



Degree Project in Mathematical Statistics

Second cycle, 30 credits

Predicting the Impact of Supply Chain Disruptions Using Statistical Analysis and Machine Learning

**HANNES ANDERSSON
JOHN SJÖBERG**

Abstract

The dairy business is vulnerable to supply chain disruptions since large safety stocks to cover up losses are not always a viable option, therefore it is crucial to maintain a smooth supply chain to ensure stable delivery accuracies. Disruptions are unpredictable and hard to avoid in the supply chain, especially in cases where production errors cause lost production volume. This thesis proposes the use of machine learning and statistical modelling together with data from Arla to predict when a shortage will occur and its duration to allow proactive decision making to mitigate the consequences of the disruption. The aim of this thesis is to create one predictive model for delay and one for duration based on data from multiple products and explore how the features and methods used can capture the product specific characteristics in the data and thereupon improve the models. The model used for evaluating these factors was a random forest classifier, and permutation feature importance was used to determine the relevant features for the models. The issue of having imbalanced data was handled by first grouping the data and then applying the oversampling method SMOTE. The two models were trained on different datasets where the duration model was trained on all disruptions and the delay model was only trained on a subset where a shortage have occurred. One finding was that applying SMOTE yielded the best results. The best duration model had an accuracy of 62% with a precision and recall of 79% and 76% respectively for the majority class, but very low for the other classes with a combined average of 21% and 24%. The most important feature for the duration was the the quotient describing the lost production. The best delay model had an accuracy of 62% with more accurate predictions over all classes and an average precision and recall of 59% and 57%. The most important feature for the delay was how often a product is produced.

Sammanfattning

Prediktering av följderna från störningar i en försörjningskedja med användning av statistisk analys och maskininlärning

Mejeribranschen är sårbar för störningar i försörjningskedjan eftersom stora säkerhetslager för att täcka förluster inte alltid är ett genomförbart alternativ, därför är det avgörande att upprätthålla en smidig försörjningskedja för att säkerställa stabila leveransnivåer. Störningar är oförutsägbara och svåra att undvika i en försörjningskedja, särskilt i de fall där produktionsfel orsakar minskad produktionsvolym. Denna uppsats föreslår användning av maskininlärning och statistisk modellering tillsammans med data från Arla för att prediktera när en brist kommer att uppstå i förhållande till störningen samt bristens varaktighet för att möjliggöra proaktiva beslut som förmildrar konsekvenserna av störningen. Målet med denna uppsats är att skapa en prediktiv modell för fördröjning och en för varaktighet baserad på data från flera produkter och undersöka hur de variabler och metoder som användes kan fånga produktspecifika egenskaper i data och därav förbättra modellen. Modellen som användes för att utvärdera dessa faktorer var en random forest klassificerare, och permutation feature importance användes för att utvärdera de använda variablerna för modellerna. Obalanserad data hanterades genom att först gruppera datan och sedan tillämpa översamlingsmetoden SMOTE. De två modellerna tränades på olika data där varaktighetsmodellen tränades på alla störningar och fördröjningsmodellen endast tränades på de fall där en brist uppstått. En slutsats var att tillämpning av SMOTE gav de bästa resultaten. Den bästa varaktighetsmodellen hade en noggrannhet på 62% med hög precision och recall på 79% respektive 76% för majoritetsklassen men mycket låg för de andra klasserna med en genomsnittlig precision och recall på 21% och 24%. Den viktigaste variabeln för varaktigheten var kvoten som beskriver den förlorade produktionen. Den bästa fördröjningsmodellen hade en noggrannhet på 62% med stabilare prediktioner över alla klasser och en genomsnittlig precision och recall på 59% och 57%. Den viktigaste variabeln för fördröjningen var hur ofta en produkt produceras.

Acknowledgements

We would like to express our sincere gratitude to our supervisor at KTH, Jan Sand and our supervisor at Arla, Teddy Edlund. Jan provided us with insightful feedback and constructive criticism that helped us improve the quality of our work. Teddy offered his expertise and support, and provided us with access to valuable resources and data. We would also like to send a special thanks to Thomas, Anmol, Fredrik and Johan at Arla for their help and support throughout the thesis.

Stockholm, May 2023

Hannes Andersson
John Sjöberg

Acronyms

Arla	Arla Foods AB
CV	Cross Validation
DA	Delivery Accuracy
IQR	Interquartile Range
SMOTE	Synthetic Minority Oversampling Technique
<i>k</i>NN	<i>k</i> -Nearest Neighbour
PFI	Permutation Feature Importance

Contents

1	Introduction	1
1.1	Background	1
1.2	Problem	2
1.3	Explanation of Key Concepts	2
1.4	Purpose	3
1.5	Goal	3
1.6	Research Questions	3
1.7	Methodology	3
1.8	Delimitations	4
1.9	Outline	4
2	Literature Review	5
3	Theory	6
3.1	Supervised Machine Learning	6
3.2	Decision Trees and Random Forests	6
3.2.1	Decision Tree	7
3.2.2	Random Forest	8
3.3	Model Evaluation	8
3.3.1	Confusion Matrix	8
3.3.2	Cross-Validation	9
3.3.3	Boxplot	9
3.3.4	Feature Importance	10
3.4	SMOTE	11
3.5	Feature Engineering	11
3.5.1	Feature Creation	11
3.5.2	Normalization	11
3.5.3	Pearson Correlation Coefficient	12
4	Method	13
4.1	Defining a Disruption	13
4.2	Procurement of Data	13
4.2.1	Sales and Production Data	14
4.2.2	Defining Shortage	14
4.2.3	Shortage Mapping	16
4.2.4	Additional Features	18
4.3	Feature Engineering	19
4.3.1	Feature Creation	19

4.3.2	Feature transformation	20
4.3.3	Final features	20
4.4	Data Preprocessing	21
4.5	Implementation	23
4.5.1	Treatment of Imbalanced Data	23
4.5.2	Feature Selection	24
4.5.3	Models	25
5	Results and Analysis	26
5.1	Duration Model	26
5.1.1	Oversampling	26
5.1.2	Results for Feature Setups	27
5.1.3	Analysis of Duration Models	30
5.2	Delay Model	30
5.2.1	Oversampling	30
5.2.2	Results for Feature Setups	31
5.2.3	Analysis of Delay Models	33
5.3	Feature Importance	34
5.3.1	Duration Model	34
5.3.2	Delay Model	35
6	Discussion and Conclusions	36
6.1	Duration Model	36
6.2	Delay Model	37
6.3	Future Work	38
6.4	Conclusions	39
	References	40

Chapter 1

Introduction

This master's thesis was written at KTH and carried out in collaboration with Arla Foods AB (Arla). Arla is a global dairy cooperative. Chapter 1 contains information about the background, purpose, problem formulation etc. and aims to give the reader an extensive introduction to the thesis.

1.1 Background

Arla is a global dairy cooperative producing more than 6.8 billion kilos of varying dairy products from its 60 facilities which are sold in 144 countries world wide [2]. In the dairy business, it is generally difficult to compensate inaccurate demand predictions or supply chain disruptions with a large safety stock in order to ensure great delivery security due to the expiration date of the products. Therefore, it is imperative that the supply chain runs as smooth as possible to maintain stable delivery rates, but when a disruption occurs it is not uncommon to be followed by a period where the demand is no longer met, a shortage. Not being able to deliver the ordered amount leads to several unwanted consequences, one being empty shelves at retail grocery stores, but with insight of when a shortage will occur and how long of a period it will last, it is possible to take action and prevent some of them.

One particular disruption in the supply chain is when a production error has occurred and the produced volume does not meet the desired production volume. Knowledge of when a production error presents itself and the magnitude of the lost production is immediate, but the potential ramifications are not. This poses an opportunity to get ahead of a period of shortage, and if it is possible to predict when it will occur and for how long it lasts, proactive actions can be taken to decrease its impact. One way to tackle this prediction is using machine learning and statistical modelling with the usage of company data, which will be the main focus of this thesis.

Arla strives to maintain a high delivery accuracy but in the event of unpredictable disruptions in the supply chain, the existing safety stock might not always have the capability to compensate for the missing products and therefore, Arla is not able to deliver full capacity to the customers. In the worst case, they are not able to deliver any goods at all and some customers might end up with empty shelves in their stores. If customers are left with empty shelves, then the consumers in the store might purchase products from competitive brands which can result in customer churn.

1.2 Problem

The supply chain for a producing company as Arla is very complex and disruptions can be caused by various events. One way of investigating the impact of a disruption is to make a qualitative analysis of the supply chain and for example investigate how decisions are made and how disruptions in different parts of the supply chain have different impacts. The result of this qualitative analysis can be used to build a framework on how to act in the event of a disruption. However, this approach can be time consuming and investigating each disruption in a qualitative manner may not be completed before the impact can be seen in the sales. Therefore, a data driven approach which can give statistically based predictions of the impact instantly in the event of a disruption can be used to take proactive actions which reduces the impact. There are few examples of data driven approaches used in a supply chain to predict this kind of events and this thesis will investigate if such an approach can be implemented to improve the decision making at Arla.

An underlying assumption caused by the fact that individual products have few usable data points is that the link between a disruption and a shortage is the same for all products and can therefore be combined when training a predictive model. A problem that occurs when combining the data is that each product has different behaviours regarding sales and production.

1.3 Explanation of Key Concepts

In this section, explanations of some of the most important key concepts which are unique for this thesis are presented. These are only brief explanations for an initial understanding and the exact definitions of these concepts will be described in detail later in Chapter 4.

- **Disruption:** A disruption is defined as a disturbance that occurred during production which caused the production to not being able to produce the planned amount of volume. No investigation of the cause of disruption is made, thus, when referring to a disruption it is a production instance which had a loss in production volume.
- **Shortage:** A shortage is a period of time where the demand for a product cannot be met by the supply. In this thesis, this is when the ordered volume is larger than the actual delivered volume. The length of a shortage, also referred to as the duration of the shortage is the number of consecutive days where the ordered volume is not met by the delivered volume.
- **Impact of a disruption:** The only impact of a disruption investigated in this thesis is possible shortages. The impact of a disruption is twofold and defined separately. The first is the number of days from the disruption until the start of the shortage, referred to as the delay. The second one is the duration of the corresponding shortage. A disruption always has a measured impact so if it does not lead to any shortage, no duration or delay exist and they are set to zero.

1.4 Purpose

The purpose of this thesis is to improve the decision making at Arla by using a data driven approach which can be used as support when taking actions to reduce the impact of a disruption. The purpose is also to give insight in how combined data can be handled to remove product specificity.

1.5 Goal

The goal of the thesis is to create a reliable machine learning model that can predict the impact of a disruption. To achieve this, the goal has been divided into three subgoals. The first subgoal is to remove product specific characteristics of the data using feature engineering and re-scaling methods. The second subgoal is to find a method that satisfies the traits that Arla consider important for their usage of the model. The third and last subgoal is to test the model with different measures of various features and explain which features that gives the most reliable model.

1.6 Research Questions

Based on the goals and purpose of the thesis, the research questions are the following:

- Can machine learning be used to create a reliable data driven approach to predict the impact following a disruption?
- Which of the explored features are the most relevant when predicting the impact following a disruption?

1.7 Methodology

Since there is limited research in this area, the methodology is mainly based on intuition and conclusions drawn from knowledge of Arla's business model. No model has previously been made to predict the impact of a disruption so the focus of this thesis is on the modeling part. Disruptions are underrepresented in all production instances and therefore, to obtain enough data for a model, a generalized model for all different products is intended to be done. The underlying assumption is that all products have the same behaviour when it comes to the link between the disruption and impact.

Each row of data corresponds to a disruption where its features are based on information about the disruption and its specific product, the target variables of the data are the impact, i.e. the duration and the delay to that corresponding disruption.

There is no preexisting mapping between a disruption and its impact so the first problem to handle is to map a disruption to a shortage which is based on logical assumptions. When this is done, the next step is to create the data used in the model. The delay and duration of a shortage is believed to depend on the demand of a product which in turn is highly dependent on seasonality parameters such as the time of the year, if it is a certain holiday etc. This thesis will use the existing data for daily sales forecasting of

the demand from Arla and assume that the forecasting captures these product specific seasonal behaviours. The seasonality features used to capture the behaviour of a product are therefore neglected and replaced with the demand forecast instead. The thesis will investigate how additional features such as the forecast can be used in different ways to best capture the behaviour and give a more accurate model. Each row of data is still product specific and therefore different methods are used to normalize and describe the features as quotients which are assumed to be general for all products.

1.8 Delimitations

In order to narrow the scope of the thesis the following delimitations were applied:

- Details of a disruption in terms of where in the supply chain and what type of disruption it is etc. will not be investigated.
- The sales, delivery and production data are historical and gathered from January 2022 until April 2023.
- The products used in the scope are the 285 ones produced in Linköping factory since the products in the same production site are assumed to behave similarly in terms of external factors such as delivery times etc.

1.9 Outline

Chapter 2 presents relevant background literature on how machine learning methods have been used to predict the time to an event and possible issues regarding those methods. Chapter 3 presents the relevant theory in order to understand the thesis. Chapter 4 presents the methods used to solve the problem of the thesis. Chapter 5 presents the results obtained from the analysis and evaluation of the model. In Chapter 6 the general findings are discussed, conclusions of the findings are drawn and possible future work is presented.

Chapter 2

Literature Review

In the literature research, no study regarding the application of machine learning methods to predict the time until a shortage and the duration of a shortage following a disruption in the production was found. Thus, in order to find a suitable methodology of the thesis, studies regarding the prediction of time based on an event were reviewed.

Using machine learning models to predict the time of occurrence and the duration of events based on time series data has been explored in several fields using different types of models, both as a regression and as a classification task. One application is predicting days of inpatient length of stay in days based on both categorical and continuous features. Random forests, an extension of decisions trees that are known for their interpretability, has been presented and proved successful in both instances, in classification where days are either grouped up in periods and assigned to a class or each individual day is assigned its own class [1, 15], and in regression with days as its target [11]. However, in a classification task there is a common issue of imbalanced data where the distribution of occurrences of each class in a dataset is unevenly distributed, which can cause many standard machine learning algorithms to fail in capturing the data characteristics and in turn result in unfavorable results [8]. One presented solution to this is using SMOTE, an oversampling technique where synthetic samples of minority classes are created to balance the dataset.

Other studies investigated the usage of several regression methods to predict the duration of court cases [13] and the estimated time of arrival of flights [3], both used features that represented characteristics of the target at some fix starting point for their prediction. The one method proven most effective was AdaBoost which use sequences of weak learners (generally decision trees) that iteratively increase weights on to incorrectly predicted samples in the dataset to improve their prediction in the next iteration. Within the studies common evaluation metrics for the classification tasks were derived from a confusion matrix, and for regression tasks the root mean square error was used.

This thesis will investigate machine learning to predict the impact of a disruption in the production using available features that are found suitable to describe surrounding influences. It will be conducted as a classification task due to the fact that insights regarding how well the model performs on a particular length of days or period of days can be evaluated separately. The model used will be tree based due to being the best performing in the literature study, namely random forests will be used due to its good performance in classification tasks.

Chapter 3

Theory

In this chapter, the background theory of the models, methods and other subjects are presented to cover the relevant theory needed to understand the contents of the thesis.

3.1 Supervised Machine Learning

In the field of machine learning the problem at hand will belong to one of two categories, supervised and unsupervised machine learning. Supervised machine learning problems are formed when each response variable y_i , for all observations $i = 1, \dots, n$ in a dataset, corresponds to a predictor(s) x_i , in contrast to unsupervised learning where the response variable to its corresponding predictors is unknown. Within supervised machine learning the main purpose is to predict the response $y_{n+1, \dots, m}$ for m future observations, and the type of response can either be quantitative or qualitative which separates into two branches, regression problems and classification problems respectively. A quantitative response means that in regression problems the goal is to predict numerical floating or integer value, whereas a qualitative response means that in classification problems the goal is to predict one out of a set number K different classes which can be seen as different categories. If $K = 2$ the problem is referred to as a binary classification problem, and for a $K > 2$ it is known as a multi-class classification problem. Training a supervised model refers to finding some mapping between the predictors and the response by learning from known observation in order to predict a response to newly observed predictors [9].

3.2 Decision Trees and Random Forests

A decision tree is a supervised machine learning technique that involve segmenting the predictor space into simple regions. The name comes from the fact that the segmenting can be presented as a tree which in turn makes it easy to interpret, however, decision trees are generally inferior to more advanced models. An extension of decision trees is a random forest which tend to produce more robust and accurate results, but to grasp the concept of random forests it is necessary to understand the basics of decision trees.

3.2.1 Decision Tree

A decision tree is normally visualized as an upside down tree starting from the top at a root node that through branches and internal nodes connect to leaf nodes at the bottom of the tree. From the root node, the predictor space is split by each branch binary into non-overlapping regions where a prediction of a new observation can be made by assigning the most common class from the training data in that region. Thus, when a tree is grown, each split is made with the intention of minimizing the fraction of training observations that are not the most common, known as the classification error rate given by

$$E = 1 - \max_k (\hat{p}_{mk}) \quad (3.1)$$

where \hat{p}_{mk} denotes the proportion of training observations from the k th class in the m th region. However, growing a tree is done in a top-down greedy fashion called recursive binary splitting since it is computationally infeasible to consider every region simultaneously. It is called top-down because it starts at the root, and greedy because it only considers the best possible split at the current node without considering splits further down in the tree. Let $X = (X_1, X_2, \dots, X_p)$ be a vector of all predictors in a dataset, then for each node every predictor $X_j \in X$ is evaluated regarding to the quality of the split, and the quality is measured from the nodes the split creates known as child nodes. Even though the goal is to minimize the classification error rate, it is not sensitive enough while growing trees. Instead, two other metrics, Gini index and entropy is commonly used in practice. The Gini index is defined as

$$G = \sum_{k=1}^K \hat{p}_{mk} (1 - \hat{p}_{mk}) \quad (3.2)$$

and can be seen as a measure of the total variance of all K classes in a child node. One can note that Gini index is a measure of purity in a child node since G will be small if \hat{p}_{mk} is close to either zero or one for all k , i.e. a small Gini index will indicate a child node is dominated by one class. Furthermore, an average of both child nodes weighted by the number of observations in each node determines the total purity from the split and the predictor that yields the lowest weighted Gini index is chosen. The other metric, entropy is defined as

$$D = - \sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk} \quad (3.3)$$

where it is clear that since $\hat{p}_{mk} \in [0, 1]$ then $-\hat{p}_{mk} \log \hat{p}_{mk}$ and D will be small if \hat{p}_{mk} is close to either zero or one for all k . Thus, similarly to Gini index, entropy will be small if the m th child node is pure. And the choice of predictor is then chosen by the weighted average similarly to the case of Gini index. A decision tree which grows without a stopping criteria runs the risk of high variance and overfitting to the data, this can be handled by pruning which involves systematic methods to reduce the number of splits in a tree or by setting a max depth [9].

3.2.2 Random Forest

An extension of decision trees is a random forest. As mentioned, decision trees run the risk of high variance which in turn can cause the predictor to generalize bad on new observations. One solution to lower the variance of a tree based method is by using bagging which is short for bootstrap aggregating. It is built on the idea of having an arbitrary number B bootstrap datasets which are drawn with replacement from the original dataset, then B number of decision tree classifiers are trained for each dataset and aggregated to one classifier by classifying the response for a new observation as the most commonly class predicted between all B trees, known as majority vote. However, one direct issue that arise with using bootstrap samples is that it cause dependencies. For example, if there exist a strong predictor $X_j \in (X_1, X_2, \dots, X_p)$, then it is likely that it will become the first split in every tree, thus making them highly correlated. The solution to this is adding randomness where rather than using all p predictors in each decision tree, a subset $m < p$ predictors are used instead where m is drawn uniformly from the full set of predictors. Thus there is a non-zero probability of $(p - m)/p$ that the strong predictor is excluded from a bootstrap sample in training, and this extension of bootstrap aggregating is known as a random forest. A common choice of the number of included features is $m = \sqrt{p}$ [9].

3.3 Model Evaluation

Evaluating a model is a an essential step in machine learning in order to both determine performance in the final stages as well as in the model building. To simulate how well a model performs on unseen data the full dataset is divided into a training set and a test set, where the test set represent unseen data points. When building a model, the training set can be further divided into train and validation sets for model selection purposes.

3.3.1 Confusion Matrix

In order to compare the performance of different machine learning models it is important to have evaluation metrics. Within classification tasks, a commonly used representation from which several metrics can be derived is the confusion matrix. The confusion matrix consists of the result of the classification compared to its true value for each class, and from the confusion matrix several useful metrics can be derived. For every class one determine four categories of the predicted points:

- True Positive (TP): The model has predicted the class and the actual value of the point belongs to the class
- True Negative (TN): The model has predicted another class and the actual value of the point belongs to another class
- False Positive (FP): The model has predicted the class but the actual value of the point belongs to another class
- False Negative (FN): The model has predicted another class but the actual value of the point belongs to the class,

and the number of points in each category is calculated. With these, measures such as precision and sensitivity can be calculated for each class as

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (3.4)$$

$$\text{Sensitivity} = \text{Recall} = \frac{TP}{TP + FN}. \quad (3.5)$$

Precision for a class is the percentage of the points the model has predicted to the class that actually belong to the class, it can be seen as a measure of the accuracy of predicting that particular class. Sensitivity for a class, also known as recall, is the percentage of points that belong to the class that were correctly classified, and can be seen as a measure of the strength the model has at predicting that particular class. Since the measurements are evaluated for each class, they prove useful for imbalanced datasets where the overall accuracy can be misleading and fail to capture important insights [10].

3.3.2 Cross-Validation

In the context of training a model, one method used for evaluation is Cross Validation (CV). It is conducted by having a part of the training data left out from training in order to validate the model, these types of splits of the training data is then repeated to create different subsets of the training data for training and validation, and the final evaluation of the model is given by an average of all trained models results. The process can be conducted in a number of ways, one of the most common being k -fold CV where the training data gets randomly divided into k equally sized groups called folds, then $k - 1$ of the folds are used for training and the remaining fold is used for validation. It is conducted k times in order to use all combinations of the k folds, and the final evaluation of the model is the average performance of the k models. An extension of k -fold CV is stratified k -fold CV where the balance of classes is preserved in each fold, i.e., the ratio of each particular class is approximately the same for all folds to ensure that there is training and validation data that represents all classes for each of the k models. Stratified k -fold CV is suitable for classification tasks and essential when a dataset is imbalanced [14].

3.3.3 Boxplot

A boxplot is a graphical representation of a dataset that displays the distribution of the data and identifies outliers, an example of a boxplot is visualized in Figure 3.3.1. It is based on the minimum and maximum values, the first and third quartiles, and the median. The box in the figure represents the Interquartile Range (IQR), which is the distance between the first and third quartiles. The whiskers extend from the box to the minimum and maximum values within 1.5 times the IQR of the box. Observations that fall outside the whiskers can be considered outliers and are visualized as individual points in the figure [6].

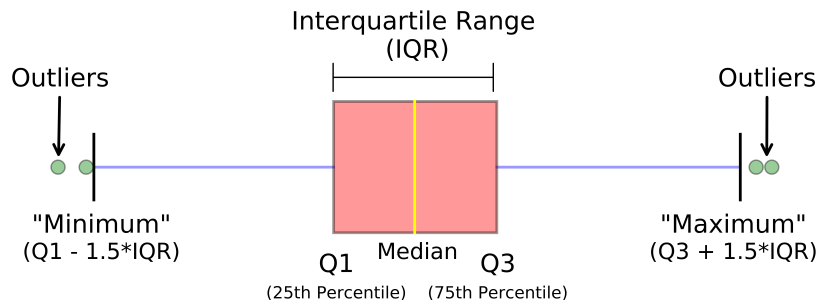


Figure 3.3.1: Representation of a boxplot and its different components [6].

3.3.4 Feature Importance

Feature importance is a concept in machine learning which refers to the process of determining the relative importance of the different features used in a machine learning model with regards to the target variable. By identifying which features are most important, the ones that are not affecting the model performance are redundant and can be removed to reduce the complexity of the model and make more accurate predictions.

Permutation feature importance

Permutation Feature Importance (PFI) is a method to calculate the importance of a feature. The idea is to evaluate the error of the model's prediction after permuting the values of that feature. That is, to shuffle all the rows of data for that feature and train the model with the permuted dataset. If the new model's prediction error increased, the feature is deemed as "important" since the model relied on that feature for the prediction. On the contrary, a feature is deemed "unimportant" if the prediction error after shuffling the feature is left unchanged. This indicates that the model ignored that feature for the prediction. How the algorithm proceeds is presented in Algorithm 1.

Algorithm 1 Permutation feature importance

Input: Trained model \hat{f} , feature matrix X , target vector y , error measure $L(y, \hat{f})$.

1. Estimate the original model error $e_{\text{orig}} = L(y, \hat{f}(X))$

for each feature $j \in \{1, \dots, p\}$ **do**

- Generate feature matrix X_{perm} by permuting feature j in the data X .
- Estimate error $e_{\text{perm}} = L(y, \hat{f}(X_{\text{perm}}))$ based on the predictions of the permuted data.
- Calculate permutation feature importance as difference $FI_j = e_{\text{perm}} - e_{\text{orig}}$

end for

3. Sort features by descending FI .

The list of FI 's is describing the order of features based on importance, the most important feature have the highest FI . Negative FI can occur which simply means that the model by chance performed better without that feature and therefore affected the model, thus, the feature is deemed "unimportant". One downside with PFI is its weakness for correlation between features in the dataset. The importance can then be split between correlated features and reduce the effects of the permutation [12].

3.4 SMOTE

If the classes in a dataset are not relatively equally represented it is referred to as being imbalanced. One way of handling this is by using Synthetic Minority Oversampling Technique (SMOTE) which creates synthetic samples in the feature space close to the minority class in order to even the distribution between classes in a dataset. A synthetic sample is created by randomly taking a data point in a minority class and then by applying k -Nearest Neighbour (k NN), which finds the k closest samples near that point using euclidean distance, assigning a new sample, randomly, in the feature space on a line between one of the k samples and the randomly chosen data point with the same class label as the minority class. The method is reiterated for each minority class until a desired balance in the data set is achieved [4].

3.5 Feature Engineering

Feature engineering is the process of selecting, transforming, and creating new features leveraging the raw data to improve the performance of machine learning models. Effective feature engineering can help to reduce noise and improve accuracy, making it an essential step in the preparatory work before one can start modeling [5].

3.5.1 Feature Creation

Feature creation is a component of feature engineering that is the process of generating new features. The new features can involve combining or extracting information of already existing features. If done well, feature creation can improve the performance of a machine learning model since the new features might be more explanatory.

3.5.2 Normalization

Normalization is a feature transforming method that scales the values of a feature in a data set so that its values lies within the interval of zero and one. By using this method, a feature with a higher range of values will not dominate a feature taking lower values. There are several methods of normalization and one of them are the min-max normalization which is done accordingly. Given a sequence of observations $X = \{x_i\}_{i=1}^n$ one can normalize an arbitrary value x_i belonging to the sequence X as

$$x_i^* = \frac{x_i - \min(X)}{\max(X) - \min(X)}, \quad (3.6)$$

where x_i^* is the normalized value [7]. In 3.6, one can see that if x_i takes on the largest value of the sequence X , the quotient is one. Contrary, if x_i takes on the smallest value in the sequence X , the quotient is zero. Thus, the scale indicates how close a value x_i is to the largest respectively smallest value of that sequence.

3.5.3 Pearson Correlation Coefficient

The Pearson correlation coefficient is a measure of the linear relationship between two parameters, and one use case is to detect linearity between features in a dataset. The correlation coefficient between two features X_1 and X_2 is calculated as

$$\text{Cor}(X_1, X_2) = \frac{\sum_{i=1}^n (x_{1,i} - \bar{x}_1)(x_{2,i} - \bar{x}_2)}{\sqrt{\sum_{i=1}^n (x_{1,i} - \bar{x}_1)^2} \sqrt{\sum_{i=1}^n (x_{2,i} - \bar{x}_2)^2}} \quad (3.7)$$

where $x_{1,i}$ and $x_{2,i}$ are the i th value in the dataset of X_1 and X_2 respectively, \bar{x}_1 and \bar{x}_2 are their corresponding mean, and n is the number of data points. The coefficient takes values between -1 and 1 where a value close to 1 indicates high correlation, a value close to 0 indicates no correlation, and a value close to -1 indicates negative correlation. The coefficient between all features in a dataset can be visualised using a heat map, also referred to as a correlation plot, which can be used to identify correlation in a dataset [9].

Chapter 4

Method

The project was conducted in an iterative manner since the knowledge of the company and the data was limited beforehand. More insights were gained as the project proceeded where new problems and other important aspects were discovered. Using an iterative approach, the new problems were discussed and solutions were found for those deemed relevant to the scope of this thesis. The software used to conduct this thesis were Databricks, SQL and Python including relevant packages for machine learning.

This chapter is divided into three main parts, procurement of data, feature engineering and modeling. The first part describes how the data was created by mapping a disruption to its impact and thereby obtain the feature and target variables. The second part describes how current features was used to transform and create new features which are to be investigated in this report. The last part presents the model and feature selection for the considered models and which hyperparameters that were used during evaluation.

4.1 Defining a Disruption

In this thesis, the disruptions considered are limited to the ones caused at the production site. A disruption is defined as when a production occurrence has not produced the desired quantity. Thus, a disruption is strictly limited to only be determined by the produced quantity versus the desired quantity, and does not consider the details of what actually caused the disruption. Arla has an accepted production error of $\pm 10\%$, and since this thesis investigates impacts due to loss in production, the interesting production instances are the ones with an actual produced volume lower than 90% of the planned volume. Considering that this is the accepted error set by Arla, it is assumed that all production with a percentage above 90% are covered by safety stock and should not lead to any shortage.

4.2 Procurement of Data

When using supervised machine learning methods, the data must consist of feature variables and target variables. In this case, the feature variables are the ones describing information about the disruption and the two target variables are the delay and the

duration of the corresponding shortage. This particular dataset was not available from beforehand, so databases were combined and assumptions were made to map the feature variables to their targets.

The scheme of how the data was created is the following:

1. Find days in sales data which do not deliver the ordered quantity.
2. Use assumptions to convert these days into shortages.
3. Find the disruptions in production which have not produced the planned quantity.
4. Use assumptions and rules to map the disruptions to corresponding shortages.

When the data had been created, the additional features used for separation of the products were added. The databases and the assumptions will be described in this section, as well as how they were used to create the data.

4.2.1 Sales and Production Data

The sales data consists of $\sim 273\,000$ rows of data representing the total sales on a daily level for all products. The relevant features of the sales data are the date, the ordered quantity and the delivered quantity. With this, a quotient of the delivered quantity by the ordered quantity was calculated for every product and day of sales. Every row with a quotient below 1 is an instance of where a delivery has not met the ordered quantity. With the sales filtered on this criterion, the new dataset consists of $\sim 33\,000$ rows of sales. These were used to determine the shortages which is described in detail in Section 4.2.2.

The production data consists of $\sim 23\,000$ rows of data where each one represents information about the total production for a specific product on a daily level. The relevant features of the production data are the date, the planned output quantity and the actual output quantity. Similarly as for the sales data, a quotient of the actual output quantity by the planned output quantity was calculated for each day of production and product. The disruptions could then be obtained by taking all rows with a production quotient below 90%. The resulting dataset of disruptions was then reduced to ~ 2800 different production instances. These are the only disruptions that, by previous assumptions regarding accepted production error, can lead to a shortage, and how they are linked is described in Section 4.2.3.

4.2.2 Defining Shortage

The theoretical definition of a shortage describes it as a situation where the demand of a product exceeds the available supply. An example of this situation is presented in Figure 4.2.1, where the undelivered areas take place when the demand has not been met and a shortage is present. A shortage in this thesis is a period of consecutive days where each day fails to meet the demand. A shortage can also have a length of one day if there is just one day of unsatisfied deliveries.

In practice, companies may have an acceptable Delivery Accuracy (DA) of which they consider anything above it as a successful delivery. The theoretical definition of a shortage can therefore be less meaningful since it indicates shortages in cases where the company

is satisfied. A more accurate definition is then to use the acceptable DA as a threshold, and the consecutive days with deliveries below the threshold defines as shortages, since the company has not fulfilled their goal. At Arla, the acceptable DA is 98.5% which is used as a threshold in this thesis.

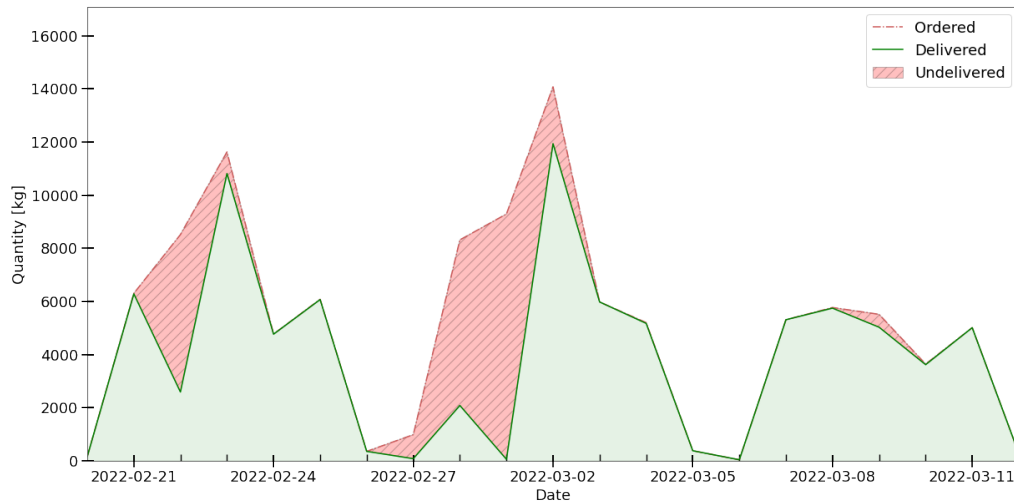


Figure 4.2.1: An example of when the delivered quantity does not meet the ordered quantity. Each area of undelivered products indicates a shortage.

Another criterion set by Arla is that in order for a shortage to be deemed as recovered, it must achieve acceptable DA for two consecutive deliveries. Therefore, further conditions must be set to handle the case where shortages achieve acceptable DA for one delivery in that period. The previous definition would deem the shortage as recovered and split the actual shortage into two periods, i.e., two separate shortages. Figure 4.2.2 presents an example of a period with undelivered goods with an acceptable DA present. The updated definition of a shortage will now consider this period of time as one shortage instead of two separate ones.

To summarize, a shortage in this thesis is defined as a period of time which starts when the DA is below the acceptable 98.5% and ends when the DA is recovered for two consecutive deliveries. In Figure 4.2.3, a period of time is visualised where all modified criteria for shortage are used. The first shortage has a delivery with acceptable DA for one day which is taken into consideration. The second shortage is separated from the first one since it has been enough consecutive deliveries with accepted DA. These two shortages have now been successfully identified.

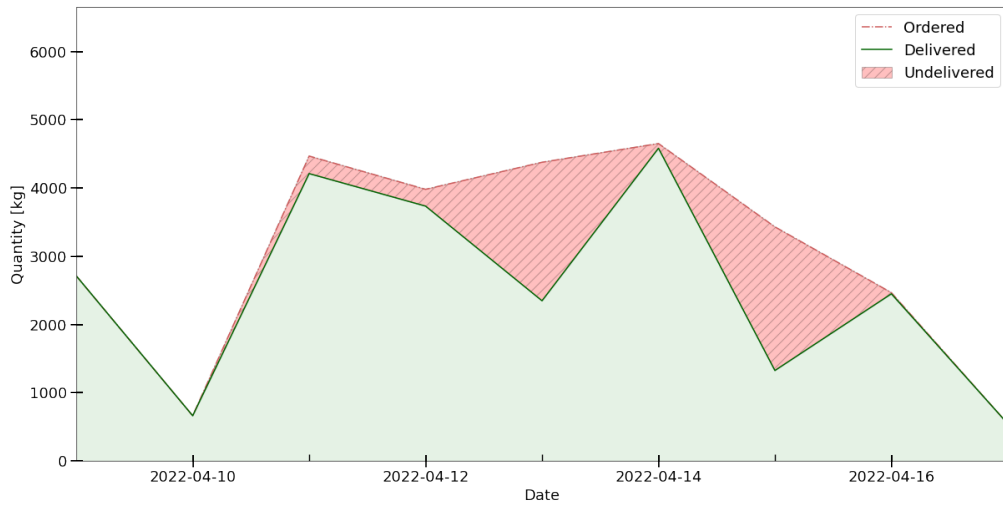


Figure 4.2.2: A shortage with one day of acceptable DA.

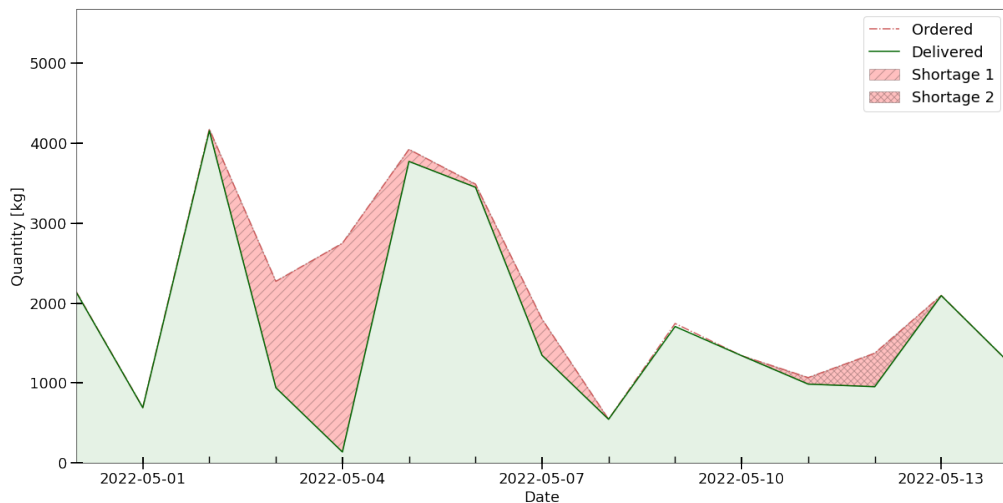


Figure 4.2.3: An example of how the finished definition identifies two shortages.

4.2.3 Shortage Mapping

The basic idea behind mapping a shortage to a disruption is to investigate a period of time ahead of the disruption and check if any shortages start during that period. If a shortage is found, the production data, i.e., the feature variables for that disruption, are linked with the delay until the shortage starts and its duration, namely, the target variables. If no shortage is found in the search period, the delay and duration linked to that disruption will be set to zero which indicates that the disruption did not cause any shortage. An instance of a disruption being linked to a shortage constitutes to one row of data consisting the feature and target variables desired when later implementing the predictive model. Since different products may not behave similarly, assumptions on how the shortages are linked to a disruption will define the rules for the mapping.

Together with Arla, these two main assumptions were made:

- A corresponding shortage cannot occur the same day as a disruption.
- A disruption cannot be linked to a shortage which occurs after the succeeding time of production.

The second assumption poses challenges due to the varying production frequencies of different products, that is, how often production occurs expressed as productions per day. Additionally, the uncertainty surrounding the timing of upcoming production poses further difficulties. To address these challenges, the time period ahead of a disruption, within which a shortage is searched for, is set as the inverse of the product's production frequency, which is the average number of days between productions. This assumes that the upcoming production will be produced similar to previous productions. To obtain more accurate frequencies, a unique frequency was calculated for each disruption, taking into account the potential variations in production frequency due to varying demand. To determine the production frequency for a disruption, the length of the time period starting from two production instances prior to the disruption and ending at the current instance was calculated. Since three instances in total were considered in calculating the length, the frequency was derived by dividing three by the length of that period.

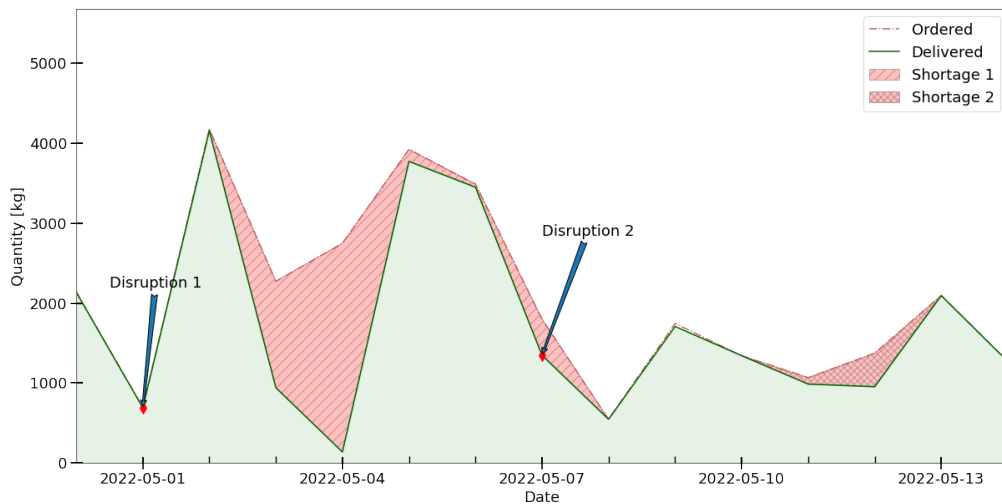


Figure 4.2.4: An example of two shortages and when their corresponding disruption occurs.

With rules declared on how to search for a shortage, the next step was to set up a rule on how to select the correct shortage. If more than one shortage is found in the search period, one must choose between them. Ideally, since the search period should end at the next time of production, the disruption being investigated is the only one able to cause those shortages. However, there are other factors in the supply chain that can cause disturbances that lead to shortages in the same period, regardless of any disruptions. Therefore a third assumption was made, the shortage with the largest quantity of undelivered products is to be mapped to the disruption, i.e., a disruption is assumed to cause larger shortages than other factors. Figure 4.2.4 presents the same shortages as in Figure 4.2.3 but points out the disruptions present during that period of

time. In this case, the first disruption is mapped to the first shortage and likewise so for the second disruption and shortage. Figure 4.2.5 presents an example of how the third assumption is applied. Here one can see how two shortages are present in the search period, and in this case the disruption is mapped to the second shortage since the impact is larger in terms of undelivered quantity.

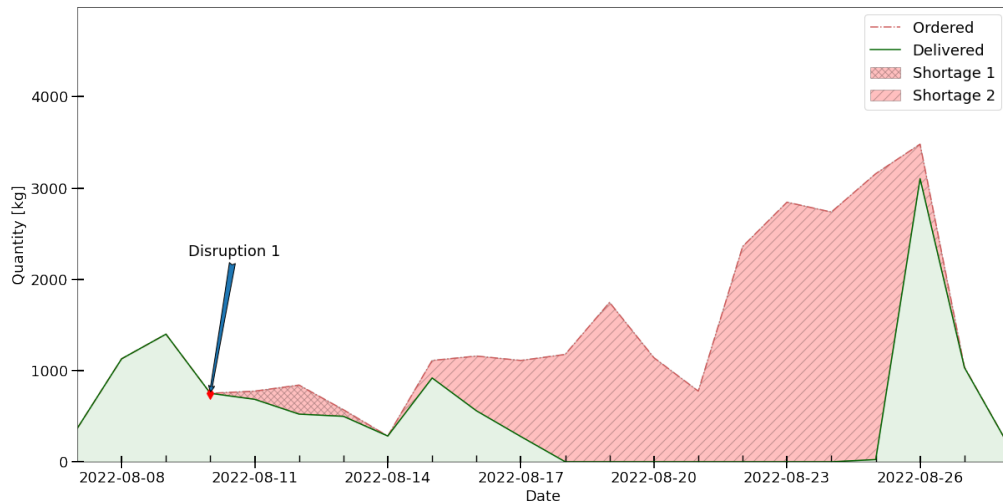


Figure 4.2.5: An example of a disruption with two succeeding shortages.

To summarize, from the day of the disruption a search for a shortage is made forward in time for a period determined by the product’s production frequency. If any shortages are identified, the disruption is mapped to the shortage with largest quantity of undelivered products. If no shortage is found, the disruption did not affect the sales and the disruption is mapped as no shortage.

4.2.4 Additional Features

Once the initial dataset had been formatted as desired, additional features were incorporated into the dataset. The selection of these features was based on intuitive understanding of factors that might influence the delay and duration. These features were later subjected to normalization and feature engineering techniques to be generalized across different products. As previously mentioned, the demand for various products is significantly influenced by seasonality, which can be accounted for in the model by incorporating variables designed to capture seasonal behavior. Rather than introducing seasonality features and expanding the feature set used in the modeling process, this thesis adopts a different approach. It leverages the existing data obtained from Arla for daily demand forecasting, assuming that this data sufficiently captures the seasonal patterns present in the dataset.

The forecasting was added on a daily basis for a period of seven days both preceding and following the time of the disruption. This approach aims to capture the product’s behavior over time in both retrospective and prospective directions. Additionally, the actual demand for the preceding seven days was included, enabling a comparison between the forecast and actual demand. This allowed for an assessment of whether the forecast

accurately predicted higher or lower demand compared to the actual values, which was assumed to influence the overall impact. Another feature considered to be of relevance was the current stock level at the end of the day of a disruption. A large stock level has the potential to compensate for unproduced volumes, thereby mitigating the occurrence of a shortage. The final additional feature incorporated was the current production frequency for each specific product as it is presumed that the impact is influenced by the frequency of production. How the frequency was calculated is described in Section 4.2.3.

4.3 Feature Engineering

Since the collected data is still scaled in terms of each individual product, methods for generalizing the data was used. The two methods of feature engineering used were feature creation and feature transformation. How these methods were used will be described in detail in this section.

4.3.1 Feature Creation

This subsection describes the methods used to create new features from the already existing ones.

Forecast of demand

The daily forecasting features captures how the demand behaves around a disruption and is used to investigate how different measures of those features will affect the model's prediction of the impact. The forecasting quantity is measured in kg which is product specific partly due to the fact that different products does not have the same weight, as well as the possibility that different products sell more than others on average, i.e., a high demand for one product can be considered low for another. Since the demand back in time is available, a ratio calculated by dividing the forecast by the demand describes the accuracy of the forecast. The forecasting accuracy was calculated using the total demand and forecast quantity of the desired period as

$$\text{Forecast accuracy} = \frac{\sum_{i=0}^d F_i}{\sum_{i=0}^d D_i}, \quad (4.1)$$

where F_i and D_i is the daily forecast and demand quantity respectively for day i before the disruption and d is the length of the desired time period in days. Measuring this feature as a ratio makes it unitless and removes the product specificity of the forecasting features back in time. The defined accuracy can take on values above one which indicates that the forecast has predicted more than the actual demand. Since the planned production volume is based on the forecast they should ideally agree with each other, and if for example the forecast cause overproduction and a disruption occurs, the loss in produced quantity might be covered by the overproduction and therefore the actual quantity produced meets the demand even tough a disruption has occurred.

Stock level

The closing stock level at the day of a disruption is measured in kg, which with the same reasoning as for the daily forecasting is product specific. To remove product specificity, a ratio calculated by dividing the forecast of the demand one day after the disruption by the stock level the day of the disruption was used. This ratio describes how the next day's forecasting stands in relation to that day's initial stock level. This feature was calculated since the ratio can indicate whether next day's forecast of the demand, which optimally is the same as the unknown actual demand for that day, can be covered by the current stock. If the stock level can cover the next day's demand, the disruption will likely cause a less harmful impact, but if the ratio indicates a relatively small stock level, the impact might be more severe.

4.3.2 Feature transformation

The daily forecast ahead in time is used to measure if the demand is high or low, and the quantity of the forecast, as earlier mentioned, is product specific. Since data of the actual demand is not available for the time after the disruption, the forecast accuracy used in Section 4.3.1 cannot be used for these features. To deal with this problem, normalization of these forecasting features for each product scales them into values between zero and one. The normalization is done by for each specific product select the sequence of all its forecast data and obtain the minimum and maximum values of the sequence. When these values are obtained, one can transform all future forecasting features by using the minimum and maximum values and normalize according to (3.6) presented in Section 3.5.2. The forecasting features are now normalized in relation to all forecasting data and describes that the demand is higher when the normalized value is closer to one, and lower when the normalized value is closer to zero.

4.3.3 Final features

The final features from this section that were used for the testing of the models are presented in Table 4.3.1. Features 1-2 are derived from the production data. Features 3-5 are the forecasting accuracy for different time periods. Features 7-8 are the normalized forecasting for different time periods.

Table 4.3.1: A description of the features investigated in this thesis.

Nr.	Variable/Feature	Description
1	<code>quotient</code>	How much of the planned production quantity that was actually produced at the production instance of the disruption.
2	<code>freq</code>	The current production frequency, calculated using the three latest production instances including the disruption.
3	<code>stock</code>	The forecasting of tomorrows predicted sales divided by the current stock level.
4	<code>fc_0</code>	The forecasting accuracy for the day of the disruption calculated using (4.1) where $d = 0$.
5	<code>fc_Bf</code>	The forecasting accuracy calculated using (4.1) where d is the number of days corresponding to the inverse of that specific product's production frequency.
6	<code>fc_B7</code>	The forecasting accuracy one week back in time calculated using (4.1) where $d = 7$.
7	<code>fc_Ff</code>	The mean value of the k upcoming days for the normalized forecast of the demand where k is the number of days corresponding to the inverse of that specific product's production frequency.
8	<code>fc_F7</code>	The mean value of the 7 upcoming days for the normalized forecast of the demand.

4.4 Data Preprocessing

Data preprocessing is important in order to clean the data and improve the performance of machine learning models. To detect outliers, the data was graphically represented using boxplots. Data points with values lying outside 1.5 times the IQR were, as described in Section 3.3.3, deemed outliers. All relevant points deemed outliers were then investigated individually and if a data point was considered to be abnormal based on knowledge of the production and sales, it was removed from the dataset. In Figure 4.4.1, one can see three examples of how boxplots are used to detect outliers of features. One can especially see that the box of `fc_0` was compressed due to great outliers, which needed further investigation. This can be due to the daily demand being volatile, hence the daily forecasting is difficult to predict which causes instances with a higher ratio. One can then see in the other two boxplots that the ratios covers a smaller range of values,

hence the forecast is more accurate when being evaluated on a longer time period, and possibly not as sensitive as on a daily basis.

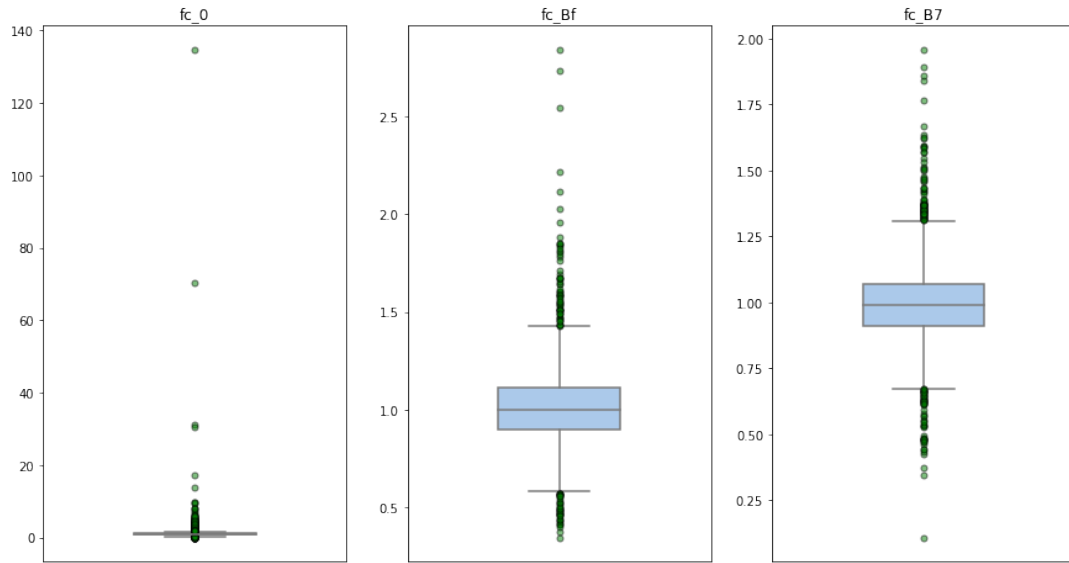


Figure 4.4.1: Boxplots for all data of features 1-3 from Table 4.3.1.

When using forecasting features to describe behaviour before and after the event of a disruption, it is likely that correlation exists between the features. To investigate this, a correlation plot was used to detect possible correlation between features. The correlation plot is visualized in Figure 4.4.2. The preprocessing resulted in a dataset containing 1076 rows of data, each for an individual disruption, gathered from a total of 145 unique products.

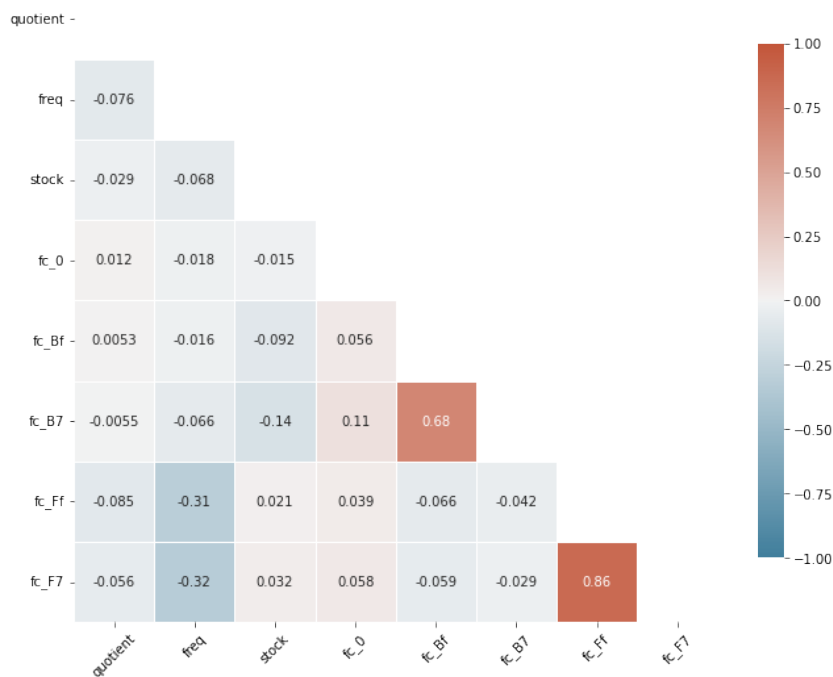


Figure 4.4.2: The correlation between all features in the dataset.

4.5 Implementation

In the case of multiple target prediction, a common approach is to separate the target variables and employ individual models for each one. This approach assumes that there is no significant correlation between the target variables. In the thesis, this was applied to predict the delay and duration, resulting in the training of two separate models. The designed implementation require the models to be used sequentially. The first model is used to predict the duration, if the duration is predicted to be zero, it indicates the absence of any delay, as no shortage occurs. On the other hand, if the predicted duration is greater than zero, the second model is used to predict the delay associated with the shortage. The model responsible for predicting the duration was trained using the entire dataset, while the second model, which solely predicts the delay if a shortage occurs, was trained on instances where the duration was greater than zero. To illustrate the implementation of the two models for predicting the impact of a disruption, a flowchart is provided in Figure 4.5.1.

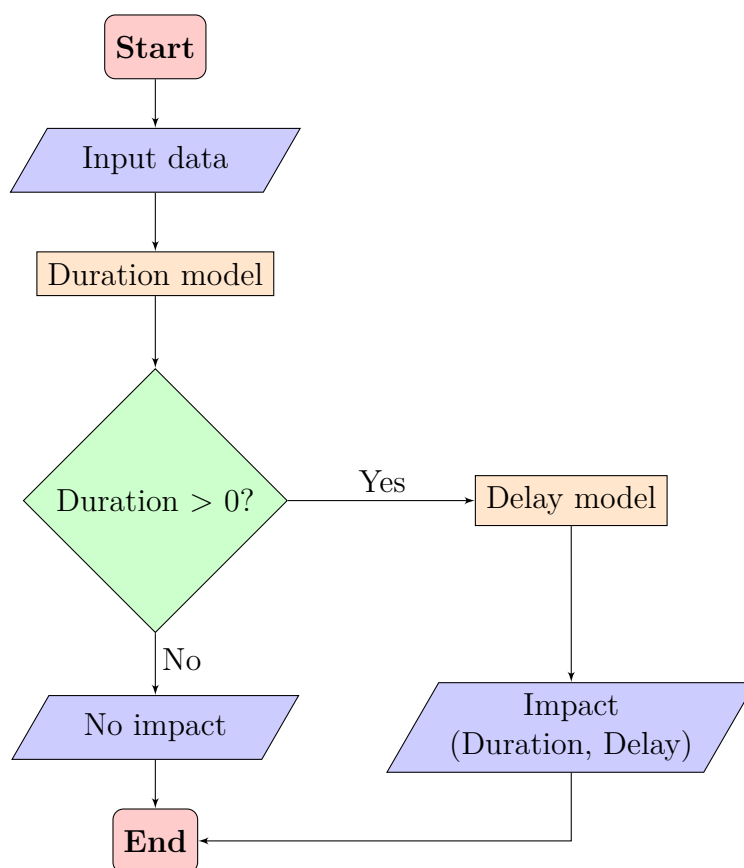
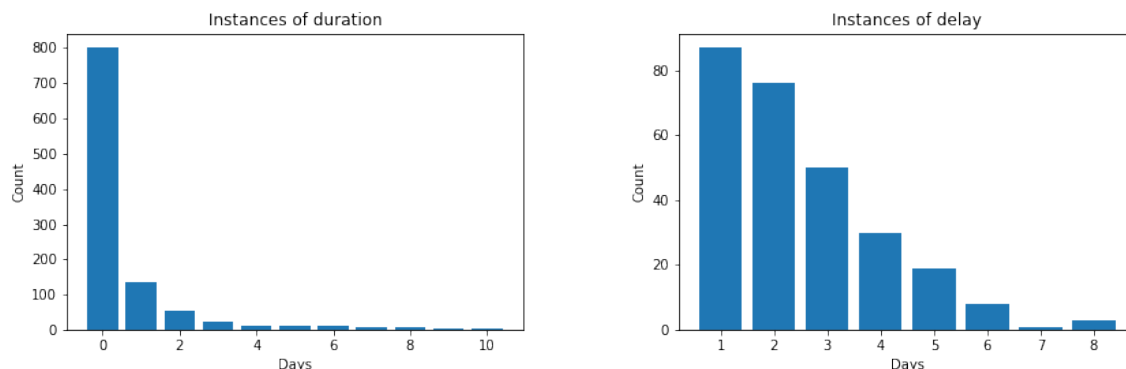


Figure 4.5.1: Flowchart of how the models are used to predict the impact.

4.5.1 Treatment of Imbalanced Data

Figure 4.5.2 illustrates the distribution of the two datasets used for the delay and the duration models respectively. The barcharts highlight the imbalanced nature of the data, particularly in the case of the duration dataset, where instances with zero days of duration (indicating no shortage) are overrepresented. The imbalanced datasets were handled using two techniques. The first technique is to group the number of days in the

dataset into larger classes that instead represents a span of days, which is also done due to the low sample size that led to some classes having very few data points. The data was grouped into different classes individually for the delay and duration. The classes were produced together with Arla where each class represents different levels of severity, and are presented in Table 4.5.1. After the days had been grouped, the second technique is to implement SMOTE to create evenly distributed classes.



(a) The distribution of values in the duration dataset. (b) The distribution of values in the delay dataset.

Figure 4.5.2: Barcharts of the datasets used for the two different models.

Table 4.5.1: The classes used for each model.

Duration		Delay	
Class	Interval in days	Class	Interval in days
0	0	1	1
1	1	2	2-3
2	2-3	3	4+
3	4+		

(a) The classes after the duration was grouped, 4+ denotes four or more days.

(b) The classes after the delay was grouped, 4+ denotes four or more days.

4.5.2 Feature Selection

Based on the correlation plot presented in Figure 4.4.2, it is evident that there exists a strong correlation between the two forecasting features related to prior days, as well as between the two forecasting features related to upcoming days. Therefore, the models does not incorporate the correlated features simultaneously and only one feature for preceding forecasting and one for future forecasting were used concurrently. This resulted in four combinations of alternating feature setups which are presented in Table 4.5.2. In the correlation plot, there is no evident correlation observed among the remaining features and the features in the alternating setups. As a result, all of these features

were incorporated into the model (considered fixed) and tested in combination with the four alternating feature setups, thus, eliminating high correlation between features used in the model. The fixed features were `quotient`, `freq`, `stock` and `fc_0` and the four alternating feature setups are presented in Table 4.5.2. The four combinations of fixed and alternating features are the feature setups explored in this thesis.

Table 4.5.2: The alternating groups of the feature setup.

Setup nr	Features
1	<code>fc_Bf</code> & <code>fc_F7</code>
2	<code>fc_Bf</code> & <code>fc_Ff</code>
3	<code>fc_B7</code> & <code>fc_Ff</code>
4	<code>fc_B7</code> & <code>fc_F7</code>

4.5.3 Models

In the literature review in Chapter 2, the random forest performed best for similar kind of predictions. Random forests are simple yet powerful predictors, and having a small dataset, it was assumed that a less complex model would perform better than a complex one. Thus, random forests with Gini index criterion was used for the two models when evaluating the performance of the different feature setups. To find the best model for evaluating each setup, the random forests used a grid search technique for hyper parameter tuning. The grid search uses a predetermined set of hyper parameters where it tests all different combinations. For each combination, a stratified 5-fold CV was used to train and evaluate that model using 80% of each full dataset. The model with highest average accuracy is deemed the best model for that particular feature setup. The parameter values used were:

- **Number of trees:** 50, 100, 150, 200, 500
- **Maximum Depth:** 5, 10, 15, 20

Due to the low sample size of the datasets, which led to some classes having very few datapoints, the model training for each feature setup was always performed on the grouped dataset. To investigate whether an oversampling technique would improve the performance of the models or not, each combination of the feature setups was also trained using SMOTE on the grouped datasets.

The best model for each feature setup and oversampling method was then evaluated on an independent test set consisting of the remaining unused 20% of each dataset in terms of accuracy, recall and precision, which are described in Section 3.3.1. This resulted in a total of eight different models investigated for the duration and delay respectively. When the best feature setup and oversampling method was selected for the two models, feature importance was used to investigate what impact the features had on each model's performance. For that, PFI was used which is described in Section 3.3.4.

Chapter 5

Results and Analysis

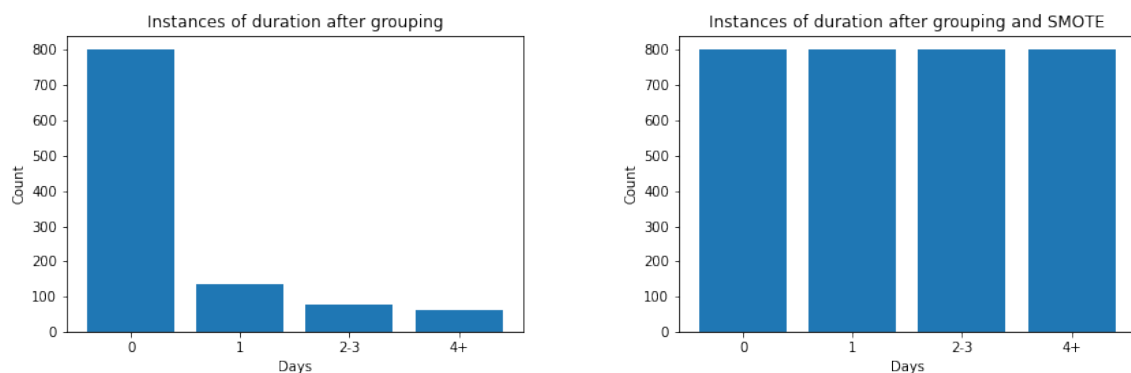
This chapter presents the findings obtained from comparing the various models outlined in Section 4.5.3. The results consist of tables that describe the optimal models for each setup, along with their corresponding scores on the test set in terms of accuracy, recall, and precision. From the grid search, the best performing hyperparameters was obtained for each feature setup. The validation scores of each fold in the CV process with these hyperparameters are presented as boxplots together with their test scores. The results also include barcharts of the datasets obtained before and after applying SMOTE as an oversampling technique.

5.1 Duration Model

In this section, the results of the duration models and dataset are presented.

5.1.1 Oversampling

The distribution of values in the duration dataset after grouping and applying SMOTE is presented in Figure 5.1.1.

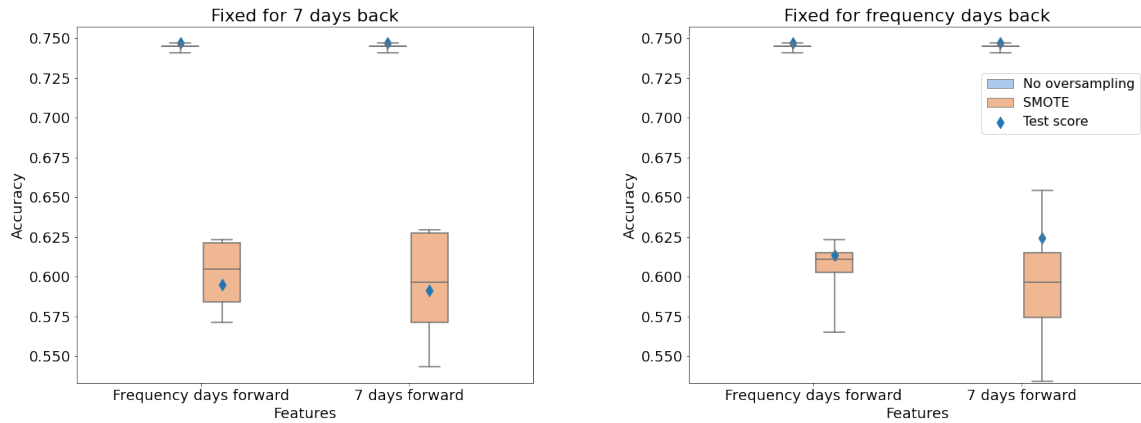


(a) The distribution of values in the duration dataset after grouping. (b) The distribution of values in the duration dataset after grouping and applying SMOTE.

Figure 5.1.1: Barcharts of the duration dataset before and after applying SMOTE.

5.1.2 Results for Feature Setups

This section presents a comparison between the best models for each feature setup with and without SMOTE using boxplots and is visualized in Figure 5.1.2.



(a) The test and validation accuracy for different feature setups when fc_B7 is fixed. (b) The test and validation accuracy for different feature setups when fc_Bf is fixed.

Figure 5.1.2: Boxplots of the duration model's validation and test accuracy. The accuracies were obtained using no oversampling method and SMOTE.

No Oversampling

This section presents the test scores of the best duration model for each feature setup using no oversampling method, the results are presented in Table 5.1.1.

Table 5.1.1: Test results of the best duration model for different feature setups without oversampling.

No Oversampling						
Model	Trees	Depth	Accuracy	Class	Precision	Recall
1	50	5	0.75	fc_B7 and fc_Ff		
				0	0.75	1.0
				1	0.0	0.0
				2-3	0.0	0.0
				4+	0.0	0.0
2	50	5	0.75	fc_B7 and fc_F7		
				0	0.75	1.0
				1	0.0	0.0
				2-3	0.0	0.0
				4+	0.0	0.0
3	50	5	0.75	fc_Bf and fc_Ff		
				0	0.75	1.0
				1	0.0	0.0
				2-3	0.0	0.0
				4+	0.0	0.0
4	150	5	0.75	fc_Bf and fc_F7		
				0	0.75	1.0
				1	0.0	0.0
				2-3	0.0	0.0
				4+	0.0	0.0

SMOTE

This section presents the test scores of the best duration model for each feature setup using SMOTE, the results are presented in Table 5.1.2.

Table 5.1.2: Test results of the best duration model for different feature setups with SMOTE.

SMOTE						
Model	Trees	Depth	Accuracy	Class	Precision	Recall
5	100	15	0.59	fc_B7 and fc_Ff		
				0	0.80	0.74
				1	0.12	0.12
				2-3	0.12	0.15
				4+	0.17	0.27
6	100	15	0.56	fc_B7 and fc_F7		
				0	0.81	0.74
				1	0.09	0.09
				2-3	0.12	0.15
				4+	0.15	0.27
7	50	20	0.61	fc_Bf and fc_Ff		
				0	0.80	0.75
				1	0.21	0.21
				2-3	0.17	0.20
				4+	0.16	0.27
8	50	20	0.62	fc_Bf and fc_F7		
				0	0.79	0.76
				1	0.16	0.15
				2-3	0.30	0.30
				4+	0.17	0.27

5.1.3 Analysis of Duration Models

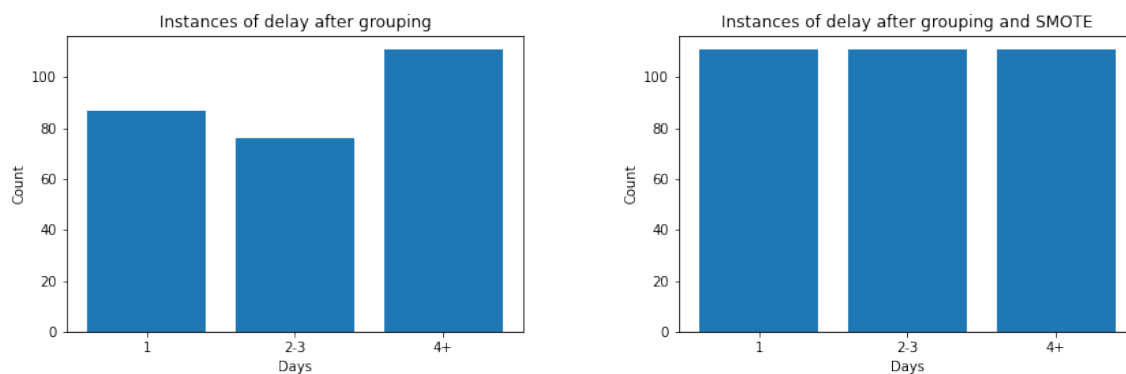
Looking at the boxplots for the duration model in Figure 5.1.2 it is clear that regardless of the combination of features tested, the models without oversampling yield a better overall accuracy. However, the overall accuracy does not depict the overall performance of the model which becomes apparent if one were to investigate the results presented in Table 5.1.1. All models have successfully predicted all points with a duration of 0 days as explained by 100% recall for the class, although, it is evident that the model is naive in the sense that it predicts all points as duration 0 and fails to capture any of the other classes. Given that the purpose of the duration model is to identify shortages following a disruption, having a model that is unable to give any indications of such an instance omits the purpose. Instead, examining the models in Table 5.1.2 that were trained with oversampling, an improvement of the predicting capabilities of the minority class is palpable for all feature combinations, especially for model 7 and model 8. Comparing the two, model 7 show higher recall and precision when predicting a 1 day duration, but model 8 exhibit a higher recall and precision when predicting 2-3 days as well as a slight increase in precision when predicting a duration longer than 4 days. Since the classes 2-3 and 4+ days are deemed to have more severe consequences than a 1 day shortage, model 8 is chosen to be the one with the best performance.

5.2 Delay Model

In this section, the results of the delay models and dataset are presented.

5.2.1 Oversampling

The distribution of values in the delay dataset after grouping and applying SMOTE is presented in Figure 5.2.1.

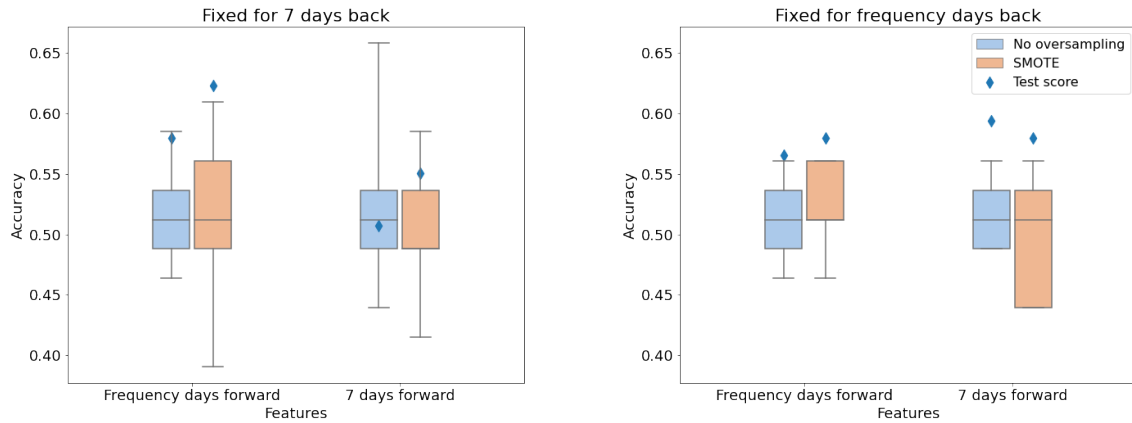


(a) The distribution of values in the delay dataset after grouping. (b) The distribution of values in the delay dataset after grouping and applying SMOTE.

Figure 5.2.1: Barcharts of the delay dataset before and after applying SMOTE.

5.2.2 Results for Feature Setups

This section presents a comparison between the best models for each feature setup with and without SMOTE using boxplots and is visualized in Figure 5.2.2.



(a) The test and validation accuracy for different feature setups when fc_B7 is fixed.

(b) The test and validation accuracy for different feature setups when fc_Bf is fixed.

Figure 5.2.2: Boxplots of the delay model's validation and test accuracy. The accuracies were obtained using no oversampling method and SMOTE.

No Oversampling

This section presents the test scores of the best delay model for each feature setup using no oversampling method, the results are presented in Table 5.2.1.

Table 5.2.1: Test results of the best delay model for different feature setups without oversampling.

No Oversampling						
Model	Trees	Depth	Accuracy	Class	Precision	Recall
1	100	100	0.58	fc_B7 and fc_Ff		
				1	0.63	0.55
				2-3	0.67	0.32
				4+	0.54	0.79
2	500	5	0.51	fc_B7 and fc_F7		
				1	0.50	0.45
				2-3	0.38	0.16
				4+	0.54	0.79
3	50	5	0.57	fc_Bf and fc_Ff		
				1	0.53	0.45
				2-3	0.50	0.37
				4+	0.61	0.79
4	200	5	0.59	fc_Bf and fc_F7		
				1	0.61	0.50
				2-3	0.57	0.42
				4+	0.59	0.79

SMOTE

This section presents the test scores of the best delay model for each feature setup using SMOTE, the results are presented in Table 5.2.2

Table 5.2.2: Test results of the best delay model for different feature setups with SMOTE.

SMOTE						
Model	Trees	Depth	Accuracy	Class	Precision	Recall
5	50	5	0.62	fc_B7 and fc_Ff		
				1	0.67	0.45
				2-3	0.65	0.58
				4+	0.59	0.79
6	50	5	0.55	fc_B7 and fc_F7		
				1	0.55	0.50
				2-3	0.50	0.37
				4+	0.57	0.71
7	500	5	0.58	fc_Bf and fc_Ff		
				1	0.57	0.36
				2-3	0.45	0.53
				4+	0.67	0.79
8	100	5	0.58	fc_Bf and fc_F7		
				1	0.59	0.45
				2-3	0.42	0.42
				4+	0.67	0.79

5.2.3 Analysis of Delay Models

Looking at the boxplots for the delay model in Figure 5.2.2 it is clear that the results in Figure 5.2.2b display more densely distributed results. However, the models trained in Figure 5.2.2a demonstrate instances where a higher accuracy has been achieved. There are no clear distinctions between oversampling and no oversampling in the boxplots. In Table 5.2.1, where the test results without oversampling are presented, model 1 and model 4 show higher accuracy as well as recall and precision on average than model 2 and model 3. Comparing these to the results in Table 5.2.2, where oversampling was

used in training, there is a clear increase in the ability to predict the minority class of 2-3 days for all feature setups. Furthermore, the model that performs best regarding accuracy, and on average, precision and recall with oversampling is model 5, which also outperforms all models without oversampling.

5.3 Feature Importance

This section presents the delay and duration models using the best feature setup from Section 5.1 and 5.2, and will investigate the importance of the features used in the models. This was done using the feature importance technique PFI.

5.3.1 Duration Model

The setup for the best duration model is presented in Table 5.3.1. The feature importance for the best duration model is presented in Figure 5.3.1.

Table 5.3.1: Best feature setup and parameters for the duration model.

Trees	Depth	Oversampling	Features					
50	20	SMOTE	quotient	freq	stock	fc_0	fc_Bf	fc_F7

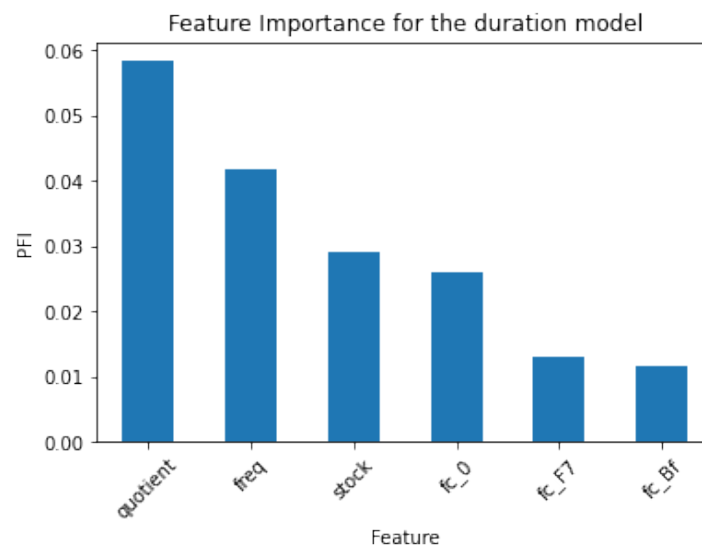


Figure 5.3.1: The PFI for the features in the best duration model.

5.3.2 Delay Model

The setup for the best delay model is presented in Table 5.3.2. The feature importance for the best duration model is presented in Figure 5.3.2.

Table 5.3.2: Best feature setup and parameters for the delay model.

Trees	Depth	Oversampling	Features					
50	5	SMOTE	quotient	freq	stock	fc_0	fc_B7	fc_Ff

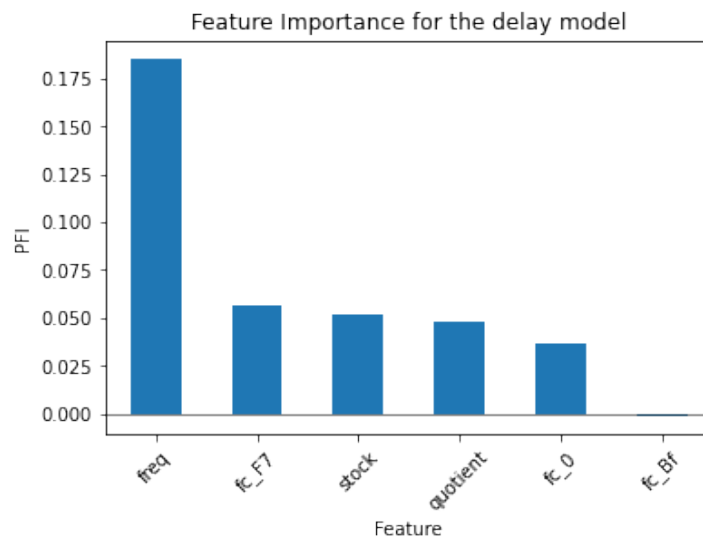


Figure 5.3.2: The PFI for the features in the best delay model.

Chapter 6

Discussion and Conclusions

This chapter presents the discussion about the results and other general findings of this thesis.

6.1 Duration Model

Even though different models were explored in Chapter 5, the best one does not necessarily produce a satisfactory result. Despite an increased ability to predict the minority classes with the use of SMOTE in the model training, the best model for duration still produce too low precision and recall scores to be used as a basis for any type of proactive action against a shortage period. Arguably, the most important metric when predicting the duration of a shortage is the precision of predictions with duration larger than zero. If the model fails to predict that a shortage will occur it simply yields that no action will be taken against it. However, if the model predicts a longer period of shortage, and Arla were to take action against it by for example reducing their delivery quantities to even out the sales and prevent the issue of empty shelves, this could instead lead to missed sales opportunities and loss of income. Therefore, having high precision in these predictions is the most important aspect, and as of now the precision of the best model is simply too low to be used in practice. Thus, the model produced without oversampling, that only predicts 0 days of shortage, could in fact be the better choice.

Using SMOTE did nonetheless allow the random forest to capture the minority classes which should be seen as a vast improvement from a modeling stand point. Having a model that only predicts 0 days is essentially the same as having no model at all. Even though grouping was performed in order to counter the imbalance of the duration dataset, it is clear from Figure 5.1.1a that the dataset remained zero heavy. This could explain why oversampling was imperative, and shows that learning from imbalanced datasets is a great challenge.

The feature importance for the duration model presented in Figure 5.3.1 shows that `quotient` is the most important feature for the model. This goes in line with what was expected, the main assumption for this thesis was that following a production instance where not all the planned goods were produced could cause a shortage, and for this to be deemed the most important feature does in a way confirm this assumption. The second most important feature according to the PFI was the production frequency `freq`,

one reason for this could be that if the relative production frequency is high, there are more opportunities close in time for the production to catch up on lost production and vice versa. One thing that stands out is the fact that the forecasting of expected sales `fc_Bf` is regarded as the least important. This could be explained by the fact that the relative high demand, depending on seasonality or the type of product, is already being accounted for in the planned quantity at each production instance, thereby leaving the feature redundant. Another reason could be that the normalization step yields a noisy feature, and as a consequence hardly any information can be extracted from it.

6.2 Delay Model

Looking at the boxplots in Figure 5.2.2, one can see that the spread of the validation accuracy take on a range of 20% which is rather large and suggest that the model is unreliable. Using the test score one can see that it is close to the median or above which can be used as an argument for the model not being overfitted, and thus reliable in that sense. From the test results in Table 5.2.2 one can see that the best model, model 5, is good at capturing the group of 4+ with a recall close to 80%, but a lower precision compared to the other classes. This could be caused by the fact that despite using oversampling the model remains biased to predicting the majority class. The other classes show decent results but for class 1, the recall is the lowest of 45%. From this it follows that the remaining 55% are distributed over the other classes and have thereby predicted a longer delay than it actually was. This could indicate that there is a chance to take proactive action against any shortage related consequences when in fact the shortage begins the very next day. However, from this perspective it could be argued that it is more important to capture shortages that begin after a longer delay since these allow time for a counter strategy to be formed and set in motion, thus making it more important to capture the two classes with longer delays.

Using SMOTE yields the best model for the delay, and even though using no oversampling technique resulted in some classes having a higher recall and precision, the models using SMOTE produce a more evenly distributed recall and precision with a higher average over the different classes which is preferable. Comparing the distribution of the delay before and after grouping in Figure 4.5.2b and 5.2.1a, it is clear that after using the grouping technique a much more balanced dataset was achieved. Furthermore, using SMOTE improved the ability to capture the minority class of 2-3 days of delay which from the previously stated arguments should be seen as an important class to identify.

Looking at the feature importance in Figure 5.3.2, it is clear that the `freq` feature is considered to be the most important to the model. This is highly reasonable considering that products that are produced more often could be intended to only cover the demand for a shorter period forward in time with each production instance, and thereby could a loss in produced goods yield ramifications quicker. From the figure, one can also see that all other features are about equally important except `fc_Bf`. This feature have a negative PFI close to 0 which indicates that it is not improving the model's performance and therefore unimportant. This suggest that using the forecasting back in time is not as relevant as the future forecast which seems plausible. The forecasting back in time captures whether or not it has been accurate in the period before a disruption, which likely is more reasonable to indicate if a shortage will occur or not. Due to the delay

model being trained on data only consisting of disruptions which leads to a shortage, the information that the forecasting backwards brings can in that sense be deemed as irrelevant when it is already known that a shortage will occur.

6.3 Future Work

The most relevant development of the current findings would be to improve the performance of the duration model or in another way successfully predicting if a shortage will occur or not. One suggestion would be to develop a binary model that either classifies shortage or no shortage, and focus on high precision for the shortage class. The current delay model could then be used to give Arla insights and justify decisions based on statistics.

Arguably, one of the most important aspects of a statistical model is the dataset it is based upon. The data used in this thesis is limited to 1076 and 274 rows of data for the duration and the delay model respectively. By increasing the number of data points, more instances of each class would be added and could decrease the models sensitivity to outliers and bad data, which arguably is even more important in an unbalanced dataset. One way to increase the number of data points could be to use all production instances despite them not causing a shortage, this would however require an alteration in the data procurement and extend the assumption that shortages are caused by more factors than just the disruptions currently used, and thus widening the scope. This would most likely also be subject to the issue of imbalanced data.

Since there is no predefined link between a disruption and a shortage, the data is based on assumptions which are simplified and could deviate from reality. Another improvement is to put more extensive work on creating the data to successfully map a shortage to the right disruption. This could be done through qualitative analyses. The main focus of the thesis was not on the procurement of the data and therefore there are much room left for improvement.

The random forest classifier was used based on previous work for similar types of predictions but the model selection can be very dependent on the data. A further development could be to examine the performance using the features brought by this thesis with other models to find better suited ones.

Regarding the features, the ones used are based on intuition and expectations of what should affect the impact. Since the impact might be correlated to features that are not intuitively useful, one could increase the additional features with for example information about the products and examine if it would lead to an improved model.

Using SMOTE increased the performance for both models but they were still best at capturing the majority class which shows that imbalanced learning is challenging and using SMOTE does not remove the problem of imbalance. SMOTE is sensitive to noisy data so further developments could be to make a more thorough analysis of the data or use other oversampling techniques that might prove to be better suited for this data.

6.4 Conclusions

Looking at the test scores compared to the validation scores from the CV, one can see that the models are not clearly overfitted on the training data since the test scores were either close to the median or above. The duration model does not give results stable enough to presume it being accurate since the precision and recall is too low for the minority classes. Although, objectively the selected model is better than always assuming no shortage since the model does in fact predict shortages. Grouping the data resulted in more balanced datasets for the delay, this reflected on the test results where the improvement after using SMOTE was not as apparent as for the duration, which after grouping still had an imbalanced dataset.

The most important features for the duration model is `quotient` followed by `freq` and then `stock`. The two least important features are `fc_Bf` and `fc_F7`, but from the results of the the PFI those features are still considered important since a change in the model's performance is present. The most important feature for the delay model is clearly `freq`, and the least important one is `fc_Bf` which should be removed from the model to reduce complexity. The other features are about equally important according to the results from the PFI.

Implementing the models to the proposed framework in Figure 4.5.1 would not successfully meet the required expectations primarily due to the low performance of the duration model which determines whether there is a shortage or not. If this prediction is inaccurate, the result of the delay model is insignificant. On the other hand, the delay model itself can rather successfully predict the delay of a shortage if it is known that it will occur.

Bibliography

- [1] Alabbad, Dina A., Almuhaideb, Abdullah M., Alsunaidi, Shikah J., Alqudaihi, Kawther S., Alamoudi, Fatimah A., Alhobaishi, Maha K., Alaqeel, Naimah A., and Alshahrani, Mohammed S. “Machine learning model for predicting the length of stay in the intensive care unit for Covid-19 patients in the eastern province of Saudi Arabia”. In: *Informatix in Medicine Unlocked* 30 (2022). ISSN: 2352-9148. DOI: <https://doi.org/10.1016/j.imu.2022.100937>.
- [2] Arla Foods AB. *2022 Annual Report Arla Foods*. Apr. 2022. URL: <https://www.arla.com/company/investor/annual-reports/>.
- [3] Ayhan, Samet, Costas, Pablo, and Samet, Hanan. “Predicting Estimated Time of Arrival for Commercial Flights”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, 2018, pp. 33–42. ISBN: 9781450355520. DOI: 10.1145/3219819.3219874.
- [4] Chawla, Nitesh V, Bowyer, Kevin W, Hall, Lawrence O, and Kegelmeyer, W Philip. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.
- [5] Dong, Guozhu and Liu, Huan. *Feature engineering for machine learning and data analytics*. CRC Press, 2018.
- [6] Galarnyk, Michael. *Understanding Boxplots*. 2019. URL: <https://www.kdnuggets.com/2019/11/understanding-boxplots.html> (visited on 04/19/2023).
- [7] Han, Jiawei, Kamber, Micheline, and Pei, Jian. “3 - Data Preprocessing”. In: *Data Mining (Third Edition)*. Third Edition. The Morgan Kaufmann Series in Data Management Systems. 2012, pp. 83–124. ISBN: 978-0-12-381479-1. DOI: <https://doi.org/10.1016/B978-0-12-381479-1.00003-4>.
- [8] He, Haibo and Garcia, Eduardo A. “Learning from Imbalanced Data”. In: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), pp. 1263–1284. DOI: 10.1109/TKDE.2008.239.
- [9] James, Gareth, Witten, Daniela, Hastie, Trevor, and Tibshirani, Robert. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. URL: <https://faculty.marshall.usc.edu/gareth-james/ISL/>.
- [10] Kulkarni, Ajay, Chong, Deri, and Batarseh, Feras A. “Foundations of data imbalance and solutions for a data democracy”. In: *Data Democracy*. Academic Press, 2020, pp. 83–106. ISBN: 978-0-12-818366-3. DOI: <https://doi.org/10.1016/B978-0-12-818366-3.00005-8>.

- [11] Mekhaldi, Rachda Naila, Caulier, Patrice, Chaabane, Sondes, Chraibi, Abdelahad, and Piechowiak, Sylvain. “Using Machine Learning Models to Predict the Length of Stay in a Hospital Setting”. In: (2020). Ed. by Álvaro Rocha, Hojjat Adeli, Luís Paulo Reis, Sandra Costanzo, Irena Orovic, and Fernando Moreira, pp. 202–211.
- [12] Molnar, Christoph. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. 2022. Chap. 8.
- [13] Oliveira, Raphael Souza de, Reis Jr., Amilton Sales, and Sperandio Nascimento, Erick Giovani. “Predicting the number of days in court cases using artificial intelligence”. In: *PLOS ONE* 17 (May 2022), pp. 1–17. DOI: 10.1371/journal.pone.0269008.
- [14] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. *Cross-validation: evaluating estimator performance*. URL: https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation-evaluating-estimator-performance (visited on 05/20/2023).
- [15] Profeta, Martina, Maria Ponsiglione, Alfonso, Ponsiglione, Cristina, Ferrucci, Giuseppe, Giglio, Cristiana, and Borrelli, Anna. “Comparison of Machine Learning Algorithms to Predict Length of Hospital Stay in Patients Undergoing Heart Bypass Surgery”. In: *2021 International Symposium on Biomedical Engineering and Computational Biology*. Association for Computing Machinery, 2022. ISBN: 9781450384117. DOI: 10.1145/3502060.3503625.

