prediction model, we will dive deeper into this section by visualization and closer analysis of the predicted values. This will be done for both the one-step feature engineering LightGBM model and the two-step feature engineering Neural Network. Both predictions are based on BU 1 and 3.

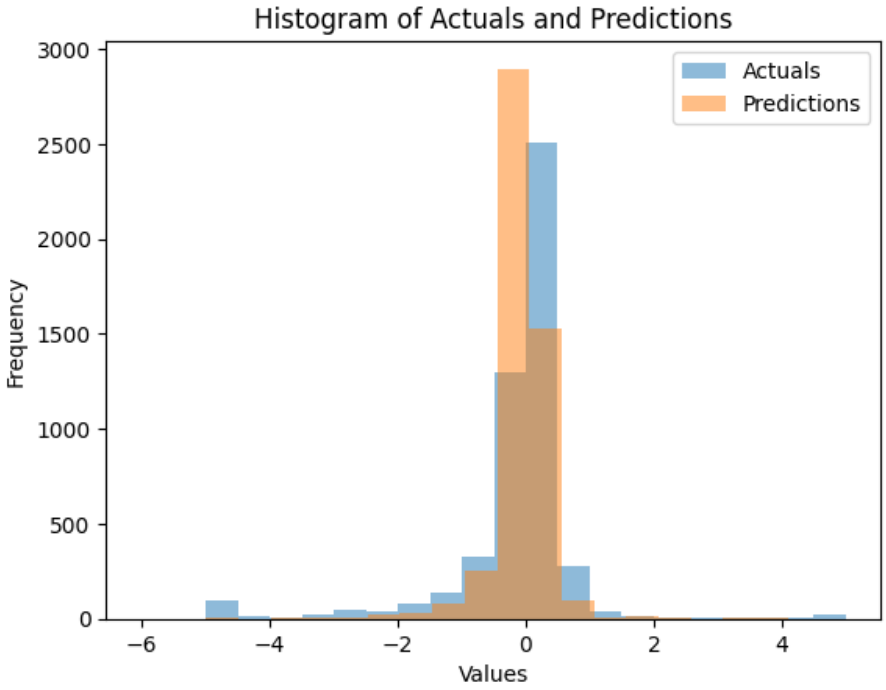


Figure 6.4: Histogram Predictions vs Actuals For FVA - LightGBM

We will begin with a visualization in which the predicted and actual FVAs are compared. The y-axis represents the frequency with which the values occur and the x-axis represents the FVA. Figure 6.4 shows the histogram for the LightGBM model while Figure 6.5 shows the Neural Network. Both models seem to show similar prediction behavior, being slightly pessimistic and cautious. The models do not like to predict extreme values and predict most often a slightly negative FVA. Both models also show very small tails, which also indicates that they do not frequently forecast extreme values. Between the LightGBM and the Neural Network, there is also a difference visible. The Neural Network is more cautious and has smaller tails to its histogram.

	Actual	LGB Prediction	LGB Abs Error	LGB Error	NN Prediction	NN Abs Error	NN Error
Mean	-0.175	-0.041	0.436	0.134	-0.024	0.481	-0.171
Std	1.018	0.443	0.811	0.911	0.349	0.923	1.027
Minimum	-5.000	-6.004	0.000	-5.288	-6.050	0.000	-5.519
25%	-0.217	-0.078	0.030	-0.118	-0.000	0.044	-0.208
50%	0.000	0.004	0.147	0.001	-0.000	0.167	0.000
75%	0.169	0.099	0.457	0.190	0.029	0.449	0.145
Maximum	5.000	4.094	5.537	5.537	3.206	5.989	5.989

Table 6.6: Prediction Descriptives

Within Table 6.6, a deeper dive is taken numerically. The pessimistic behavior sug-

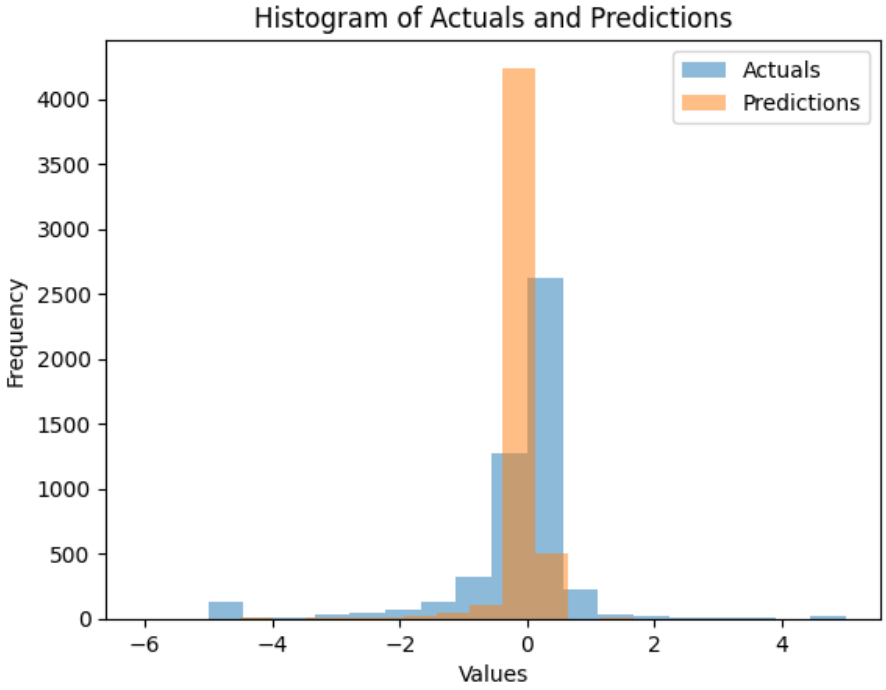


Figure 6.5: Histogram Predictions vs Actuals For FVA - Neural Network

gested in the figures is not actually correct. On average, the actuals have a lower FVA which is mainly caused by a tail of forecast enrichments with a value of -5. Furthermore, the predicted values have a lower standard deviation which shows that it is more 'cautious' in predicting values. To confirm that the Neural Network is more cautious compared to the LightGBM model, its standard deviation is also lower (0.349<0.443). Furthermore, the LightGBM models in the range from 25% up to 75% encompass only enrichments that either improve or decrease the forecasting accuracy slightly, while this is exacerbated for the Neural Network with values even closer to 0. Lastly, the highest prediction of the Neural Network is much lower than that of the LightGBM model (3.206<4.094)

Table 6.7 shows the absolute error of the predicted FVA and how often this occurs. We can see that it often is able to predict the FVA rather accurately, but there are some instances in which the model seems to mispredict error terms completely. For every threshold, the Neural Network has a higher count of occurrences, indicating worse behavior.

Absolute Error	**Count LightGBM**	**Count Neural Network**
>0	4,398	4,953
>1	540	570
>2	236	274
>3	133	181
>4	84	127
>5	31	78

Table 6.7: Prediction Errors per Model

Lastly, we can look at both BU and product categories (Tables 6.8 and 6.9 show the

LightGBM performance and Tables 6.11 and 6.12 show the NN performance) further to identify the models' strong and weak points. The LightGBM model seems to have a higher variance in accuracy across the product categories. For instance, within BU 1, it is able to predict category BX with an average error of 0.068, while category AZ has an average prediction error of 0.665. Especially for BU 1, it is interesting to dive into the average enrichment sizes. Table 6.10 shows that the best-performing categories AX and BX have average enrichment sizes close to zero. Even though it does not hold true for each category, however, it does seem that the prediction errors increase together with the sizes of the enrichments. Given that the enrichment sizes are also all highly positive in the top right of Table 6.10, this is very likely to be the case. The Neural Network is much more consistent ranging from 0.496 up to 0.668. This pattern is also mirrored by the results in BU 1.

BU 3	X	Y	Z
A	0.264	0.433	0.825
B	0.258	0.426	0.553
C	0.468	0.387	0.784

Table 6.8: Prediction Error - BU 3 - LightGBM

BU 1	X	Y	Z
A	0.181	0.429	0.665
B	0.068	0.233	0.455
C	0.280	0.268	0.352

Table 6.9: Prediction Error - BU 1 - LightGBM

BU 1	X	Y	Z
A	1.99	585.97	81.12
B	0.23	64.46	107.27
C	12.26	5.28	8.13

Table 6.10: Average Enrichment Size - BU 1 - LightGBM

BU 3	X	Y	Z
A	0.582	0.668	0.513
B	0.505	0.551	0.522
C	0.496	0.501	0.496

Table 6.11: Prediction Error - BU 3 - Neural Network

BU 1	X	Y	Z
A	0.515	0.534	0.504
B	0.469	0.520	0.539
C	0.475	0.451	0.443

Table 6.12: Prediction Error - BU 1 - Neural Network

Taking all these findings into account, the LightGBM model is selected as the best-performing model. It has a lower MAE, lower hyperparameter optimization time, and only requires a single step of feature engineering. Therefore, in the explainability and notification sections, the results from the LightGBM model are used.

6.3 Explainability

To explain the important features of the machine learning model, a Shapley summary plot is created through the SHAP package (Lundberg et al., 2017). The SHAP package will run a permutation explainer that will identify the 'rationale' behind each prediction. After this, one can explore the results through a number of plots. Results for the entire dataset or singular predictions can be explored. The summary plot shows the overall feature

importances in Figure 6.6.

On the vertical axis, all features are shown in descending order (i.e., the most important feature is at the top). On the horizontal axis, the different instances of a feature are labeled based on their SHAP value, which indicates the impact on model output. Points that are on the right of the divider indicate that it was beneficial to the prediction (i.e., predicted a higher FVA), while points on the left indicate that this instance was negative for model prediction (i.e., lower FVA). Lastly, each point on this horizontal axis is labeled with a color, representing its value for the feature it belongs to, with magenta indicating a high feature value and blue a low feature value. Below the features are itemized and explained per category for clarity.

- Forecast-Related:
  - *Enrichment Size:* According to SHAP, enrichment size is considered to be the most important feature. The colors in the plot show that a small enrichment size (i.e., a large negative adjustment) is beneficial to the expected FVA. The second most important feature is the squared enrichment size, a feature created by AutoFeat. By taking the square, negative adjustments become positive values and thus, it looks purely at the magnitude. There are numerous red dots on both sides of the divider, indicating that large adjustments can both improve or decrease the FVA. However, the original feature also indicates the importance of the direction.
  - *Statistical Forecast:* Amongst a lot of blue dots, we can see a few red ones on the right side of the divider. The small number of red dots indicates that there are a few very large statistical forecasts. However, these are expected to improve the FVA. Due to the scaling, it is hard to see what effect smaller statistical forecasts have on the expected FVA.
  - *Previous Forecast:* Apparently, when the last forecasted quantity was low, it indicates that the FVA is predicted to be higher.
  - *Number of Adjustments:* Both the number of adjustments, and its logarithmic counterpart, show interesting behavior for their effect on FVA. Whenever an enrichment has been adjusted a great number of times, it can either be a predictor of both a good or bad FVA. This could be through the fact that a planner often receives new information and updates the forecast accordingly, or they tinker with the forecast for a long time to find the best value. When the number of adjustments is closer to 1, it is not a clear predictor of FVA.
- Time-Related:
  - *Time-lag:* Time-lag seems to have a small effect on the predicted FVA. The behavior seems to follow the number of adjustments, i.e., high values can either indicate that the FVA will be good or bad.
  - *Previous FVA:* A product-customer's previous FVA is a good predictor for an upcoming FVA, especially when previous enrichments were successful. Planners are thus consistent in improving FVA through either good information availability, easy forecastable products or some other factors beyond the scope of this dataset. For this feature, AutoFeat has also added its squared variant, looking at absolute values.

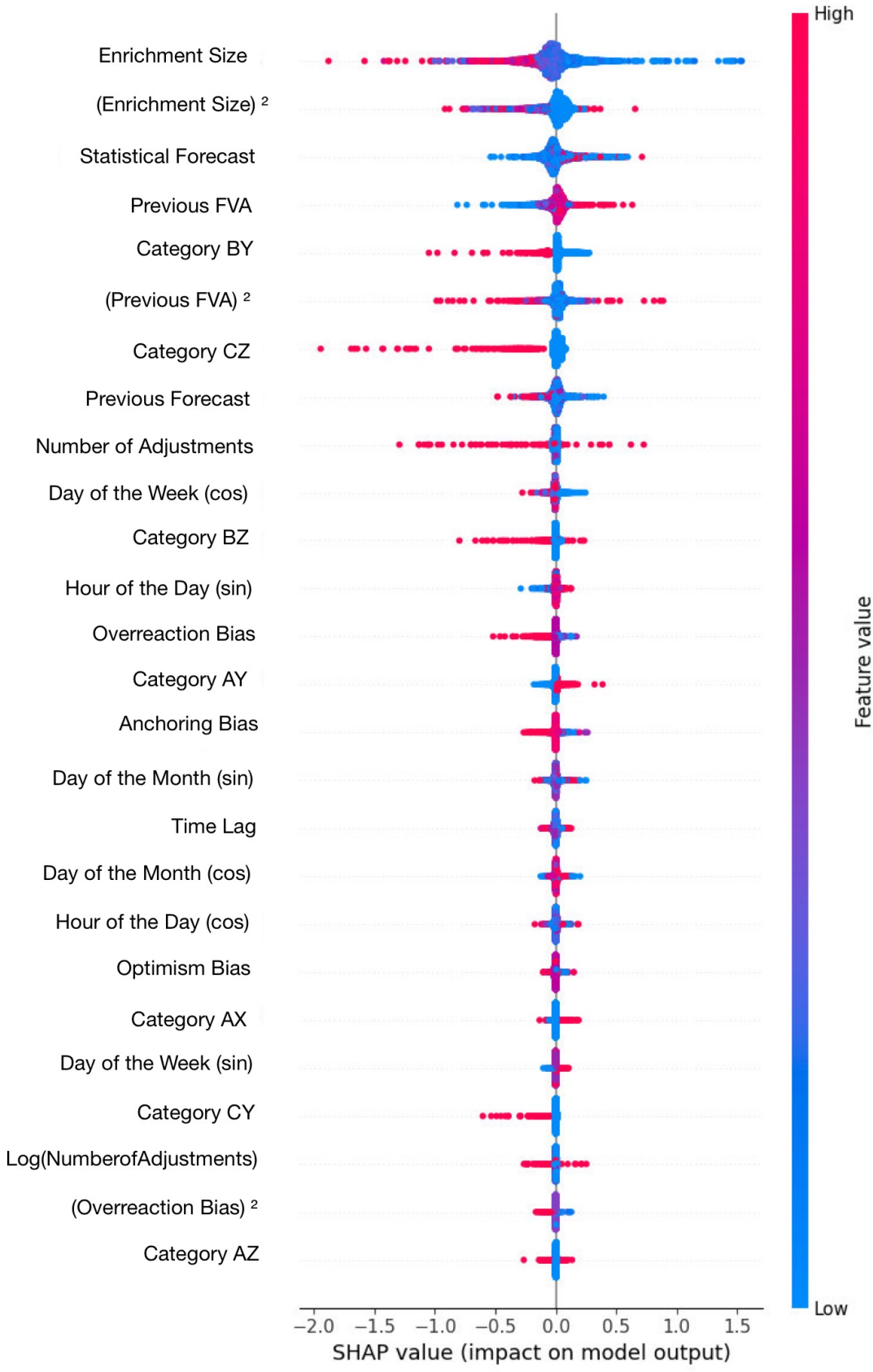


Figure 6.6: Shapley Summary Plot

  - *Time:* Due to the sine-cosine encoding, it is hard to interpret the feature values for these features. They are also scattered around the plot, making their interpretation hard. We do see that the cosine of the day of the week is the strongest predictor out of all time features. However, feature values are scattered.

- Product-Related:
  - *Product Category:* Product categories are scattered around the SHAP summary plot. We can see that product categories BY, CZ, and CY are clear predictors of a low FVA. This is not surprising since these categories all deal with relatively high variance, at which planners struggle (Sanders, 1992; Franses, 2013). Categories AY & AX are enrichments predictors of a better FVA. However, these predictors are less important according to SHAP values. These categories concern the products with the highest value and are thus important to enrich correctly.

- Planner-Related:
  - *Optimism Bias:* This feature is not considered to be very important in predicting the FVA, since it is reported in the lower region of the summary plot. There is also not a clear relation between the height of the value and the predicted FVA.
  - *Anchoring Bias:* Just like the statistical forecast, the colors on the graph do not clearly indicate the values for the feature. There are a large number of red dots that indicate that the predicted FVA will be lower.
  - *Overreaction Bias:* The most important planner-related feature is a planner's overreaction bias. There is a clear indication that a high overreaction bias predicts a lower FVA.

From the analysis of Figure 6.6, it is clear that 'Forecast-related' features are the most important in predicting FVA. 'Product-related' features are also very important, given that the product falls in a certain product category. 'Time-related' features are not strong predictors apart from the lagged FVA and overreaction bias seems to be the most important 'Planner-related' feature.

For each enrichment, a separate SHAP plot could be made to understand the decision of the prediction model. Within Figure 6.7, a single enrichment is explained. As a basis, the model makes a default prediction of -0.037 as seen on the X-axis. From this initial value, it moves the expected FVA based on the features. As expected, the enrichment size is the dominant factor for the adjustment. Since we seem to be dealing with an averagely-sized, downward adjustment, the FVA is improved slightly. We know this since the base feature indicates a feature value of -1.071 (scaled) while the squared features indicate a value close to zero. The fact that one adjusts downwards is very good, while the average size is not. Furthermore, the lagged FVA and its squared counterpart both predict an improvement in FVA. The previous forecast is slightly below average, which predicts a good FVA and the statistical forecast is slightly above average and this is also beneficial for the FVA. Furthermore, there is quite some time-lag for this prediction which reduces the FVA slightly. Lastly, the product is not a BY category, and since this is not the case, it predicts a slightly higher FVA.

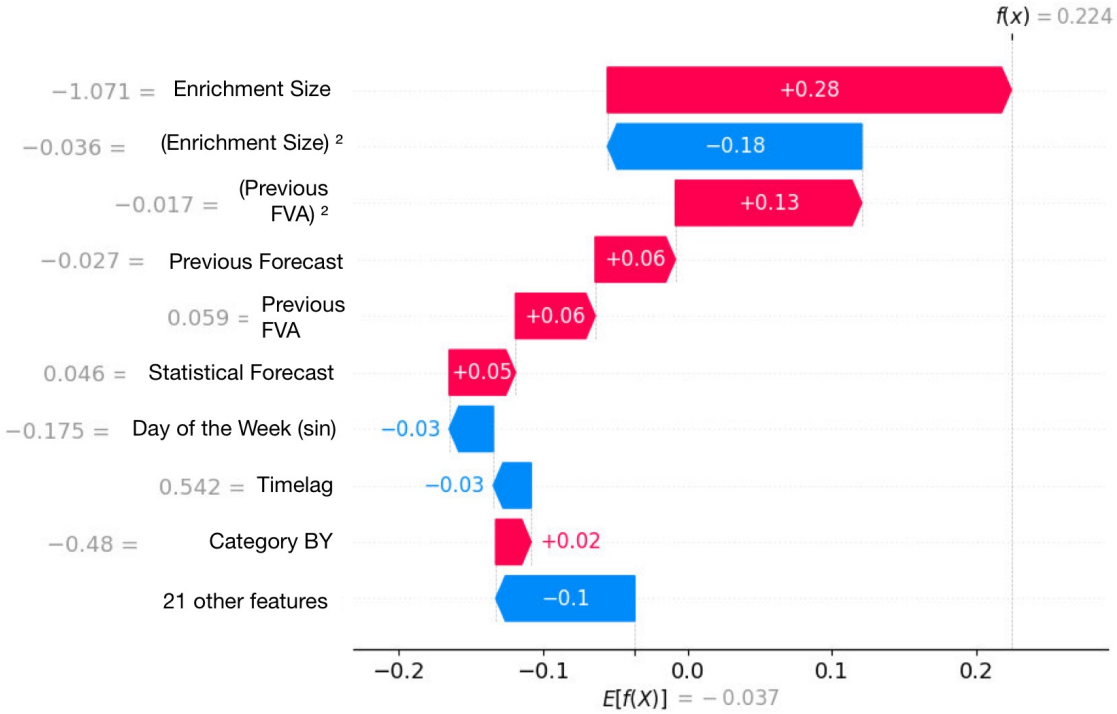


Figure 6.7: Individual Enrichment - SHAP

6.4 Notifications

A prediction model that is accurate in predicting the FVA is only useful up to a certain level. The real value comes through its ability to preventively identify when enrichments are expected to reduce the FVA and notify the planner accordingly. Below, we will explore when planners should be alerted and what gains this could result in.

When we look at the overall statistics of BU 1 and 3, the average FVA is slightly below zero (-0.005), which indicates that on average, planners reduce the forecasting accuracy. In most cases, planners will subtract or improve the FA by small values, but in some cases, they reduce the value significantly. Ideally, you want to alert planners when you identify that an enrichment might damage the accuracy. The LightGBM prediction model is able to predict the FVA based on the enrichment factors, while SHAP is able to explain why this prediction leads to a certain value. In this section, we have to identify at what value we notify planners about a potential bad enrichment, the so-called threshold FVA.

Based on the literature about algorithm aversion by Dietvorst et al. (2015), we know that giving advice can be a difficult task. Humans are more inclined to receive advice from other humans than from algorithms, even if the algorithms are better. This is exacerbated when the algorithm makes mistakes. The threshold FVA determines when the planner receives an alert. We consider an alert to be correct when the actual FVA also turns out to be below zero. For instance, when the model predicts an FVA of -2.5, an actual FVA of -0.5 is considered correct. We also do not want to alert planners too frequently, since they are more likely to discard notifications. Thus, we must identify a threshold FVA that balances the number of notifications and the percentage of correct notifications.

Whenever a notification is sent, the Shapley values for the individual forecast can be

used to explain why the model predicts a low value. As an example, Figure 6.8 shows an enrichment that the prediction model selected for a notification. This notification explains the low predicted FVA through a number of features. The planner gets notified about features that they can adjust, like the enrichment size, but also about features that are related to their overreaction bias and the hierarchy level at which they adjust. Currently, this enrichment is executed at a very high hierarchy level, which means that it will affect a lot of products. Based on this advice, the planner should reconsider executing the enrichment at a lower hierarchy level or reconsider the enrichment size. When the planner would look at a lower hierarchy level they could be more precise in adjusting each product individually based on the information available for each product.

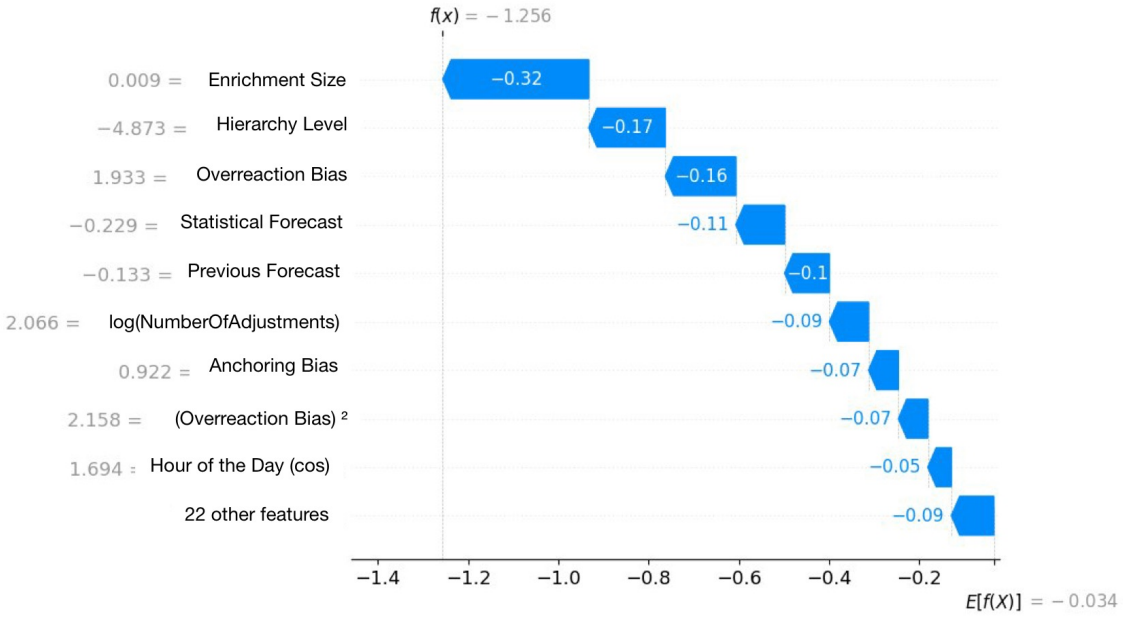


Figure 6.8: Individual Notification

Within this thesis, we focus on an implementation plan without implementing it in Company A. The results from this section are drawn from the testing set of the LightGBM model. Therefore, we are not aware of how effectively planners will incorporate the advice from the alerts, and if this will result in a positive FVA. Therefore, we will set a baseline by using the statistical forecast. Whenever the model identifies a 'harmful enrichment', the enrichment is replaced by the original statistical forecast, resulting in an FVA of 0. This method is robust and easy to implement, since planners' behavior is unknown.

In Table 6.13, we have tested a range of FVA thresholds for quantity of notifications and its accuracy. Like displayed in Table 6.6, most predictions are located around an FVA of 0. The number of notifications sent will reduce greatly when the threshold value is moved further down. Not all notifications that are sent are correct and the extent to which this happens is seen in the '% Incorrect' column. For a threshold FVA of -1.25, this value is lowest, and it increase when the threshold is moved further up. Without notifications, the FA of enriched forecasts is 0.478 and the statistical FA is 0.483. This results in a negative FVA of -0.005. Through notifications, we are able to improve the forecast accuracy for every threshold value. The FVA also becomes positive for all options.

According to Table 6.13, a threshold value of 0.00 will result in the best performance,

since the FA is highest. However, this would lead to a large number of alerts of which many are incorrect. This threshold value should be preferred when the planners' acceptance of the model is not taken into account. When one would implement this model at an organization, we have to understand that: i) planners could have the ability to improve the forecast beyond the statistical forecast and ii) that they should not receive too many alerts. To see at what threshold level the notifications could be most effective, we also calculated the 'FVA per notification'. As the name suggest, this column divides the improvement of the FVA compared to the number of notifications required to achieve this. At a threshold level of -0.50, the FVA improves most per notification.

Threshold	**% Notification**	**Notifications**	**% Incorrect**	**FA**	**FVA**	**FVA per Notification**
-1.50	1.42	70	15.71	0.484	0.002	$8.21 * 10^{-5}$
-1.25	1.94	96	13.54	0.468	0.003	$7.63 * 10^{-5}$
-1.00	2.77	136	14.60	0.487	0.005	$6.49 * 10^{-5}$
-0.75	4.25	209	17.62	0.490	0.007	$5.45 * 10^{-5}$
-0.50	7.27	356	20.61	0.511	0.028	$9.10 * 10^{-5}$
-0.25	14.05	686	27.23	0.524	0.042	$6.66 * 10^{-5}$
0.00	50.56	2222	28.99	0.532	0.050	$2.43 * 10^{-5}$

Table 6.13: Notification Summary Table

Further research should be conducted on planners' ability to actually incorporate the notifications from such a model and see the extent by which they can improve their FVA. Based on their ability and acceptance of the model, the threshold shall vary. However, if we utilize the method described before (i.e., a notification will result in the enriched forecast to be reset to the statistical forecast), we achieve an FA that is better than either the original enriched or statistical forecasts.

7 Conclusion

In conclusion, we will start by answering the research questions that have been defined in the introduction, providing answers, and putting them in relation to the literature. After which, we delve into the scientific and practical implications combined with recommendations for application.

RQ1: What patterns do planners exhibit when adjusting system-generated forecasts and how does this influence the overall forecasting accuracy?

There are mixed results regarding the ability of planners to add value to the statistical forecast. Literature reveals a pattern of over-optimism among planners in their forecasts (Fildes et al., 2009; Eroglu & Croxton, 2010; Syntetos et al., 2009; Trapero et al., 2013), often anticipating higher sales quantities than statistical methods would predict. An effective forecast enrichment should be an intervention based on a statistical method when the human planner can incorporate a clear piece of information that is not modeled by the system. More often, good enrichments are adjustments downwards compared to the statistical forecast and these have been found to increase the accuracy (Fildes et al., 2009). However, for planners, it is also more difficult to model a noisy time series since they are inclined to search for patterns in the noise (Sanders, 1992). Planners also adjust forecasts more frequently and significantly when the timelag is lower. Furthermore, planners can have a large number of biases, since humans have cognitive limitations that prevent full rationality. Additionally, humans can be hesitant to incorporate external advice, especially when this advice is not generated by another human (Dietvorst et al., 2015).

RQ2: How can tree-based models and neural networks be applied to estimate enrichment quality and how can their results be explained?

As we have seen in the literature, these two models go about their predictions in a completely different manner. Both kinds of models can be used for either regression or classification tasks. Given the fact that we have labeled training data available both can be utilized in a supervised manner. Decision trees purify their data by creating splitting points of nodes based on certain criteria that maximize purity. There are a large number of tree-based models available, of which LightGBM (Ke et al., 2017) has a preference due to its properties of Gradient-based One-Sided Sampling (GOSS) and Exclusive Feature Bundling (EFB).

Neural networks operate by executing mathematical transformations within the nodes and require tuning of weights and biases in order to create accurate predictions. These are trained through the use of backpropagation. Within the neural network, the user can specify the number of nodes, activation functions, and the number of hidden layers, together with the dropout rates. Overall, neural networks can map more intricate relations, at the cost of potential overfitting and more computational resources.

In order to explain the decisions of the models, SHAP (Lundberg et al., 2017) can be used to find the contribution of each individual feature. In turn, it is able to explain which features are deemed to be important to the overall prediction and which features are not. It can be investigated at either model or individual prediction level.

RQ3: What characteristics from human forecast enrichments that are described in the literature show up in this dataset?

Conclusions regarding a planner's ability to add value have often been mixed within this field as results from Petropoulos et al. (2016), Goodwin et al. (2007), Sanders & Ritzman (1995) and Belvedere & Goodwin (2017) have shown. Building on these mixed results, the BUs and planners show largely varying results, where in some cases value is being added and in other cases, value is being subtracted. Some patterns that underline these results are found amongst planners at Company A, through their over-optimistic behavior in their forecast enrichments. This bias is also found in the results of Fildes et al. (2009); Eroglu & Croxton (2010); Syntetos et al. (2009); Trapero et al. (2013). Even though planners within each BU exhibit this bias, the extent by which it occurs varies, with much higher values in BU 2.

These results are also found regarding overreaction bias, with a clear correlation between the level of optimism and overreaction bias. The low and non-significant values for anchoring bias indicate that planners do not rely strongly on the statistical forecast. This opposes the findings by Sanders (1992) and Baets & Harvey (2018). Although planners adjust forecasts mainly for products that are highly variable and high in value which is suggested by Scholz-Reiter et al. (2012), their performance varies and does not consistently improve the accuracy of these categories, except for category CX. Thus, there is no clear evidence that planners are able to complement the statistical forecasting methods in product categories that have higher variances. In most product categories, planners reduce the forecasting accuracy compared to the statistical baseline forecast. This is in line with the conclusion from Sanders (1992) that humans are less accurate in noisy data.

RQ4: What features are of great importance to the quality of forecast enrichments?

Khosrowabadi et al. (2022) researched what kind of features have an influence on good forecast enrichments. Within their paper, they find out that the adjustment direction, sales quantity, price, freshness, and enrichment size are important factors. In this thesis, different features have been taken into account. It expands on the previous research by combining both planner, forecast, time, and product-related features in a single prediction model. This gives us an understanding of which kinds of features are the most important indicators. After selecting the LightGBM model, with capped FVA and 1-step feature engineering, we are able to predict the FVA of a forecast enrichment at an accuracy level of 0.432 (MAE) based on a number of features. The most important features include Enrichment Size, Lagged FVA, Statistical Forecast, Previous Forecast, Product Category, Number of Adjustments, and Overreaction Bias. A positive FVA is more likely when the enrichment size is large and negative, the FVA has been high in the past, the statistical forecast is high, and the previous forecast is low.

RQ5: In what conditions could planners be notified about their expected enrichment quality?

Providing humans with advice can be a complex subject as seen in the paper by Dietvorst et al. (2015). Humans seem to distrust algorithms, especially when they see the algorithm struggle in a certain scenario. Therefore, one must be careful in notifying planners by preventing incorrect alerts. This dataset contains numerous enrichments that

lead to a decrease in FVA. When combined with the mean average prediction error of the model, it is advisable not to alert planners in every instance where a negative FVA is projected, since there will be a large number of notifications of which many will be false (i.e., FVA turns out to be positive). In order to solve these two problems, certain threshold values are explored. The predictions can be explained through their individual Shapley plots that show what features cause the prediction to be below the threshold. Based on the planners' ability to effectively enrich these forecasts and their acceptance of the model, a threshold value could easily be selected.

Taken together, we can conclude that behaviors from planners in literature also are represented within the process industry. We support the literature's mixed findings regarding planners' forecast-enhancing capabilities. A LightGBM model has been created to predict the quality of an enrichment based on features that are related to the forecast, time, product, or planner. We expand the number of features that are predictors of FVA and create a method through which planners can be alerted about a predicted bad forecast enrichment.

8 Discussion

8.1 Practical Implications

Besides academic findings, there are also practical implications for EyeOn. We have seen that planners within Company A do not always add value through their enrichments. This situation has been clear for consultants at EyeOn for a long while. However, it was not clear what features affected the accuracy and what their importances were. The first practical implication is that the findings in this thesis give EyeOn a clearer overview of how various features affect accuracy of forecast enrichments. This expanded knowledge makes consultants able to quicker identify reasons for lacking forecast enrichments for their customers.

Furthermore, through this prediction model, EyeOn could expand its Jedox software to include a notification system to prevent planners from drastically reducing the FVA. EyeOn should investigate how planners would respond to the notifications and if they are able to effectively improve the FVA. If they would like to utilize the system in its current state, we have shown that certain enrichments could be replaced by the statistical forecast to achieve a higher accuracy than either method separately. EyeOn could try to sell an additional forecast enrichment service to their customers. Since it is a general model that can be applied to several companies. EyeOn would be able to provide additional service and business insights to their customers, while the customers will enjoy better forecasting performance. Through their previous research on decisional guidance, they have a lot of information to effectively guide planners.

This thesis is also a blueprint from which the system can be implemented for other customers. Once they have data available from another organization, they can investigate what features they can utilize. The data only has to be cleaned and categorical columns have to be indicated, after which the LightGBM model can be tuned and the SHAP values can be calculated to give them the required insights. This can then be translated to an appropriate notification threshold for that specific customer.

8.2 Limitations

The results are based on the results of a single company within the process sector. Therefore, we know that the results are true for this organization, but we do not know for sure if they are mirrored by others. Additionally, there are some oddities within the dataset. For BU 2, the hierarchy levels do not completely line up with the actual customer-product level, creating skewed results. Therefore, this data has not been utilized in the final machine learning models, and is the overall dataset reduced in size. There are also a lot of data points that are removed from the dataset due to missing sales data, or sales quantities of zero. Lastly, planners have the ability to upload all their enrichments in a single file. This might be very useful for the planner at Company A but does not give EyeOn the full picture of their enrichment behavior.

The limited data points also play a part in the next limitation, namely the model accuracy and hyperparameter tuning. Machine Learning models perform better when they are being fed larger datasets and have sufficient time to train. The dataset in this thesis is relatively small for a machine learning model to identify all patterns and thus accuracy could be improved with more input data. Additionally, the hyperparameter tuning uses a

two-step grid search. This method is used to reduce computational intensity since further specifying hyperparameter values would exponentially increase run time. In turn, it is not sure the hyperparameter will find the optimal parameter tuning, however, it will find a pretty good tuning. Either expanding runtime, using more powerful hardware or a different optimization methodology could improve results.

The notification methodology has also not been tested by actual planners to capture their behavior toward it. The recommendations are based on guidelines and results from the testing dataset. Planners will not accept the alert in every situation, and they will not perfectly adjust their enrichment every time. Thus, an experiment or a roll-out at a customer could confirm these results and show how planners handle this advice in a real business setting.

8.3 Future Research

This thesis is a stepping stone within the field of behavioral operations management. From this work, several new paths could be explored to deepen our knowledge about effective forecast enrichments.

The main suggestion is focused on exploring how planners would react to the alerts suggested by the algorithm. Prior work by Dietvorst et al. (2015) has shown that humans have trouble working with advice from an algorithm, especially when they stumble. However, since then, more research has been executed on decisional guidance, and the rise of AI models to the public knowledge might affect positively have affected people's perception. People might be more willing to accept advice through increased trust in AI and improved clear guidance on how they should adjust forecasts. Furthermore, it should be investigated if planners are able to effectively add value based on notifications.

Additionally, prior research has found that there are more product-related features that can predict enrichment quality (Khosrowabadi et al., 2022). However, the dataset in this thesis did not include any. Thus, one could for instance not discern between the value of different products or their levels of perishability. Hence, we do not know where these features slot in among the importances of the features in this thesis. Another feature that is common but lacking in the dataset is a promotion indicator. Within the process industry, this is also not applicable, however, in other industries, one could include all these features for a complete overview.

Lastly, we briefly touched during this thesis on the comments that planners can add to their enrichments. An expansive analysis using NLP could provide the company with more clear insights into their enrichment behavior and improve the explanation for their performance. Another option could be to create a predefined set of comments that planners can select from. Hereby, the comments will be more structured and generalizable.

References

Baecke, P., Baets, S. D., & Vanderheyden, K. (2017, 9). Investigating the added value of integrating human judgement into statistical demand forecasting systems. *International Journal of Production Economics*, *191*, 85-96. doi: 10.1016/j.ijpe.2017.05.016

Baets, S. D., & Harvey, N. (2018, 4). Forecasting from time series subject to sporadic perturbations: Effectiveness of different types of forecasting support. *International Journal of Forecasting*, *34*, 163-180. doi: 10.1016/j.ijforecast.2017.09.007

Belvedere, V., & Goodwin, P. (2017, 7). The influence of product involvement and emotion on short-term product demand forecasting. *International Journal of Forecasting*, *33*, 652-661. doi: 10.1016/j.ijforecast.2017.02.004

Bescond, P.-L. (2020, 6). *Cyclical features encoding, it's about time!* Retrieved from `https://towardsdatascience.com/cyclical-features-encoding-its-about-time-ce23581845ca`

Bex, T. (2021). *Kaggler's guide to lightgbm hyperparameter tuning with optuna in 2021.* Retrieved 2023-02-11, from `https://towardsdatascience.com/kagglers-guide-to-lightgbm-hyperparameter-tuning-with-optuna-in-2021-ed048d9838b5`

Binomial test. (2008). In *The concise encyclopedia of statistics* (pp. 47–49). New York, NY: Springer New York. Retrieved from `https://doi.org/10.1007/978-0-387-32833-1_36` doi: 10.1007/978-0-387-32833-1_36

Bishop, C. M. (2006). *Pattern recognition and machine learning.*

Breusch, T. S., & Pagan, A. R. (1979). *A simple test for heteroscedasticity and random coefficient variation* (Vol. 47).

Broeke, M. V. D., Baets, S. D., Vereecke, A., Baecke, P., & Vanderheyden, K. (2019, 9). Judgmental forecast adjustments over different time horizons. *Omega (United Kingdom)*, *87*, 34-45. doi: 10.1016/j.omega.2018.09.008

Brownlee, J. (2020, 6). *How to use standardscaler and minmaxscaler transforms in python.* Retrieved from `https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/`

Chollet, F., et al. (2015). *Keras.* `https://keras.io`.

Dietvorst, B. J., Simmons, J. P., & Massey, C. (2015). Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, *144*, 114-126. doi: 10.1037/xge0000033

Eroglu, C., & Croxton, K. L. (2010, 1). Biases in judgmental adjustments of statistical forecasts: The role of individual differences. *International Journal of Forecasting*, *26*, 116-133. doi: 10.1016/j.ijforecast.2009.02.005

Eyeon. (2022). Retrieved from `https://eyeonplanning.com`

Field, A. P. (2009). *Discovering statistics using spss : (and sex and drugs and rock 'n' roll).* SAGE Publications.

Fildes, R., Goodwin, P., Lawrence, M., & Nikolopoulos, K. (2009, 1). Effective forecasting and judgmental adjustments: an empirical evaluation and strategies for improvement in supply-chain planning. *International Journal of Forecasting*, *25*, 3-23. doi: 10.1016/j.ijforecast.2008.11.010

Franses, P. H. (2013). Improving judgmental adjustment of model-based forecasts. *Mathematics and Computers in Simulation*, *93*, 1-8. doi: 10.1016/j.matcom.2012.11.007

Goodwin, P. (2002). Integrating management judgment and statistical methods to improve short-term forecasts. *Omega*, *30*, 127-135. Retrieved from `www.elsevier.com/locate/dsw`

Goodwin, P., & Fildes, R. (1999). Judgmental forecasts of time series affected by special events: Does providing a statistical forecast improve accuracy? *Journal of Behavioral Decision Making*, *12*, 37-53. doi: 10.1002/(SICI)1099-0771(199903)12:1<37::AID-BDM319>3.0.CO;2-8

Goodwin, P., Fildes, R., Lawrence, M., & Nikolopoulos, K. (2007, 7). The process of using a forecasting support system. *International Journal of Forecasting*, *23*, 391-404. doi: 10.1016/j.ijforecast.2007.05.016

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2019). *Multivariate data analysis* (8th ed.). Cengage Learning. Retrieved from `www.cengage.com/highered`

Horn, F., Pack, R., & Rieger, M. (2019). The autofeat python library for automated feature engineering and selection. In *Joint european conference on machine learning and knowledge discovery in databases* (pp. 111–120).

Jain, A. K., Mao, J., & Mohiuddin, K. (1996). Artificial neural networks: A tutorial. *Computer*, *29*, 31-44. doi: 10.1109/2.485891

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*, 685695. Retrieved from `https://doi.org/10.1007/s12525-021-00475-2` doi: 10.1007/s12525-021-00475-2/

Jijo, B. T., & Abdulazeez, A. (2021, 3). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*, 20-28. doi: 10.38094/jastt20165

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). *Lightgbm: A highly efficient gradient boosting decision tree.* Retrieved from `https://github.com/Microsoft/LightGBM.`

Khosrowabadi, N., Hoberg, K., & Imdahl, C. (2022, 12). Evaluating human behaviour in response to ai recommendations for judgemental forecasting. *European Journal of Operational Research*, *303*, 1151-1167. doi: 10.1016/j.ejor.2022.03.017

Lee, W. Y., Goodwin, P., Fildes, R., Nikolopoulos, K., & Lawrence, M. (2007, 7). Providing support for the use of analogies in demand forecasting tasks. *International Journal of Forecasting*, *23*, 377-390. doi: 10.1016/j.ijforecast.2007.02.006

Lim, J. S., & O'Connor, M. (1996). *Judgmental forecasting with time series and causal information* (Vol. 12).

Lin, V. S., Goodwin, P., & Song, H. (2014). Accuracy and bias of experts' adjusted forecasts. *Annals of Tourism Research*, *48*, 156-174. doi: 10.1016/j.annals.2014.06.005

Lundberg, S. M., Allen, P. G., & Lee, S.-I. (2017). A unified approach to interpreting model predictions.. Retrieved from https://github.com/slundberg/shap

Microsoft. (n.d.). *Parameter tuning.* Retrieved 2023-02-11, from https://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html

O'Connor, M., Remus, W., & Griggs, K. (1993). Judgemental forecasting in times of change. *International Journal of Forecasting*, *9*, 163-172.

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

Petropoulos, F., Fildes, R., & Goodwin, P. (2016, 3). Do 'big losses' in judgmental adjustments to statistical forecasts affect experts' behaviour? *European Journal of Operational Research*, *249*, 842-852. doi: 10.1016/j.ejor.2015.06.002

Quinlan, J. R. (1986). *Induction of decision trees* (Vol. 1).

Radhakrishnan, P. (2017). *What are hyperparameters? and how to tune the hyperparameters in a deep neural network.* Retrieved 2023-02-12, from https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a

Ranjan, C. (2019, 7). *Rules-of-thumb for building a neural network.* Retrieved from https://towardsdatascience.com/17-rules-of-thumb-for-building-a-neural-network-93356f9930af

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, 8). "why should i trust you?" explaining the predictions of any classifier. In (Vol. 13-17-August-2016, p. 1135-1144). Association for Computing Machinery. doi: 10.1145/2939672.2939778

Sanders, N. R. (1992). Accuracy of judgmental forecasts: A comparison. *Omega*, *20*, 353-364. doi: https://doi.org/10.1016/0305-0483(92)90040-E

Sanders, N. R., & Manrodt, K. B. (2003, 12). The efficacy of using judgmental versus quantitative forecasting methods in practice. *Omega*, *31*, 511-522. doi: 10.1016/j.omega.2003.08.007

Sanders, N. R., & Ritzman, L. P. (1995). Bringing judgment into combination forecasts. *Journal of Operations Management*, *13*, 311-321.

Scholz-Reiter, B., Heger, J., Meinecke, C., & Bergmann, J. (2012, 4). Integration of demand forecasts in abc-xyz analysis: Practical investigation at an industrial company. *International Journal of Productivity and Performance Management*, *61*, 445-451. doi: 10.1108/17410401211212689

Shapiro, S. S., & Wilk, M. B. (1965, 12). An analysis of variance test for normality (complete samples)†. *Biometrika*, *52*(3-4), 591-611. Retrieved from https://doi.org/10.1093/biomet/52.3-4.591 doi: 10.1093/biomet/52.3-4.591

Shapley, L. (1952). *A value for n-person games.*

statsmodels.stats.diagnostic.linear_rainbow. (2023, 5). Retrieved from https://www.statsmodels.org/stable/generated/statsmodels.stats.diagnostic.linear_rainbow.html

Syntetos, A. A., Nikolopoulos, K., Boylan, J. E., Fildes, R., & Goodwin, P. (2009, 3). The effects of integrating management judgement into intermittent demand forecasts. *International Journal of Production Economics*, *118*, 72-81. doi: 10.1016/j.ijpe.2008.08.011

Trapero, J. R., Pedregal, D. J., Fildes, R., & Kourentzes, N. (2013, 4). Analysis of judgmental adjustments in the presence of promotions. *International Journal of Forecasting*, *29*, 234-243. doi: 10.1016/j.ijforecast.2012.10.002

Tversky, A., & Kahneman, D. (1974). *Judgment under uncertainty: Heuristics and biases* (Vol. 185).

Utes, J. M. (1982, 1). The rainbow test for lack of fit in regression. *Communications in Statistics - Theory and Methods*, *11*, 2801-2815. doi: 10.1080/03610928208828423

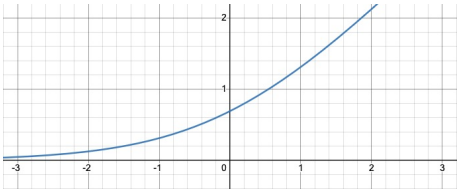
Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*(6), 80–83. Retrieved 2023-06-18, from `http://www.jstor.org/stable/3001968`

Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., & Tang, H. (2018, 11). Predicting diabetes mellitus with machine learning techniques. *Frontiers in Genetics*, *9*. doi: 10.3389/fgene.2018.00515

A Machine Learning

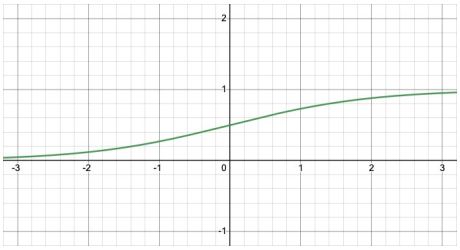
A.1 Activation Functions

In Figure A.1, the different available activation functions are visualized.



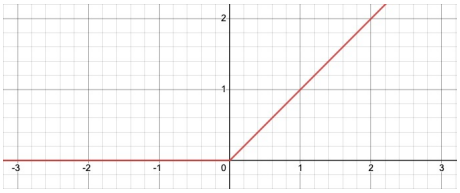
(a) SoftMax Activation

$$f(x) = log(1 + e^x)$$



(b) Sigmoid Activation

$$f(x) = \frac{e^x}{e^x + 1}$$



(c) ReLU Activation

$$f(x) = max\{0, x\}$$

Figure A.1: Activation Functions

A.2 Backpropagation

This section of the Appendix will focus on how backpropagation works to optimize the parameters in the Neural Network. To optimize an unknown bias in the network, it first has to be initialized at a certain value (0 by default). We shall assume that all parameters in the neural network have been optimized, apart from the last bias. One could look at Figure 3.1 to visualize this problem. All weights and biases are optimized apart from the bias that is added in the end node. After the bias in this node has been initialized with the value of 0, the sum of the squared residuals is calculated. The residuals are the difference between the observed values and the predicted values. This difference is calculated between each observed and predicted value. Each residual is squared after which it is summed, then one can find the overall model quality. Now we know what the sum of squared residuals (SSR) is for a bias of zero, the bias parameter can be adjusted to another value that is tested to see what its SSR is.

$$SSR = \sum_{i=1}^{n} (Observation_i - Prediction_i)^2 \tag{A.1}$$

However, testing a large number of values can be computationally intensive, and thus the gradient descent method is preferred. To utilize the gradient descent one has to employ

the chain rule:

$$\frac{dSSR}{dBias} = \frac{dSSR}{dPrediction} * \frac{dPrediction}{dBias} \tag{A.2}$$

$$\frac{dSSR}{dPrediction} = \sum_{i=1}^{n} 2 * (Observation_i - Prediction_i) * -1 \tag{A.3}$$

$$\frac{dPrediction}{dBias} = \frac{d}{dBias}(\sum(input) + bias) = 1 \tag{A.4}$$

$$\frac{dSSR}{dBias} = \sum_{i=1}^{n} 2*(Observation_i - Prediction_i)*-1*1 = \sum_{i=1}^{n} -2*(Observation_i - Prediction_i) \tag{A.5}$$

In Equation A.2, the basic chain rule is shown, given that one wants to differentiate the SSR over the bias. Equation A.3 shows the derivation of the first fraction. One multiplies the residuals by 2 and reduces the exponent by 1, and multiplies it by the derivation of the inner part ($Observation_i - Prediction_i$) for prediction, which is -1. Equation A.4 derives the prediction over the bias. As we have seen in the previous section, the prediction can be seen as the weighted activation function outputs plus the bias that is introduced at the end node. This leads to an answer of 1. Equation A.5 shows the final solution for the chain rule.

Now gradient descent can be applied. First, $\frac{dSSR}{dBias}$ has to be filled in for the initial bias level. This output will be the slope in the gradient descent. The slope is multiplied by the learning rate to find the step size. Finally, one will take the initial bias value and decrease it by the step size, which creates a new bias that can be re-introduced in the system. This process is repeated until the step size is close to zero. At that point, the system is not making any real improvements anymore, and the bias is finalized.

B Forecasting Behavior

B.1 Forecasting Behavior

This section will explore the time-related behavior of planners. Within the models, we have included the hour of the day, day of the week and day of the month as predictors of FVA. We will identify when and how planners adjust forecasts.

For each enrichment, we have a timestamp feature, that saves the exact time (UTC) at which a forecast enrichment has been submitted. Through this information, we can see during what days of the week an enrichment has taken place. For each BU, enrichments can be made on different days, some of which include weekend days, but these are limited. However, all enrichments are labeled using UTC and this could distort the picture. For instance, planners in the US or Japan might adjust forecasts on Monday or Friday but they are reported in UTC as being adjusted on the weekend. We do not have data on each planner's location. The specific days at which forecasts are adjusted seem to be very different based on the BU and do not seem to show clear general phenomena.

When zooming in on the specific time during which adjustments are made, we can see that most adjustments are made in the morning for each BU. Within BU 3, all planners

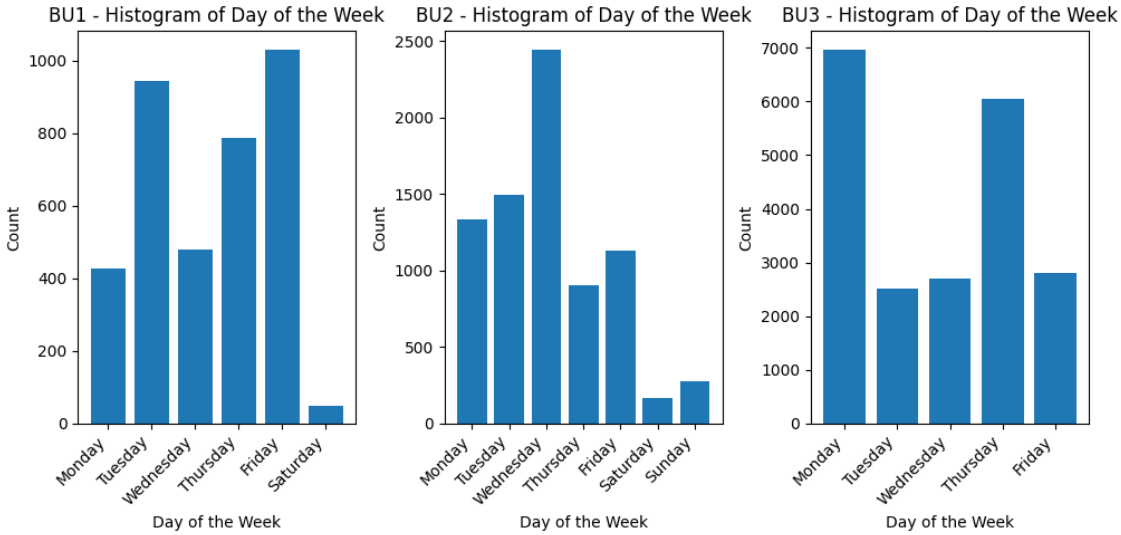


Figure B.1: Adjustment Count Per Day of the Week

likely operate within the same timezone, since there are no forecasts in the evenings or nights. For BU 2, the large majority of adjustments are made during 'Western-European working times', but there are still several adjustments executed at night. BU 1 seems to have a slight dip during the night but adjustments seem to be made around the clock.

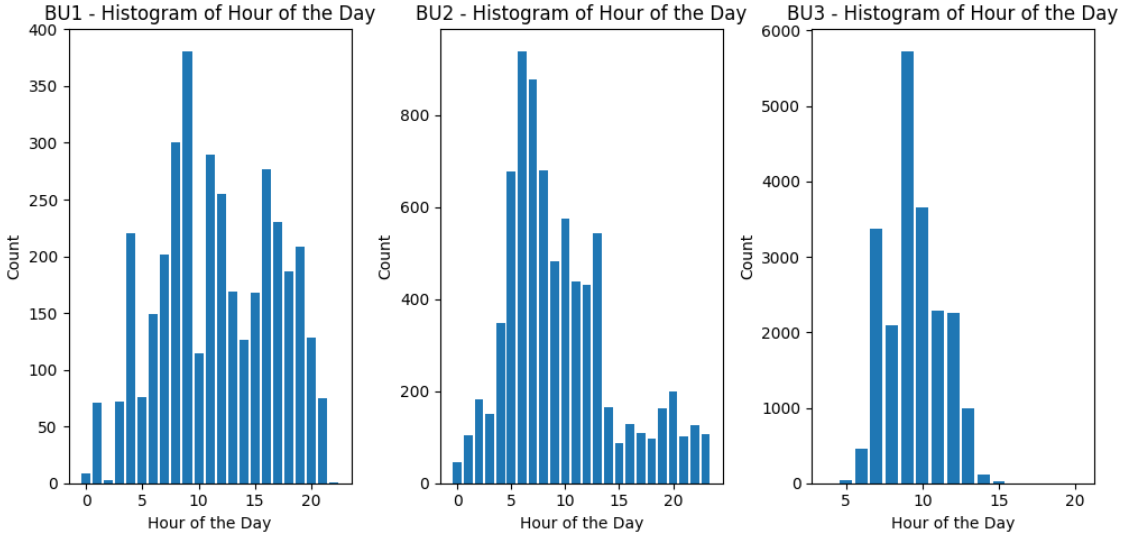


Figure B.2: Adjustment Count Per Hour of the Day

Lastly, we can have a look at probably the most interesting set of graphs regarding the timing of adjustments, the day of the month at which the enrichments are executed. For each of these BUs, EyeOn provides a monthly statistical forecast, which the BUs can adjust on a monthly level. Adjustments that are made during this month, will be delivered in the following month. Within these graphs, we can see several interesting things happening. Firstly, for BU 1 and BU 2, we can see a peak at the beginning of the month. This corresponds with Market Intelligence (MI) updates that the planners receive. Based on this, planners make the adjustments within the Jedox software. We can see a large peak across

all three BUs towards the end of the month. This can be either due to 'deadline-seeking behavior' or the wait until the last moment to utilize the most up to date information. Additionally, it could be the case that planners maintain their adjustments in Microsoft Excel during the month and upload them at the end of the month.

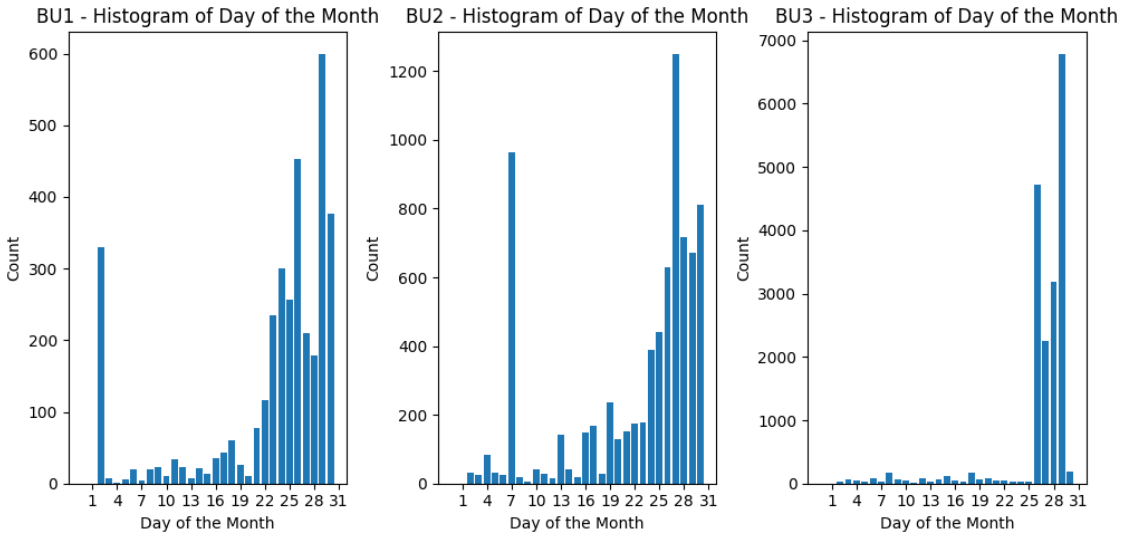


Figure B.3: Adjustment Count Per Day of the Month

We can further zoom in on a single planner to improve the understanding of their behavior. As a prime subject, we will select planner '24'. This planner works within BU 3 of Company A and has made by far the most enrichments in the last year. Within 11 months, they managed to execute over 16,000 enrichments. Planner 24 does execute various adjustments during the month, however, those are limited to a few adjustments per day. At the end of every month, they can adjust several thousand forecasts within 90 minutes. We cannot be sure if they input everything by hand or upload an Excel file into Jedox, however, the interval between adjustments is too small for planners to decide on order quantities. Within Table B.1 we have identified the peaks by filtering out the days on which more than 100 adjustments are made by this planner. Apart from August and November, planner 24 enriches the forecasts, in the same manner, every month.

Date	**First Time**	**Last Time**	**# Adjustments**	**Total Time (min:sec)**	**Mean Interval (sec)**
25-02-2022	07:46:23	08:18:29	1,999	32:06	0.96
28-03-2022	08:59:29	09:54:57	2,170	55:28	1.53
27-04-2022	12:18:57	13:09:28	2,322	50:31	1.31
30-05-2022	06:54:11	08:13:05	2,318	78:54	2.04
28-06-2022	08:44:51	10:19:50	2,170	94:59	2.63
28-07-2022	09:06:26	10:25:13	2,292	78:47	2.06
26-09-2022	10:57:28	12:17:47	1,888	80:19	2.55
27-10-2022	10:34:05	14:21:33	804	227:28	16.98

Table B.1: MI Adjustments Planner 24

C Statistical Methods

In this appendix, the statistical methods used in this thesis are briefly explained in alphabetical order.

C.1 Binomial Test

The Binomial Test is used when when dealing with dichotomous outcomes (i.e., success/failure, yes/no, 1/0). The test checks if the observed proportion of successes in a sample matches a pre-specified proportion under the null hypothesis. The binomial test provides a p-value, to verify if the observations deviate significantly from the expected proportion. The test assumes that each trial is independent and has an equal probability of success ("Binomial Test", 2008).

C.2 Breusch-Pagan Test

The Breusch-Pagan Test by Breusch & Pagan (1979) is used to check if the linear regression model is heteroscedastic. The null hypothesis for this test assumes that the error variances are all equal (homoscedasticity), and the alternative hypothesis assumes that the error variances are not equal (heteroscedasticity). A significant test statistic rejects the null hypothesis, indicating the presence of heteroscedasticity in the data.

C.3 Shapiro-Wilk Test

The Shapiro-Wilk Test is a widely used method for testing the normality of a data set (Shapiro & Wilk, 1965). The null hypothesis presumes that the population is normally distributed. The test measures the degree of data deviation from a normal distribution. The test statistic, W, is a ratio of the best estimator of variance to the sample variance. A smaller value of the Shapiro-Wilk test statistic suggests that the sample distribution deviates more from normality. A p-value less than the significance level (usually $p<0.05$) enables us to reject the null hypothesis, implying that the data is not normally distributed.

C.4 Student's t-test

The Student's t-test is a popular statistical test that checks if there are significant differences between the means of two groups. The test relies on the Student's t-distribution, which is shaped according to the degrees of freedom. The standard deviation is derived from the sample data. By comparing the means and variability of two datasets, the t-test enables us to test the null hypothesis that the means of the two groups are equal. A significant result implies a significant difference in the means of the groups (Field, 2009).

C.5 Wilcoxon Signed-Rank Test

The Wilcoxon Signed-Rank Test is a non-parametric statistical method used to compare two paired groups and determines if their population mean ranks differ. The test does not make assumptions regarding the normality of the data and is therefore appropriate for data that does not meet this requirement. The test employs the rank of the absolute differences between pairs, followed by a comparison of the sum of ranks with the expected sum under the null hypothesis. The p-value derived from this test indicates whether the differences between pairs are significantly different from zero (Wilcoxon, 1945).

D Insight in MAE%

Within this section of the appendix, figures for the behavior of the MAE% are shown. Within the figures, we can see that the MAE% still has the potential to show very high value and can be strongly asymmetric. Keep in mind that for readability, a logarithmic scale has been utilized.

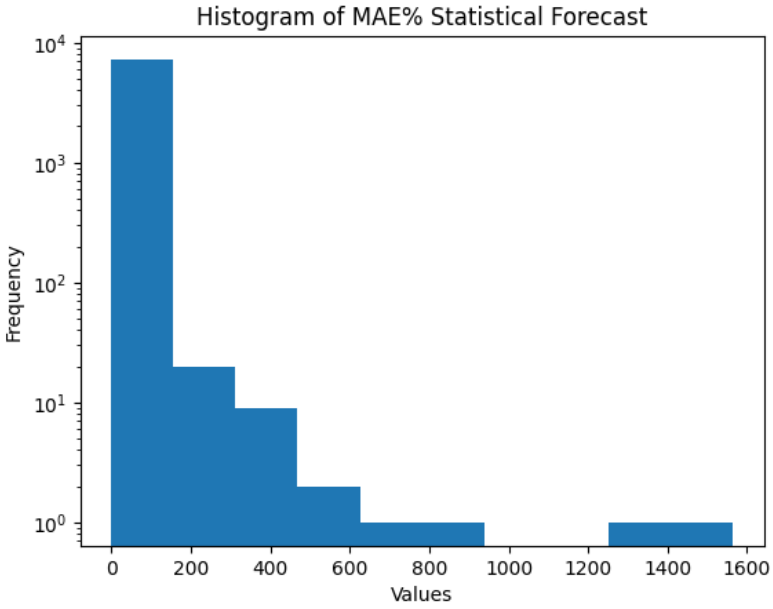


Figure D.1: Histogram of MAE% of Statistical Forecasts

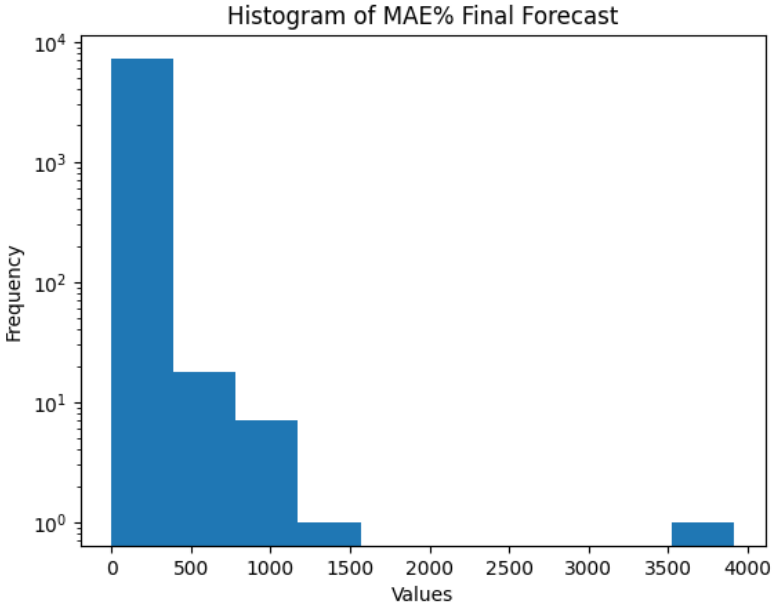


Figure D.2: Histogram of MAE% of Final Forecasts

E Correlations

E.1 Correlations towards FVA

Figures in this section show the correlations between the independent variables and the FVA.

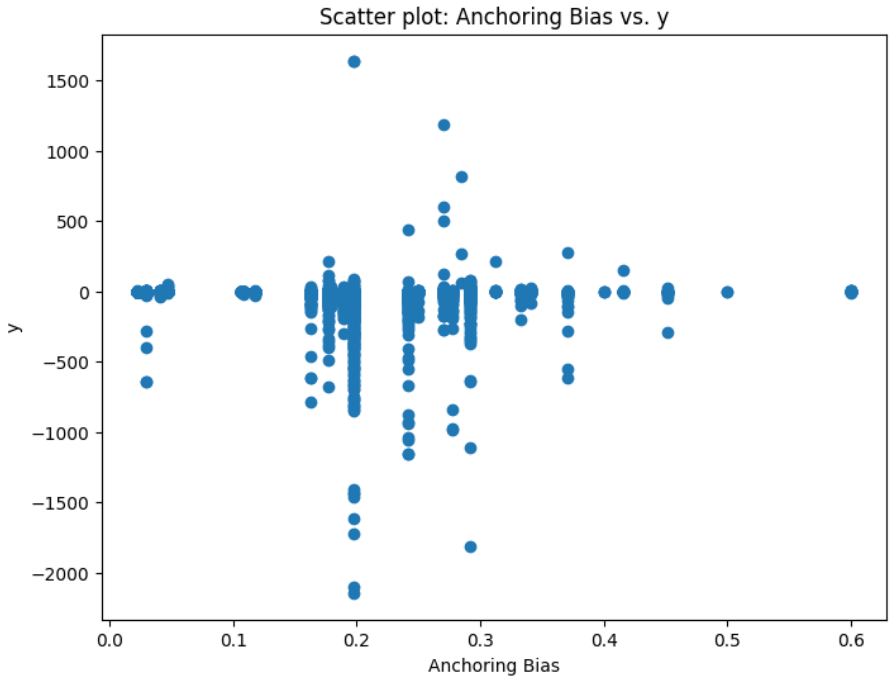


Figure E.1: Correlation - Anchoring Bias - FVA

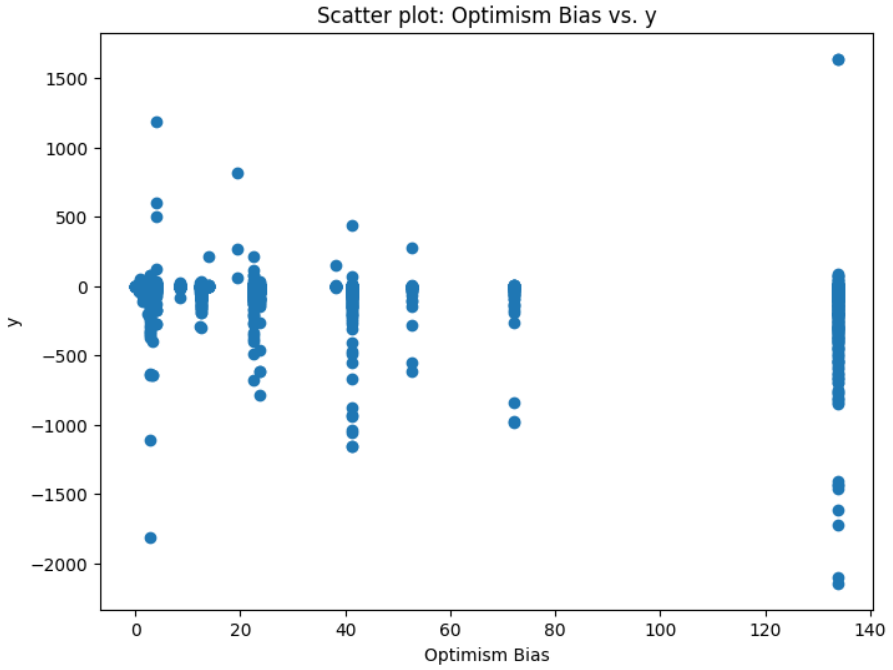


Figure E.2: Correlation - Optimism Bias - FVA

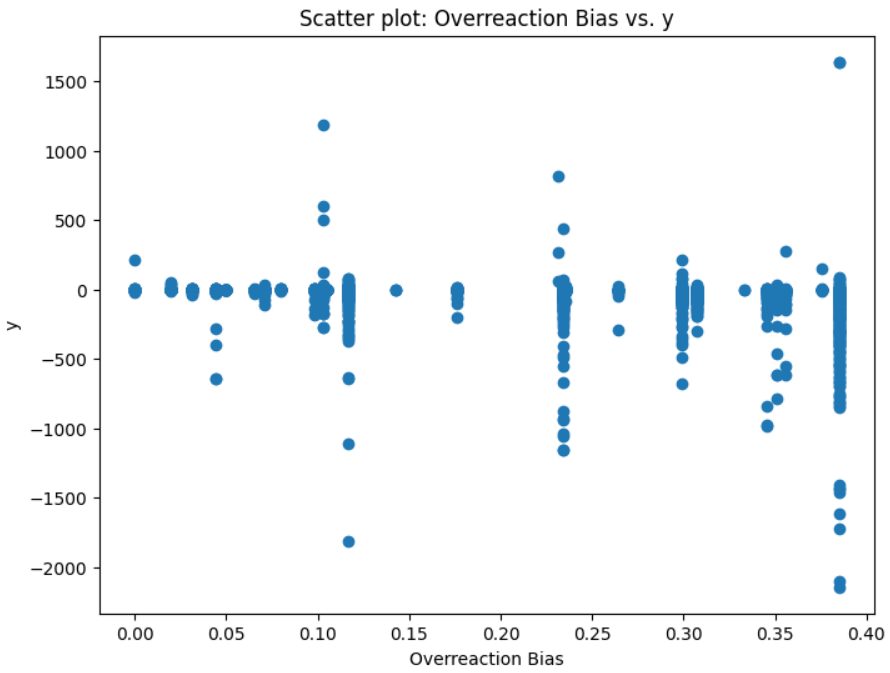


Figure E.3: Correlation - Overreaction Bias - FVA

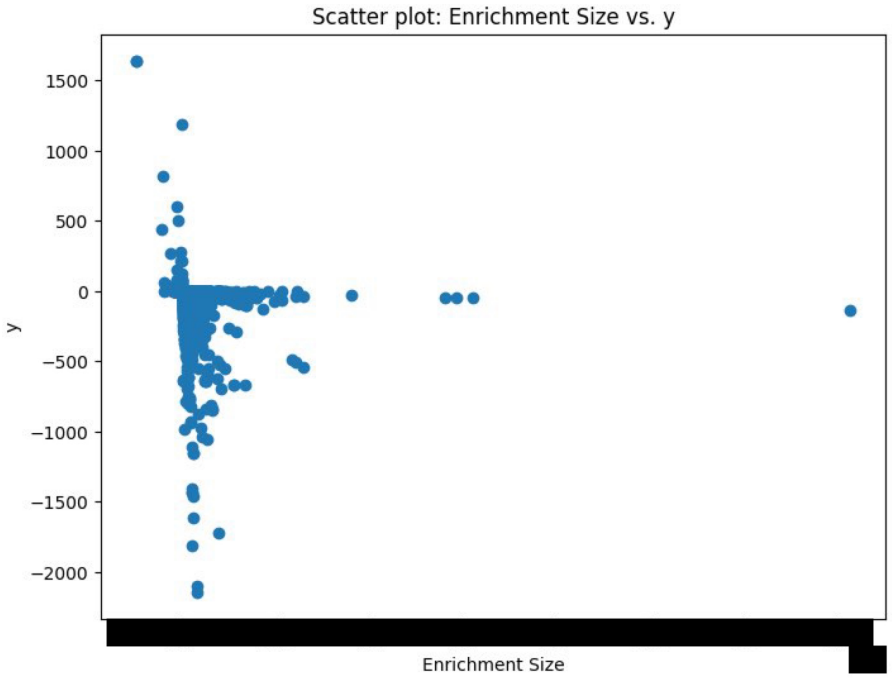


Figure E.4: Correlation - Enrichment Size - FVA

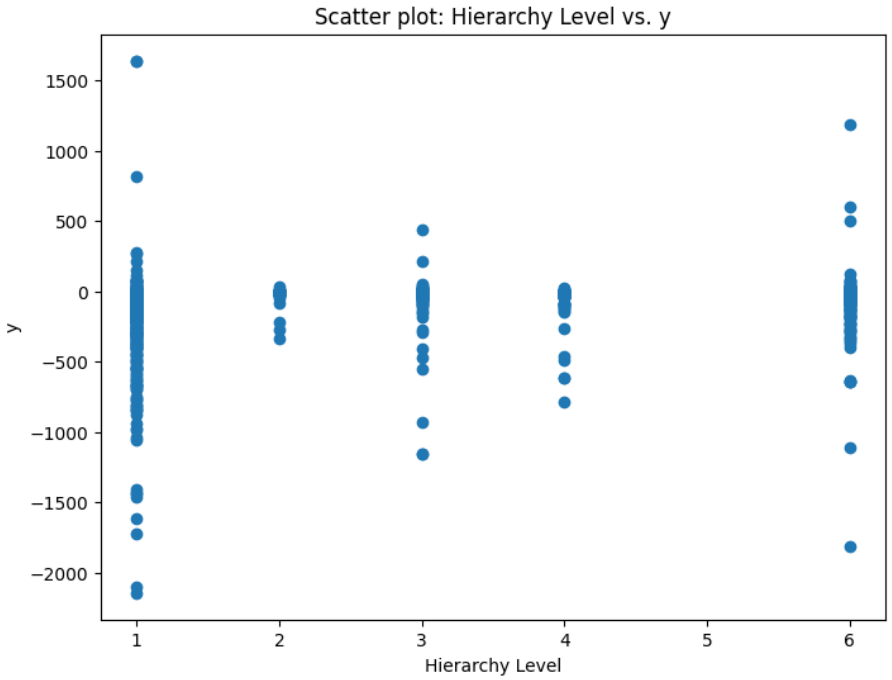


Figure E.5: Correlation - Hierarchy Level - FVA

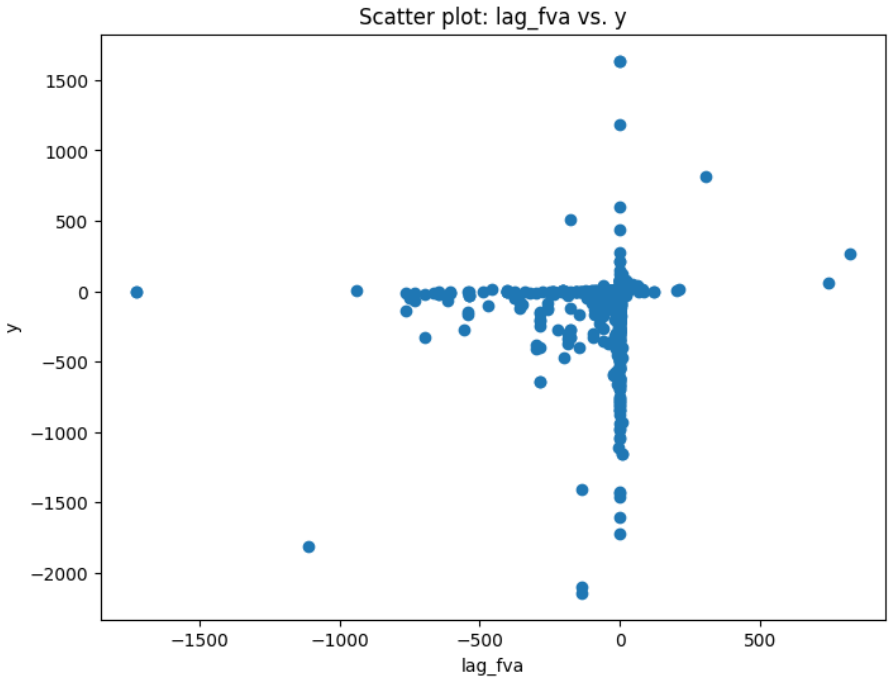


Figure E.6: Correlation - Lagged FVA - FVA

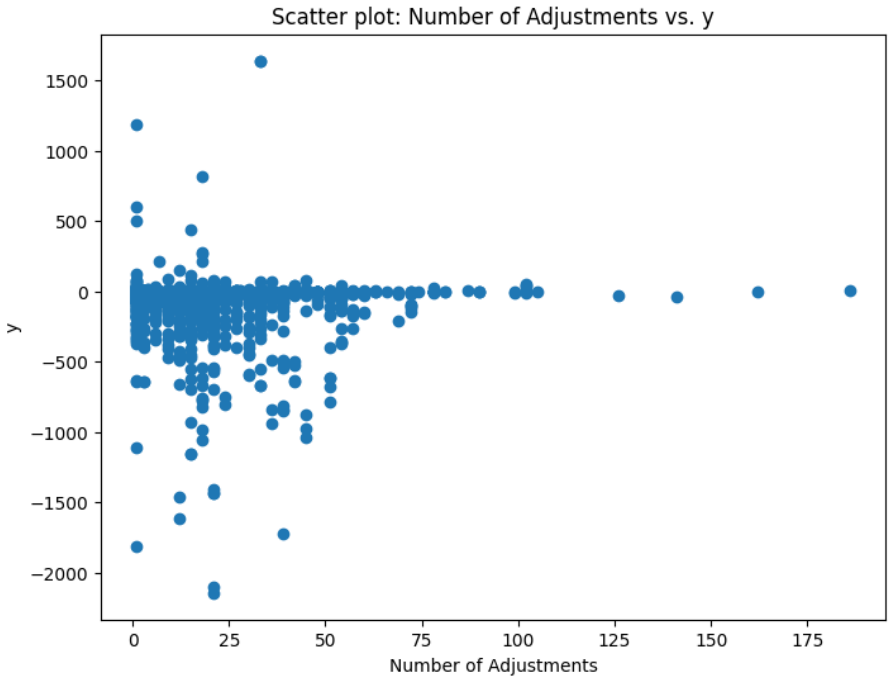


Figure E.7: Correlation - Number of Adjustments - FVA

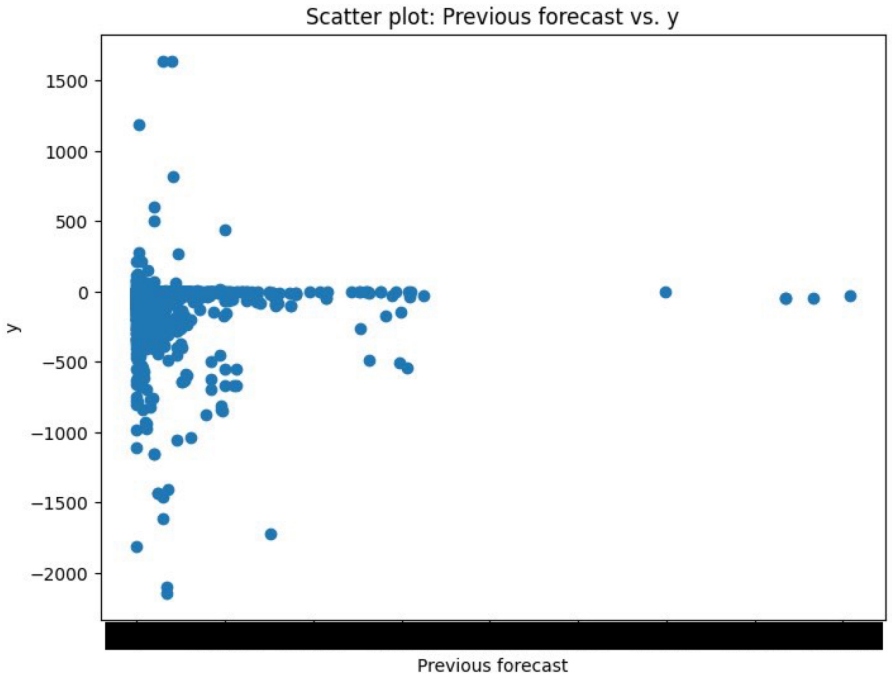


Figure E.8: Correlation - Previous Forecast - FVA

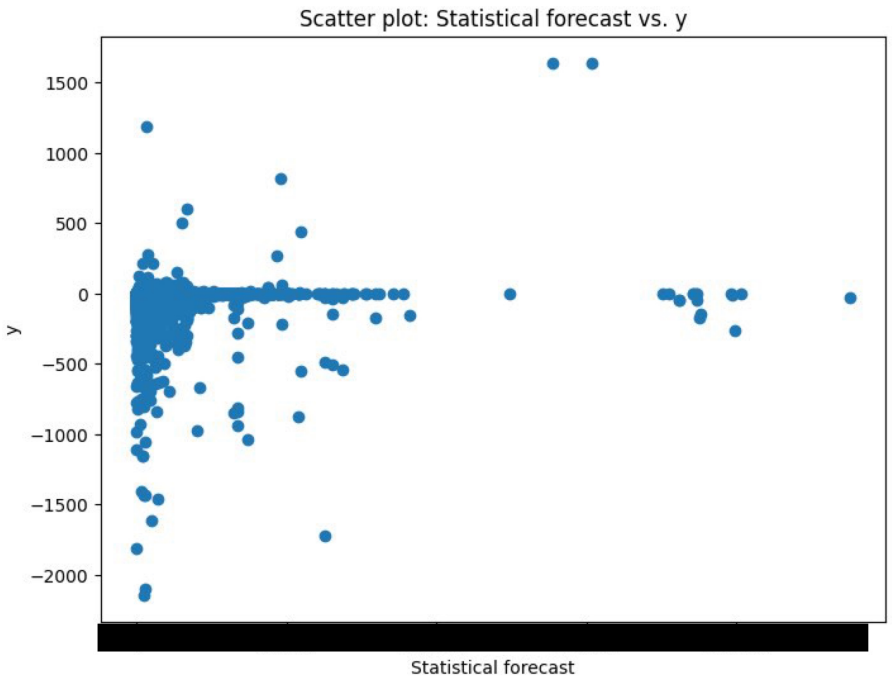


Figure E.9: Correlation - Statistical Forecast - FVA

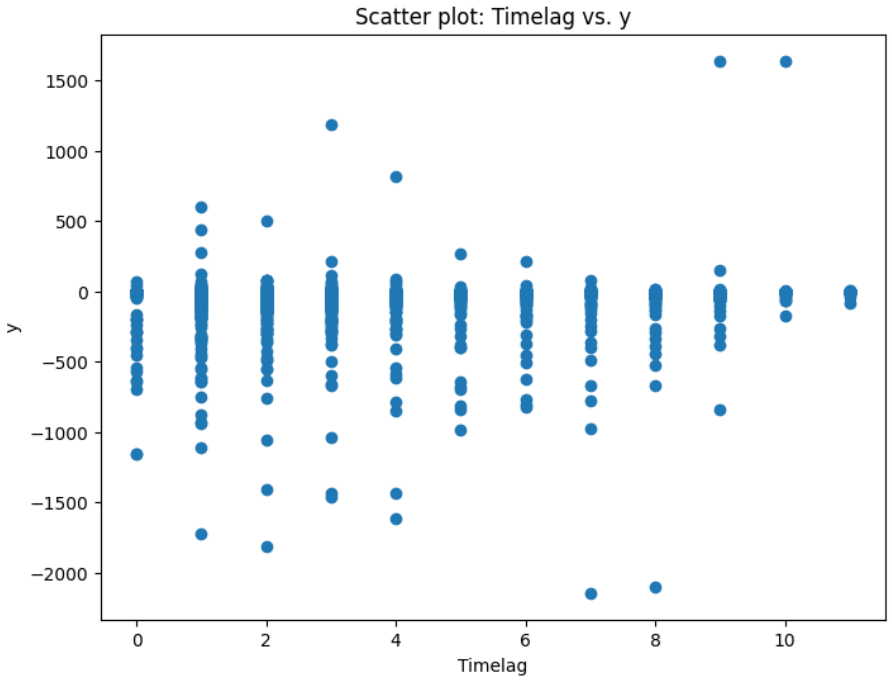


Figure E.10: Correlation - Time-lag - FVA

E.2 Correlations amongst Biases

The figures below show the correlations between the various biases.

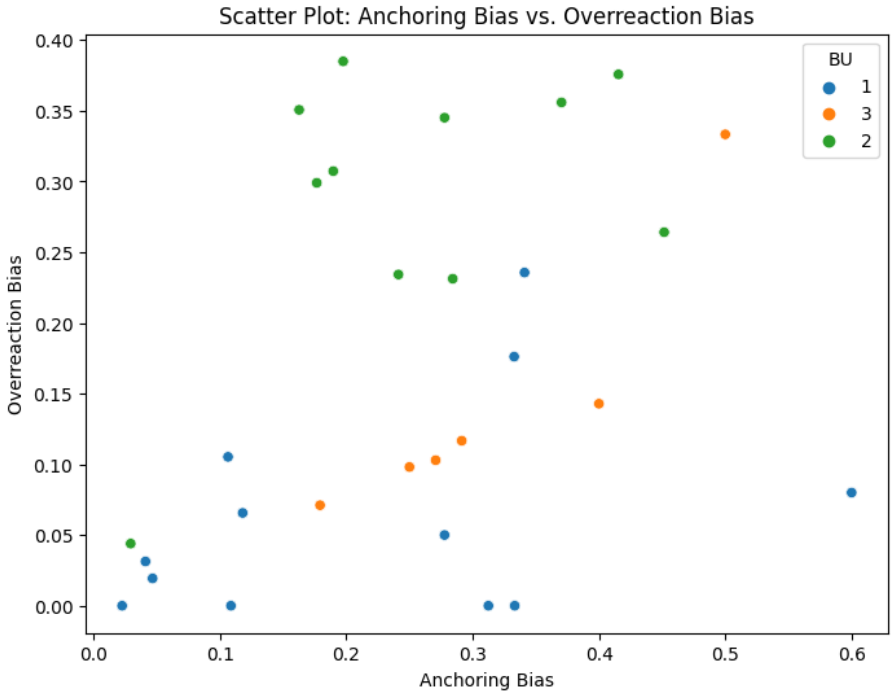


Figure E.11: Correlation - Anchoring Bias - Overreaction Bias

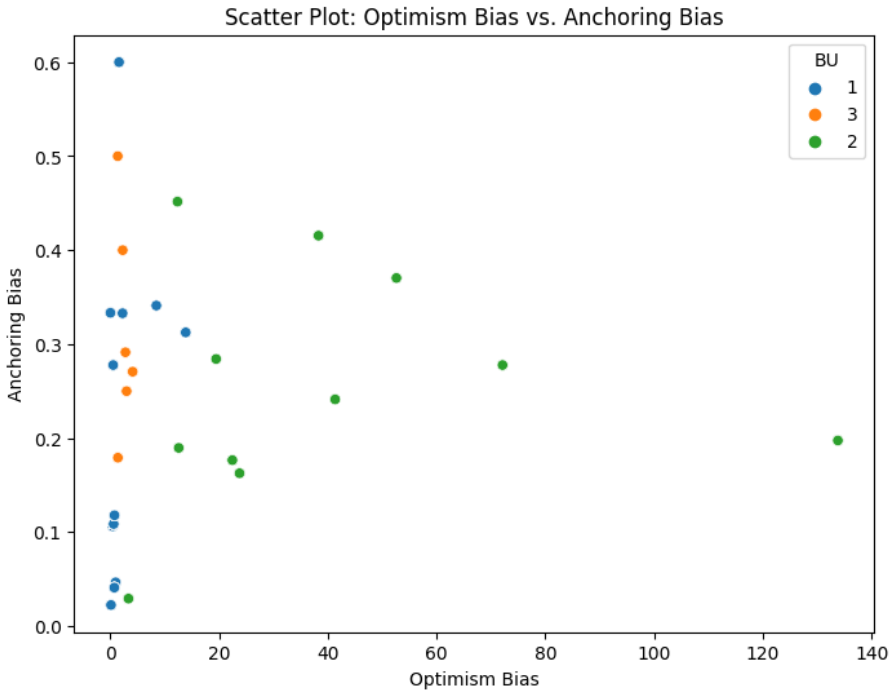


Figure E.12: Correlation - Optimism Bias - Anchoring Bias

E.3 Correlation Table

	Timelag	Statistical forecast	Previous forecast	Hierarchy Level	Number of Adjustments	Optimism Bias	Anchoring Bias	Overreaction Bias	Previous FVA	Enrichment Size	Hour of the Day_sin	Hour of the Day_cos	Day of the Week_sin	Day of the Week_cos	Day of the Month_sin	Day of the Month_cos
Timelag	1.00															
Statistical forecast	not sig	1.00														
Previous forecast	0.03	**0.78**	1.00													
Hierarchy Level	-0.32	-0.08	-0.14	1.00										-		
Number of Adjustments	0.19	0.24	0.25	**-0.65**	1.00											
Optimism Bias	0.11	0.22	0.26	**-0.60**	**0.57**	1.00										
Anchoring Bias	-0.34	0.06	not sig	0.40	-0.21	-0.10	1.00									
Overreaction Bias	0.08	0.12	0.17	**-0.73**	**0.58**	**0.68**	0.13	1.00								
Previous FVA	not sig	-0.08	-0.13	0.10	-0.13	-0.18	0.02	-0.12	1.00							
Enrichment Size	0.06	0.33	**0.62**	-0.15	0.19	0.22	-0.03	0.16	-0.23	1.00						
Hour of the Day_sin	-0.03	0.04	0.03	0.05	not sig	0.12	0.28	0.09	-0.02	0.02	1.00					
Hour of the Day_cos	0.22	0.01	0.05	-0.44	0.27	0.27	-0.35	0.24	-0.06	0.05	0.14	1.00				
Day of the Week_sin	0.12	-0.02	-0.02	-0.15	0.04	0.06	-0.07	0.08	not sig	not sig	-0.2	-0.02	1.00			
Day of the Week_cos	-0.02	0.02	0.01	-0.03	0.04	0.06	0.1	0.10	not sig	not sig	0.07	0.13	0.13	1.00		
Day of the Month_sin	0.18	-0.02	not sig	-0.09	-0.02	-0.04	-0.31	-0.15	not sig	0.02	0.11	0.22	0.24	0.16	1.00	
Day of the Month_cos	-0.19	-0.05	-0.07	0.19	-0.2	-0.14	0.32	-0.05	0.04	-0.08	not sig	-0.13	0.1	0.18	-0.07	1.00

Table E.1: Correlation table

F Linear Regression

In this section of the appendix, the assumptions of linear regression are explained and tested. The assumptions are based on the information of Hair et al. (2019), who explains the assumptions which will be explored.

1. *Linearity:*
   Given the name of the model is 'Linear Regression' it makes sense that the linearity assumption should be met. The assumption indicates that the dependent and independent variables are linked through a linear relationship. In order to test this for multiple linear regression, one should plot the residuals and predicted value for y. The assumption is met once the residuals are located randomly along the x-axis. Within Appendix E we can see that none of the figures show a particularly strong linear relationship between the independent and dependent variables. To further check if there is linearity, the 'rainbow test' is executed (Utes, 1982). This test is executed through the statsmodel python package *statsmodels.stats.diagnostic.linear_rainbow* (2023). The statistic reports a value of 1.134 with a p-value of 0.001. Thus, the linearity assumption is violated.

2. *Normality:*
   Ideally, the residuals should follow a normal distribution, centered at 0. This property should guarantee the validity of statistical tests and correct parameter estimation. The estimation techniques like Ordinary Least Squares (OLS) or Maximum Likelihood Estimation (MLE) require that the residuals are normally distributed. To assess the assumption, you can make a p-p or a q-q plot. If the assumption is satisfied, the data points should closely follow this linear relationship. In Figure F.1, we have plotted the residuals and we can see that it does not follow the normal distribution closely. The left tail of the distribution is larger than the right tail. We can also see very high peaks at the center and some 'lumps' in the left tail as well. When investigating further with a Q-Q plot, we would like to see the dots closely follow the plotted red line. In Figure F.2, we can see that this is also not the case. The residuals deviate significantly from the desired behavior. One can also execute a Shapiro-Wilk Test ((Shapiro & Wilk, 1965)) to check numerically if the residuals follow a normal distribution. For this data, the test statistic is 0.832 with a p-value of 0.000. This further substantiates that the residuals are not normally distributed and thus, the assumption of normality is violated.

3. *Homoscedasticity:*
   Homoscedasticity means that the errors are spread equally across the values. When X either increases or decreases, the variance stays equal. However, when the errors do vary in size depending on their location, the data can be considered to be heteroscedastic. Heteroscedasticity could be caused by either the variables selected, the presence of outliers, omitted important variables, or incorrect data transformation. To identify if your data is heteroscedastic, one can easily plot residual plots. If the data is considered to be heteroscedastic, the plot should produce a distinctive fan or cone shape. Furthermore, you can also use statistical tests like the Breusch-Pagan test (Breusch & Pagan, 1979). In Figure F.3, the dots should be randomly scattered around an imaginary line of y=0. However, this is not the case and this shows

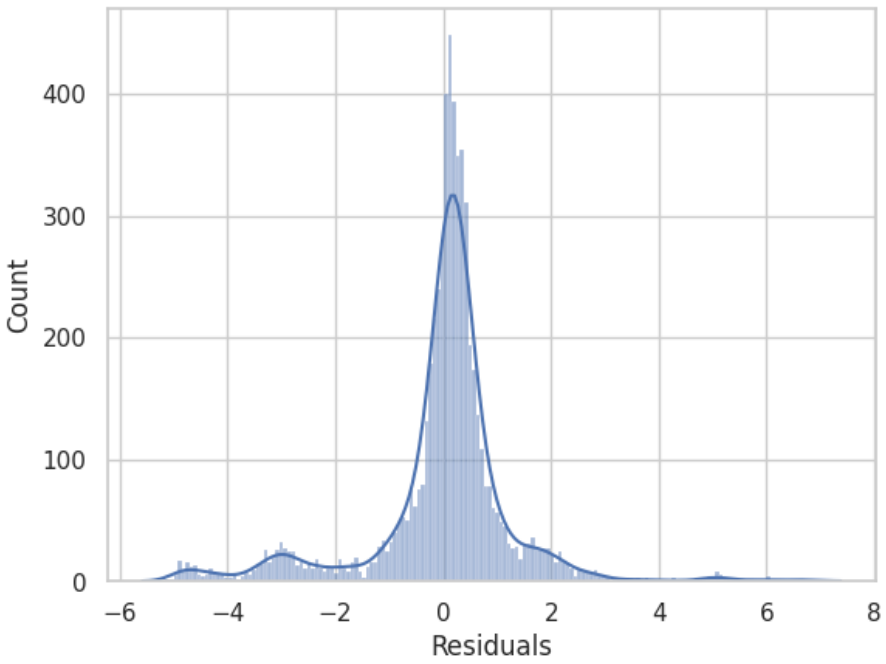


Figure F.1: LR - Normality Assumption - Residuals Plot

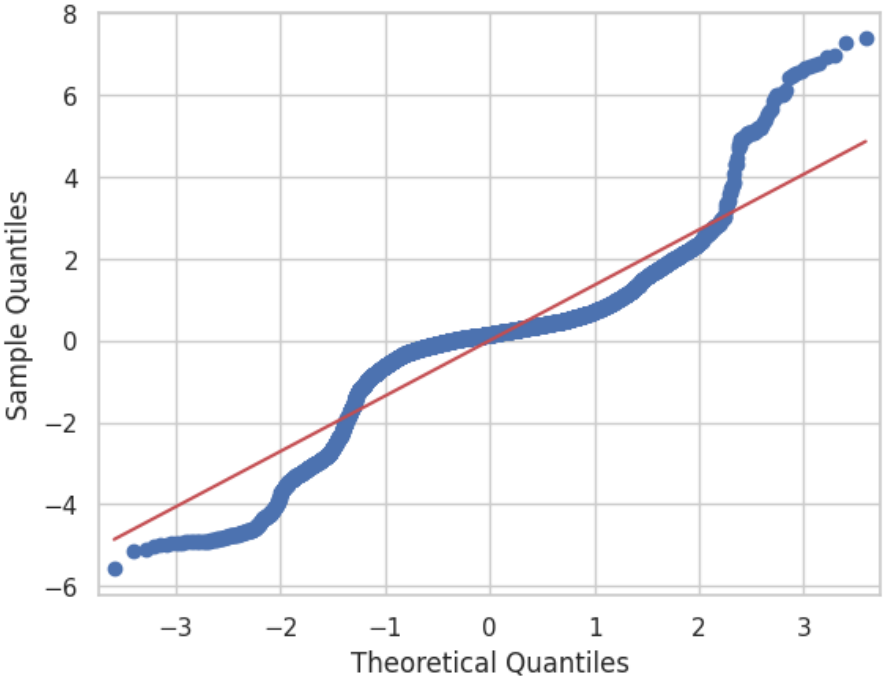


Figure F.2: LR - Normality Assumption - QQ Plot

that the data is also strongly heteroscedastic. There is also a numerical test to identify the level of heteroscedasticity. This is the so-called Breusch-Pagan test (Breusch & Pagan, 1979) as described in Section 5.3.1. Table F.1 shows the calculated values for the statistics. Based on the high Lagrange and f-values, combined with the extremely low p-values, the table shows strong evidence that the null hypothesis can be rejected and in turn, that heteroscedasticity is clearly present in the model.

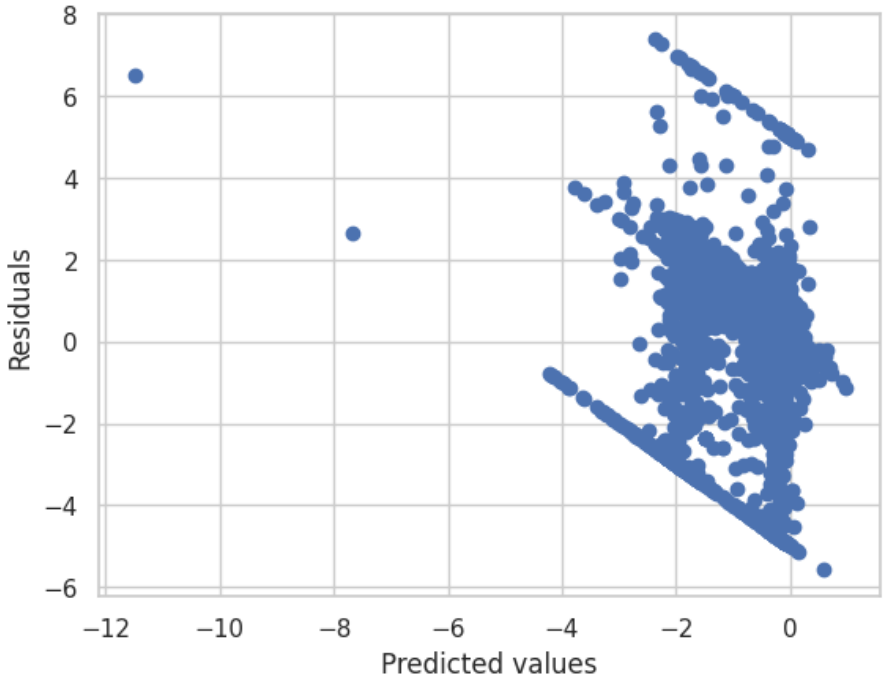


Figure F.3: LR - Homoscedasticity Assumption

Statistic	**Value**
Lagrange multiplier statistic	894.43
p-value	$2.83 * 10^{-170}$
f-value	40.18
f p-value	$8.09 * 10^{-187}$

Table F.1: Breusch-Pagan Test

4. *Autocorrelation/Exogeneity :*
The last assumption states that error terms are independent within linear regression. In other words, the error term of a certain observation does not influence the error term of another observation. The presence of autocorrelation leads to an underestimated true standard error. This can lead to a lower p-value and state that certain variables are significant even when they are not. To test this assumption, the Durbin-Watson statistic can be utilized for the presence of autocorrelation.

$$d = \frac{\sum_{t=2}^{T} (e_t - e_{t-1})^2}{\sum_{t=1}^{T} e_t^2} \tag{F.1}$$

where e_t represents the residuals at time t, and T is the number of observations. The

statistic ranges from 0 to 4, with values closer to 0 indicating positive autocorrelation while values closer to 4 indicate negative autocorrelation. Values around 2 are desired since they suggest that there is no clear autocorrelation. For the linear model, the Durbin-Watson statistic is 1.9845, which indicates that there is no significant autocorrelation. Thus, this assumption is met.

Ideally, we would also like no multicollinearity between variables. Even though it is often referred to as an assumption, it is not. Multicollinearity is the case when several explanatory variables are very closely related to one another. Thus, this makes it difficult for the results of a single variable to be identified. One can check for multicollinearity by using both a scatter plot and a Variance Inflation Factor (VIF) (Formula F.2) . If the scatter plot shows a strong linear correlation between two different variables, we can infer that multicollinearity is taking place. Since we have a highly dimensional dataset, we are not plotting a scatterplot but we aim purely at the VIF , which is a statistical method to test the multicollinearity. A higher value of VIF, indicates a larger extent of collinearity, starting from a VIF of 1 (no multicollinearity). If the VIF is higher than 2, one should consider respecifying the model.

$$\text{VIF}(X_i) = \frac{1}{1 - R_i^2} \tag{F.2}$$

Feature	**VIF**
Timelag	1.32
Statistical Forecast	2.78
Previous Forecast	3.68
Hierarchy Level	7.01
Number of Adjustments	2.05
Optimism Bias	2.32
Anchoring Bias	4.60
Overreaction Bias	6.99
Previous FVA	1.09
Enrichment Size	1.82
Hour of the Day (sin)	1.51
Hour of the Day (cos)	1.56
Day of the Week (sin)	1.23
Day of the Week (cos)	1.14
Day of the Month (sin)	1.48
Day of the Month (cos)	1.22

Table F.2: Multicollinearity Test

Values closer to 1 indicate that there is little or no multicollinearity, while Table F.2 presents the results for each predictor. They can roughly be divided into three different groups.

The features of Timelag and Lagged FVA have very low values for their VIF which indicates almost no multicollinearity. There is another group with Hierarchy, Anchoring Bias, and Overreaction which have strong correlations which indicate that there could be multicollinearity. The remaining features suggest low levels of multicollinearity. The categorical features have been removed from this statistic since it cannot be calculated.

G Performance Table

#	Model	BU	Scaled	Clip	FE	# Feat	Training MAE	Testing MAE	Training Bias	Testing Bias	Runtime (hh:mm:ss)
1	LR	All	No	No	No	29	8.922	8.716	0.000	0.166	00:00:00
2	LR	1 & 3	No	No	No	28	2.017	2.215	0.000	-0.068	00:00:00
3	LR	2	No	No	No	26	34.282	35.008	0.000	0.137	00:00:00
4	LR	All	Yes	No	No	29	8.923	8.717	0.000	0.166	00:00:00
5	LR	1 & 3	Yes	No	No	28	2.017	2.215	0.000	-0.068	00:00:00
6	LR	2	Yes	No	No	26	34.282	35.008	0.000	0.137	00:00:00
7	LR	All	Yes	Yes	No	29	0.801	0.812	-0.000	0.003	00:00:00
8	LR	1 & 3	Yes	Yes	No	28	0.556	0.553	0.000	0.083	00:00:00
9	LR	2	Yes	Yes	No	26	1.772	1.838	0.000	-0.051	00:00:00
10	LGBM	All	No	No	No	29	6.490	7.704	-0.013	0.156	01:42:46
11	LGBM	1 & 3	No	No	No	28	1.027	1.560	-0.037	0.267	01:21:56
12	LGBM	2	No	No	No	26	24.235	26.406	0.004	-0.159	00:44:08
13	LGBM	All	Yes	No	No	29	6.432	7.612	-0.011	0.115	01:51:20
14	LGBM	1 & 3	Yes	No	No	28	1.015	1.535	-0.031	0.334	01:23:17
15	LGBM	2	Yes	No	No	26	23.943	26.656	-0.000	-0.184	00:49:34
16	LGBM	All	Yes	Yes	No	29	0.494	0.669	-0.001	-0.024	03:09:10
17	LGBM	1 & 3	Yes	Yes	No	28	0.489	0.507	0.018	-0.033	02:07:26
18	LGBM	2	Yes	Yes	No	26	1.031	1.064	-0.001	0.000	01:04:29
19	LGBM	All	Yes	Yes	1	35	0.491	0.594	-0.172	-0.201	02:56:44
20	LGBM	1 & 3	Yes	Yes	1	33	0.334	0.435	-0.695	-0.697	04:02:14
21	LGBM	2	Yes	Yes	1	39	0.608	0.954	0.027	0.046	01:33:12
22	LGBM	All	Yes	Yes	2	104	0.436	0.592	-0.166	-0.188	06:03:11
23	LGBM	1 & 3	Yes	Yes	2	102	0.324	0.437	-0.670	-0.694	04:52:25
24	LGBM	2	Yes	Yes	2	84	0.641	0.959	0.036	0.067	02:22:24
25	NN	All	No	No	No	29	5.213	6.941	-1.000	-1.000	14:15:53
26	NN	1 & 3	No	No	No	27	1.308	1.420	-1.001	-1.001	10:52:18
27	NN	2	No	No	No	26	22.665	23.602	-0.776	-0.782	04:10:59
28	NN	All	Yes	No	No	29	6.167	4.833	-0.717	-0.715	14:08:08
29	NN	1 & 3	Yes	No	No	27	1.123	1.285	-0.682	-0.794	10:52:31
30	NN	2	Yes	No	No	26	19.212	21.844	-0.640	-0.604	03:28:26
31	NN	All	Yes	Yes	No	28	0.627	0.649	-0.272	-0.294	14:16:02
32	NN	1 & 3	Yes	Yes	No	27	0.474	0.495	-0.945	-0.953	10:42:31
33	NN	2	Yes	Yes	No	26	0.955	1.242	-0.027	-0.027	03:31:58
34	NN	All	Yes	Yes	1	37	0.622	0.649	-0.376	-0.390	13:56:15
35	NN	1 & 3	Yes	Yes	1	32	0.472	0.497	-0.936	-0.930	10:45:24
36	NN	2	Yes	Yes	1	36	0.968	1.184	-0.049	-0.060	03:33:02
37	NN	All	Yes	Yes	2	112	0.599	0.652	-0.283	-0.290	14:32:10
38	NN	1 & 3	Yes	Yes	2	97	0.456	0.486	-0.767	-0.759	10:52:21
39	NN	2	Yes	Yes	2	77	0.744	1.162	0.014	0.027	03:33:29

Table G.1: Results Prediction Models