# Handelshøyskolen BI

## GRA 19703 Master Thesis

Final Thesis Master of Science 100%

### Predefinert informasjon

| | |
|---|---|
| **Startdato:** | 08-01-2024 09:00 CET |
| **Sluttdato:** | 01-07-2024 12:00 CEST |
| **Eksamensform:** | T |
| **Termin:** | 202410 |
| **Vurderingsform:** | Norsk 6-trinns skala (A-F) |
| **Flowkode:** | 202410\|\|11436\|\|IN00\|\|W\|\|T |
| **External assessor:** | External assessor 1 |
| **Internal assessor:** | Internal assessor 1 |

### Deltaker

| | |
|---|---|
| **Navn:** | Anders Paulsen Haugland og Harsh R. Trivedi |

### Informasjon fra deltaker

| | |
|---|---|
| **Tittel *:** | An analysis of Last-Mile Stop Delivery Times with Machine Learning |
| **Navn på veileder *:** | Jan Kudlicka |

| | | | |
|---|---|---|---|
| **Inneholder besvarelsen konfidensielt materiale?:** | Nei | **Kan besvarelsen offentliggjøres?:** | Ja |

## Gruppe

| | |
|---|---|
| **Gruppenavn:** | (Anonymisert) |
| **Gruppenummer:** | 81 |
| **Andre medlemmer i gruppen:** | |

# Master Thesis

BI Norwegian Business School

# An analysis of Last-Mile Stop Delivery Times with Machine Learning

Hand-in Date:

01.07.2024

Campus:

BI Oslo

Supervisor:
Jan Kudlicka

Examination code and name:

GRA 19703 Master Thesis

Programme:
Master of Science in Business Analytics

# ABSTRACT

In this thesis, we study the prediction of stop delivery times using multiple linear regression and machine learning models (Decision Tree, Random Forest, eXreme Gradient Boosting and Support Vector Regression). We analyse approximately 2.4 million deliveries over a two-year period using data from Posten Bring AS, OpenStreetMap, and the Norwegian Meteorological Institute, combining quantitative data with courier interview insights. Our methodology includes data preprocessing, model training with scikit-learn, and feature interpretation using permutation importance and SHAP values. We find that factors such as the number of mailboxes at a stop, the number of 'bag-on-door' deliveries, and the number of 'parcel in a mailbox' significantly increase predicted stop delivery times, indicating that higher delivery volumes will extend overall delivery times even if the number of stops remains unchanged. In contrast, temperature and precipitation appear to have minimal impact. We conclude that it is possible to predict accurate stop delivery times without knowing the true stop/driving times by setting one to a fixed value. Additionally, we conclude that the variability in delivery times across different units suggests the need for tailored strategies in urban versus suburban areas.

# AKNOWLEDGEMENTS

We would like to express our sincere gratitude to the following individuals for their invaluable contributions to this project.

First and foremost, we thank our supervisor, Jan Kudlicka, our many engaging, good conversations and brainstorming sessions. We appreciate his down-to-earth nature and his ability to view things from both practical and academic perspectives.

We are also grateful to Tarald Bjørdal Langan from Posten Bring AS. Tarald has supported us from the beginning with bi-weekly meetings, connecting us with various parts of Posten to obtain crucial data inputs. He also facilitated our interviews with Posten couriers at the distribution unit and provided valuable advice and domain knowledge throughout the project.

Our appreciation extends to Martin Klingenberg from ALV AS for his assistance with the Uvær API. Introduced to us through a mutual colleague, Martin was always very helpful and responsive whenever we had doubts related to extracting and interpreting weather data. We also extend our thanks to the Meteorological Institute for supplying the necessary weather data.

Finally, no text in this thesis has been generated or suggested using AI. We have used OpenAI (2024) to improve the text and to suggest grammatical or spelling corrections, and used our discretion to accept or reject any of the suggestions. We have also used OpenAI (2024) to suggest or improve the code in the computer programs used to conduct the research reported in this thesis.

# LIST OF ABBREVATIONS

| | |
|---|---|
| **AdaBoost** | Adaptive boosting |
| **ANN** | Artificial Neural Network |
| **API** | Application Programming Interface |
| **B2B** | Business to Business |
| **B2C** | Business to Consumer |
| **CART** | Classification and Regression Tree |
| **DT** | Decision Tree |
| **ExtraTrees** | Extremely randomized trees |
| **GPS** | Global Positioning System |
| **HDBSCAN** | Hierarchical Density-Based Spatial Clustering of Applications with Noise |
| **LMD** | Last-Mile Delivery |
| **MAE** | Mean Absolute Error |
| **MAPE** | Mean Absolute Percentage Error |
| **ML** | Machine Learning |
| **MSE** | Mean Square Error |
| **OLS** | Ordinary Least Squares **r**egression |
| **RF** | Random Forest |
| **RL** | Reinforcement Learning |
| **RMSE** | Root Mean Square Error |
| **SHAP** | SHapley Additive exPlanations |
| **SNR** | Signal to Noise Ratio |
| **SVR** | Support Vector Regression |
| **SVRP** | Stochastic Vehicle Routing Problem |
| **TSF** | Travel Speed Factor |
| **TSP** | Travelling Salesman Problem |
| **VRP** | Vehicle Routing Problem |
| **XGBoost** | eXtreme Gradient Boosting |

# Content

# 1 Introduction

## *1.1 Background and motivation*

In the changing landscape of logistics and delivery services, the efficiency and reliability of Last-Mile Delivery (LMD) operations are important for customer satisfaction and operational success (Rejikumar et al., 2020). Furthermore, , through a quantitative survey conducted in Sweden measuring participants' most recent e-retail experiences, Vakulenko et al. (2019) found that the LMD experience significantly impacts overall customer satisfaction with the entire e-retail process. This indicates that LMD is not only vital for recipient satisfaction but also affects business-to-business (B2B) customers, who can be directly influenced by the efficiency and reliability of LMD operations.

Accurate prediction of last-mile stop delivery times is essential for effective operational planning and scalability of LMDs. Hughes et al. (2019) examined the use of machine learning (ML) techniques to predict the duration of time delivery vehicles spend at each stop, using GPS data. They defined "stop delivery time" as the time required to distribute goods to their final destinations at each stop. This is the time we seek to distinguish and predict. Stop delivery time is also sometimes referred to as service time.

Our motivation stems from the growing complexity and variability inherent in LMDs (Giuffrida et al., 2022) . These final legs of the delivery process are often the most expensive, consisting of 41% of overall supply chain costs (Capgemini, 2019), and difficult to predict due to factors such as frequent stops, low drop sizes and urban traffic congestion, impacting overall service quality and operational efficiency. We recognise the need to predict and mitigate uncertainties in stop delivery times, thus ensuring more reliable delivery schedules and optimising resource allocation.

The motivation for this study is further fuelled by the ever-increasing demands of a connected world, where consumers expect rapid, accurate, cheap and eco-friendly options (*Wise Systems*, n.d.). In such an environment, delays become less tolerable, and the pressure to perform increases.

The thesis is conducted in collaboration with Posten Bring AS, a major postal and logistics group in the Nordic region, offering a range of postal and logistics

solutions. The company has over 12,600 employees and operates under two brands: Posten and Bring. Posten serves private individuals in Norway by delivering parcel and letters. Bring, meanwhile, provides logistics services for business customers in the Nordic countries and also serves private customers outside Norway (*Posten Bring AS - About Us*, n.d.).

Our collaboration with Posten Bring AS gives us access to real-world data and insights into the operational challenges and successes of a leading logistics provider. This collaboration also helps with understanding the practicalities of delivery logistics from the inside out, ensuring that the thesis is grounded.

### *1.2 Research problem and objectives*

This thesis focuses on predicting stop delivery times using ML models that use a variety of data, including delivery characteristics like location, parcel parameters, and stop details, as well as external factors such as weather and road travel time estimations. The objective is to accurately predict stop delivery times in different delivery contexts based on the data analysed.

Furthermore, the goal of this thesis is also to understand how each factor influences stop delivery times, providing useful information for improving routing, scheduling, and delivery reliability. By applying and evaluating various ML models, we aim to understand the prediction mechanisms and the impact of different factors on stop delivery times. Insight that can be important for enhancing routing efficiency, optimising scheduling, and ensuring delivery reliability, especially as delivery volumes increase and customer expectations grow.

### *1.3 Relevance/contribution*

The relevance of this thesis lies in its potential to significantly enhance the efficiency and reliability of LMD operations. By focusing on the prediction of stop delivery times, our work seeks to address a critical part of the delivery chain that directly impacts customer satisfaction and operational efficiency. While some research has been conducted in this field, our contribution is unique as it covers the specific context of Norway, using a new mix of variables and utilising detailed, real-world data that has not been studied before.

Theoretically, this thesis advances the application of ML in the field of logistics, particularly in the prediction of LMD times. By employing advanced predictive models, we deepen the understanding of the variables and their interactions that influence stop delivery times within a Norwegian context.

This partnership can helps Posten Bring AS understand their operations better, as well as understanding more about the potential of their data, which can be used to streamline delivery processes, optimise routes, and manage resources more efficiently. Additionally, the findings from this thesis can benefit other logistics companies involved in last-mile deliveries by providing useful insights to enhance their operations.

### *1.4 Scope and limitations*

We focus on Posten Bring AS' services: "Parcel delivery in a mailbox" and "bag on door". This is due to the relevance of these services in LMDs, and the fact that these services do not require actions from the recipient. Posten Bring AS has a standardised scanning procedure in place for these services, ensuring reliable data. Some of the delivery routes may include letters, which are unregistered deliveries that affect the stop delivery times. The lack of data related to letters is why we remove routes including letters in our scope.

Furthermore, we include only trips with at least 10 stops and assumed stop times ranging from 10 seconds to 5 minutes. Our analysis focuses exclusively on weekdays (Monday to Friday) and considers data from May 2022 to April 2024. We examine 23 distribution units across Norway, each responsible for the distribution of letters, parcels, and advertisements to private individuals and businesses in its local area. This thesis focuses solely on services for private individuals.

### *1.5 Structure*

The following chapters are organised as follows:

In Chapter Two, we do a literature review of existing research relevant to the topic of last mile delivery. We begin with an examination of traditional LMD methods, setting the context for understanding current practices and challenges. We then explore the application of ML in LMD, highlighting advancements and significant studies in the field.

Chapter Three outlines the ML techniques used in this thesis. We cover reinforced learning, unsupervised learning, and supervised learning, detailing the specific algorithms used, such as multiple linear regression, decision trees, support vector machines, random forests, and eXtreme Gradient Boosting. This chapter also describes the procedures for hyperparameter optimisation and model training.

Chapter 4 focuses on the data used in the thesis. Here we describe the data sources, including delivery data from Posten Bring AS, weather data from the Uvær API, and insights from interviews with Posten Bring AS LMD couriers. We then discuss the data preprocessing steps. More detailed information about the data can be found in the appendix.

In Chapter 5 we present and discuss the results of our analysis. This includes comparing model performances, and a closer look at how the models make their predictions.

The final chapter summarises the key findings of the thesis, discusses their implications, and suggests areas for future research.

# 2 Literature review

### *2.1 Traditional Last-Mile Delivery methods*

Bányai (2018) defines last-mile logistics in the context of the goods supply chain as: “the last stage of the supply chain.”. Gevaers et al. (2014) go more in-depth in its definition of last-mile logistics, where they define it as “the final leg in a B2C delivery service whereby the consignment is delivered to the recipient, either at the recipient's home or at a collection point.” We use this last definition of LMD throughout this paper.

The essence of logistics and, particularly, the concept of LMDs, is extensively analysed in the literature. Bosona (2020) provides a comprehensive overview, highlighting the critical role last-mile logistics play in the broader supply chain. It underscores the importance of efficiency in this final delivery leg, which often bears the highest costs and complexities. Furthermore, it highlights the rising demands of LMD, attributed to several key factors, including growing populations, increased urbanization, denser city planning, the global expansion of markets, the rise of e-commerce and omnichannel retailing strategies, and broader economic development. Serkan Özarık et al. (2022) further discusses the intricacies of managing these deliveries, especially in urban settings where factors such as congestion and accessibility pose significant challenges.

Liu et al. (2021) examined the use of delivery data from a food delivery service provider to enhance the punctuality of LMD services. They combined travel-time predictions with order-assignment optimisation, considering that the overall delivery time includes both travel time and the variable service time at customer locations. Their findings indicate that the prediction model is effective as it accurately reflects the driver’s routing behaviour while avoiding overly complex modelling. Additionally, they show that these prediction models can be effectively combined with stochastic and robust optimisation tools.

In an effort to come up with routes that incorporate the actual paths that drivers make due to navigational choices and preferences, Ghosh et al. (2023) proposed a data-driven model that effectively corresponds with the hierarchical nature of delivery data, where each stop is associated with a specific geographical zone.. they explained that initially, a sequence for the zones is established at a global level.

Following this, within zones, sequences of stops are determined such that, integrated with the predetermined zone sequence, a full solution is obtained. A similar approach may be applicable to estimating stop delivery times, by grouping stops into zones where travel times in-between groups of stops are estimated to be short.

Liu et al. (2021) determine the stop delivery time at each customer location by calculating the difference between the actual delivery time and the estimated arrival time. This estimated arrival time is derived from the departure time of the previous location and the projected travel time between the two locations obtained from the query. As travel time variability often increases with longer routes, Liu et al. (2021) account for this uncertainty by including an error term in a multiplicative model. To reduce variability in the travel time estimations, Muñoz-Villamizar et al. (2021) identified several variables related to traffic, parking availability, and urban density, that significantly impact delivery time. Weather, road conditions, and varying delivery windows further compound these challenges (Jenelius  Koutsopoulos 2013).

### *2.2 Machine Learning in Last-Mile Delivery*

Hughes et al. (2019) used GPS tracking to identify the stop delivery times and walking times from the stop to a client's location. They also considered historical data of actual number of deliveries in a stop, as well as data from OpenStreetMap. Using this data as their foundation, they proposed employing ML models to predict the duration of stop delivery times and determine if these times exceed a specific threshold.

Their approach included a two-tiered hybrid scheme: initially, a classification model is used, followed by either a regression or hazard duration model. They found that the best hazard duration model (log-normal) yielded more accurate results than sophisticated ML models when considering absolute errors, while ML models performed better with mean square errors. They also found that treating the prediction of stop delivery times as a classification problem, most ML models outperformed the best hazard duration model. Notably, the K-nearest neighbours model significantly outshone all others. Additionally, employing a two-tier model greatly enhanced prediction accuracy, primarily due to accurate initial classifications.

In the case of assembly services, Wolter Hanne (2023) tried to predict stop delivery times by looking more into factors that affect the efficiency of the stop. They discovered that Artificial Neural Networks (ANNs) outperformed traditional methods like multiple linear regression and support vector machines when considering variables such as furniture type, goods weight, and the experience of service technicians. While these methods may work well for longer stop delivery times, they are not faced with the challenge of generally short stop delivery times, as in the case of parcel in a mailbox delivery.

ML can be used to identify common patterns in deliveries and to determine the optimal route based on various parameters (Snoeck et al., 2020). Numerous studies have approached this problem using various ML algorithms (Giuffrida et al., 2022), applied in different fields such as multimodal transport (Servos et al., 2020) and across multiple sectors. Some research also extends to the use of neural networks for enhanced predictive accuracy (Mo et al., 2023).

From and Mangan (2023) proposed a hypothesis focusing on the use of ML analysis to identify and understand the impact of factors such as mileage and the number of parcels on LMD performance. Their ML analysis constructed predictive models to help planners determine optimal stop times and the travel speed factor (TSF) for route planning in Dispatch Track. From and Mangan (2023) used linear regression, Decision trees and Random Forest as their ML models and conducted data preparation like one-hot encoding and converting features to binary variables. Linear regression was selected for its simplicity and intuitiveness, while DT and RF were considered more suitable to identify the relationship in the dataset. While conducting descriptive analysis, they discovered complex relationships between the variables which made identifying strong patterns more difficult.

Servos et al. (2020) concentrated on predicting travel times for freight transports, noting that while ML is widely used in various fields, its application in multimodal transport has been limited. They used ML algorithms such as Adaptive Boosting (AdaBoost), Support Vector Regression (SVR), and Extremely Randomized Trees (ExtraTrees) due to their effectiveness with low data volumes and fast processing times. Their findings concluded that Support Vector Regression (SVR) provided the highest prediction accuracy, and that their model performed better than average-based approaches.

Giuffrida et al. (2022) provided a comprehensive and critical review of key studies and findings on last-mile logistics optimisation techniques. They examined traditional vehicle routing optimisation models for the Travelling Salesman Problem (TSP) and the Vehicle Routing Problem (VRP), as well as ML models and hybrid approaches. Their review concluded that the adoption of ML in logistics was on the rise, particularly with the prevalent use of supervised learning techniques. However, they did not specifically address the use of ML for stop delivery times, only noting that one of the Stochastic VRP (SVRP) variants includes VRP with stochastic service and travel times. Stochastic meaning "it incorporates uncertainty in some parameters whose value is not known, using random variables with a known probability distribution." (Giuffrida et al., 2022).

Puente-Mejia et al. (2022) analysed urban freight GPS data to better LMD planning. Their primary goal was to determine the stop times for each urban freight vehicle and categorise the stops as either traffic, delivery, or rest related. The authors use ML models such as K-Means and HDBSCAN in their study and try to demonstrate that it is possible to apply ML models that can group GPS stops by similar features. The authors concluded the paper claiming that the stop identification algorithm they created gives reliable data regarding vehicle stop time since it does not require additional parameters other than the GPS raw data.

# 3 Methodology

To address our objective of predicting stop delivery times and understanding the influence of various factors, we employ multiple ML approaches. In collaboration with Posten Bring AS, we identify and collect relevant and available data features that may impact stop delivery times. We then gather and preprocess this data to ensure it is suitable for analysis. See Chapter 4 for a more in-depth data description.

Next, we evaluate different branches of ML to determine which approach best fits our data and objectives. This involves exploring various algorithms and techniques to identify the most promising candidates for our predictive models. In 3.2 Model Selection, we present each model that we use in this thesis. After running these models on the pre-processed data, we compare their performance to find the most accurate predictor of stop delivery times.

To gain insights into how different features influence stop delivery times, we use various tools and techniques such permutation feature importance and Shapley Additive exPlanations to interpret the models. These can help us understand which factors are most significant, how they affect delivery times, and to what extent. We examine results covering all units, as well as deeper dives into specific units to better examine feature effects.

By following this methodology, we aim to provide actionable insights for improving routing, scheduling, and delivery reliability, especially in the context of increasing delivery volumes and growing customer expectations.

## *3.1 Machine Learning*

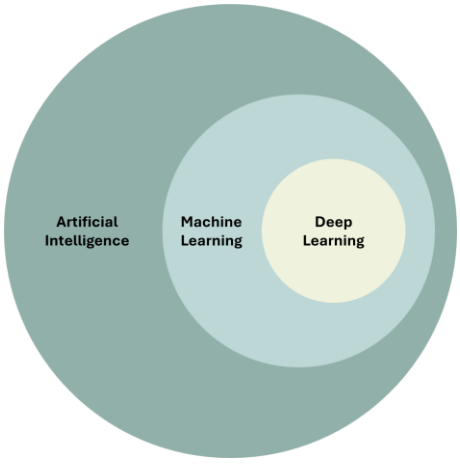



*Figure 1 Machine Learning as a subsection of Artificial Intelligence*

Oxford English Dictionary (n.d.) define Artificial intelligence (AI) as “The capacity of computers or other machines to exhibit or simulate intelligent behaviour; the field of study concerned with this.” Within AI, ML focuses on developing algorithms that improve through experience. According to Mitchell (1997), ML occurs when a computer program enhances its performance on certain tasks, measured by specific criteria, based on past data. This definition highlights the core of ML—systems that adapt and learn from previous examples, setting them apart from traditional static algorithms.

ML operates by constructing statistical models from data to solve practical problems. The process begins with gathering relevant data, followed by algorithmically analysing this data to identify patterns and make predictions (Burkov, 2019). This approach is applicable across various fields, leveraging both naturally occurring and artificially generated data sets, illustrating the adaptability and extensive application of ML techniques.

ML can be divided into three main categories: Supervised, Unsupervised, and Reinforcement learning. Supervised learning uses labelled data, while unsupervised learning works with unlabelled data. Reinforcement learning involves evaluative feedback instead of supervised signals. Supervised Learning can be divided into two types of problems: classification and regression. Classification problems generate categorical outputs, whereas regression problems yield numerical outputs (Burkov, 2019).

### *3.1.1 Reinforced Learning*

Reinforcement learning (RL) is a ML paradigm that involves an agent interacting with a dynamic environment to achieve a goal. It does so by programming the agent to learn through trial and error while receiving rewards and punishment, with the agent's goal being to maximise the benefits. According to Sutton and Barto (1999), RL is based on the combination of psychology's 'trial and error' and engineering's optimum theory.

In RL, the agent interacts with its environment through a cycle of perception and action, receiving feedback in the form of scalar reinforcement signals, which guide its learning process (Kaelbling et al., 1996). At each time step, the agent follows a policy ($\pi$) that assigns probability to actions depending on the current state. RL is

used to update this policy based on experiences to maximise cumulative rewards (Sutton and Barto, 1999). Key components of RL include a reward function—which offers unbiased feedback; the value function, which projects long-term rewards from various states, and the policy function, which leads the agent's actions (Kaelbling et al., 1996; Li, 2018). A balance between exploitation (selecting actions known to give high rewards) and exploration (performing new actions to discover their consequences) is fundamental for developing efficient and effective RL algorithms (Sutton and Barto, 1999).

While reinforcement learning is effective for autonomous decision-making and optimisation through interaction with an environment, it is not ideally suited for this thesis, which focuses on predicting stop-times. Reinforcement learning excels in scenarios requiring sequential decisions leading to a reward, but our problem is predictive rather than decision based. Thus, the RL approach, which relies on continuous feedback and long-term goal achievement, does not fit the structured, outcome-focused requirements of predicting stop delivery times.

### *3.1.2 Unsupervised Learning*

Unsupervised learning is a ML technique in which the system learns to discover patterns and structures in data without being explicitly programmed with correct answers or given direct feedback (Tyagi et al., 2022). This method is useful in situations where the data lacks labels or established categories, as it can help detect hidden patterns and intrinsic structures within the data (Li, 2018). Clustering, which groups data points based on their similarities is fundamental to supervised learning. Such grouping is useful in a variety of practical applications, including segmenting clients based on their purchasing activities without prior knowledge of customer categories. This strategy is useful not only for targeted marketing, but also for dividing enormous data sets into manageable and interpretable groups (Harmony et al., 2022a).

Beyond clustering, unsupervised learning includes approaches such as dimensionality reduction, which simplifies massive sets of data while retaining their important features (Tyagi et al., 2022). This is particularly effective for decreasing data complexity, improving visual analytics, and accelerating computing processes. Unsupervised learning methods evaluate data using a variety of similarity measures, adjusting to data formats ranging from numerical to categorical (Harmony et al.,

2022a). Unsupervised learning algorithms lay the framework for more advanced analytical tasks by learning how to represent data efficiently, making them essential tools in the field of data science and artificial intelligence. These methods also enable systems to make informed decisions without human intervention (Silva and Zhao, 2016; Tyagi et al., 2022).

In conclusion, while unsupervised learning is effective for pattern discovery and data reduction, it is not the best choice for predicting stop delivery times. This approach lacks the ability to directly correlate specific input features with predefined, labelled outcomes, which is essential for our predictive modelling goals.

### *3.1.3 Supervised Learning*

Supervised learning is an approach to ML that involves learning a system's input-output relationship from a set of paired input-output training samples (Q. Liu and Wu, 2012). It is the process of learning a relationship between a set of input variables (X) and an output variable (Y), and then using this relationship to predict outputs for new, unseen data. It is the most important methodology in ML (Cunningham et al., 2008).

This method is distinguished by its use of annotated training data, in which the system learns to predict outcomes by comprehending the relationship between input factors and their related outputs. Supervised learning includes two types of tasks: classification and regression (Jo, 2021). In supervised learning, the kind of task is determined by the nature of the target y. If y consists of discrete values from a fixed set of categorical outcomes (integers), it is a classification task. A regression task is one with continuous floating-point numbers (Nasteski, 2017).

In supervised learning, algorithms like decision trees, support vector machines (SVMs), and neural networks develop through exposure to a set of data examples—each tagged with a correct outcome (label). For instance, a decision tree makes decisions based on the values of input features, splitting data across its branches until a prediction is made at the leaves based on the most frequent outcome (Nasteski, 2017). SVMs on the other hand, aim to find a plane or hyperplane that best separates different classes in the input space. This task is framed as an

optimisation problem, with the goal of identifying the optimal plane that effectively divides the data into distinct categories (Harmony et al., 2022b).

### *3.2 Model Selection*

For this thesis, our task falls under the category of supervised learning since we are working with labelled data. Specifically, we use supervised regression models, as our target is a continuous numerical value—stop delivery times. We test four ML models to compare their performance on the data and to get more robust results. The ML models are Decision Tree, Random Forest, eXtreme Gradient Boosting and Support Vector Regression. We also test a multiple linear regression as a baseline to compare its performance with the ML models.

#### *3.2.1 Linear Regression*

Linear regression is a basic but important tool in the regression algorithm family. It helps identify relationships and dependencies among variables. This method models the relationship between a dependent variable $y$ and one or more independent variables, represented by $X$, using a linear equation to predict a continuous target variable (Nasteski, 2017). Unlike classification, which predicts categorical outcomes, linear regression focuses on continuous outcomes. The general mathematical model for multiple regression, involving a linear combination of input variables, is expressed as:

$$y = \beta 0 + \beta 1X1 + \beta 2X2 + \cdots + \beta nXn + \epsilon$$

Here, β0,β1, ...,βn represents the coefficients that the linear regression model aims to learn, and ϵ symbolizes the error term (Nasteski, 2017). Linear regression is classified as supervised learning since it uses labelled data to construct a linear relationship model between input and output. This model is then capable of making predictions based on new, unlabelled data (Burkov, 2019; Nasteski, 2017). By fitting a regression line to observed data, represented typically in a scatter plot, it minimises the discrepancies between the target values and predicted values, commonly through a loss function like mean squared error (Burkov, 2019). This loss function is critical to the model's capacity to generalise from training data to new data.

The use of linear regression extends beyond simply fitting a model to a dataset. In ML scenarios, particularly those containing many features, the model:

$$fw, b(x) = wx + b$$

assist in predicting an unknown y for a given x, where w represents a vector of weights and b is a bias term (Burkov, 2019). The goal of linear regression training is to determine the best weights for making the most accurate predictions. Its simplicity and quick computation make it a practical initial approach to understand the influence of various factors on stop delivery times.

### *3.2.2 Decision Tree*

Decision trees are a popular data mining tool that can be used to create classification or regression models. These models predict outcomes based on data inputs and are represented as a series of decision-making paths, with each decision leading to further subdivisions (Burkov, 2019; Song and Lu, 2015). A decision tree starts with a root node (the initial decision point) and branches off into several nodes representing subsequent decisions leading to final outcomes at the leaves (Burkov, 2019).

Each internal node of a decision tree examines a specific attribute and makes a decision based on a predetermined criterion, effectively splitting the data into more homogenous subsets (Nasteski, 2017; Song and Lu, 2015). This splitting continues until certain stopping criteria are met, such as a maximum tree depth or minimum node size set to prevent overfitting—a common problem in which the model becomes overly specialised to the training data, resulting in a low performance on new, unseen data (Song and Lu, 2015).

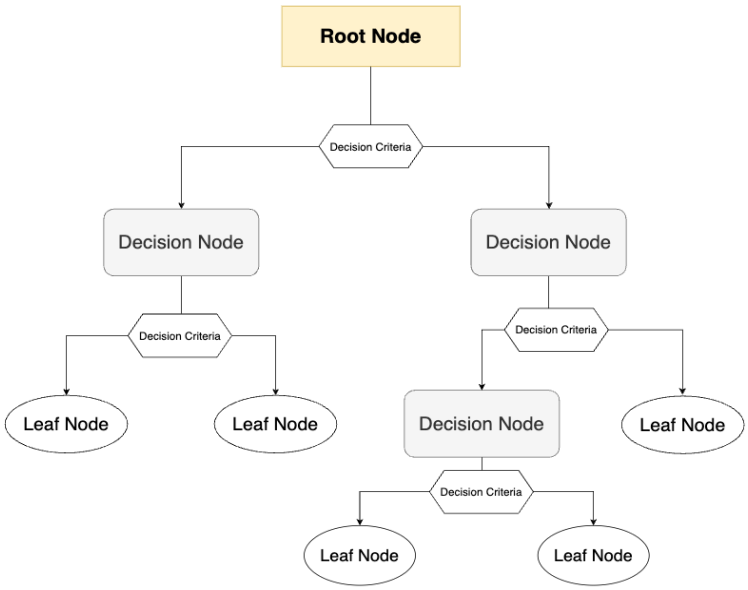



*Figure 2 Decision Tree Template*

In practice, decision trees are straightforward and simple to read, making them an effective tool for exploratory data analysis. They can handle both numerical and category data and effectively map non-linear relationships. The shape of a decision tree is established by algorithms such as ID3, C4.5, or CART (Classification and Regression Trees), which select the appropriate attribute to split on at each node depending on metrics such as information gain or Gini index (Nasteski, 2017; Song and Lu, 2015). These metrics aid in tree optimisation by ensuring that each split successfully contributes to accurately identifying or predicting the target variable.

Despite their simplicity, decision trees have some limitations. They are prone to overfitting and underfitting, particularly for complex trees, and when dealing with small datasets ( Song and Lu, 2015). Moreover, small changes in the data could result in a significantly different tree being constructed. This instability can be reduced by employing ensemble approaches such as Random Forests, which create many trees and aggregate their predictions to enhance accuracy and robustness (Song and Lu, 2015). The model can also be enhanced by applying techniques such as backtracking during the search for the ideal decision tree, however this may result in a longer build time (Burkov, 2019).

The combination of decision trees' ability to model complex, non-linear relationships between features and stop delivery time, along with their interpretability that allows us to identify the most influential features in predictions, makes decision trees suitable for our objective.

### *3.2.3 Random Forest*

Random Forests, a popular ensemble learning technique, combine numerous DTs to increase prediction performance and prevent overfitting by taking advantage of unpredictability and decision-making variety. As Breiman (2001) explained, ensemble approaches like RFs improve classification accuracy by enabling individual trees to vote for the most popular class, averaging out biases and spreading model variance among different trees.

Random Forests use bagging (bootstrap aggregating) and feature randomness to create a forest of decision trees. Bagging involves creating several subsets of the original training set using replacement, resulting in various training settings for each decision tree and lowering the risk of overfitting (Breiman, 2001). This

concept is expanded upon by selecting a random subset of attributes for each split inside a tree, with the goal of reducing correlation between individual trees in the forest. High correlation across trees in an ensemble can increase variance and susceptibility to data noise, limiting the ensemble's ability to generalise (Breiman, 1996; Dietterich, 1998).

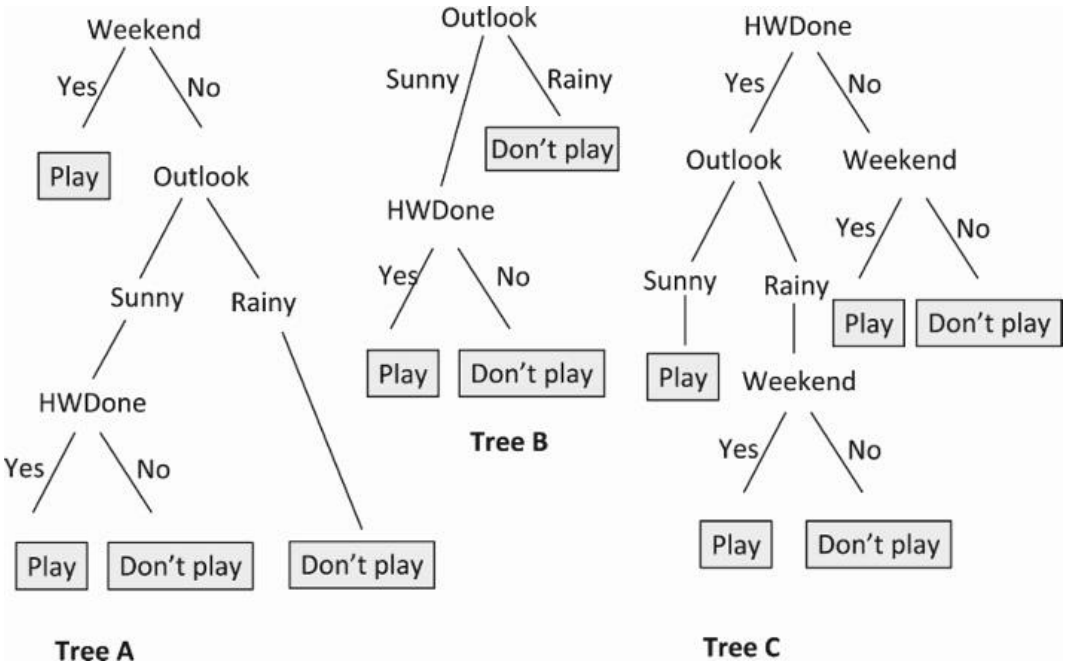



*Figure 3 Random Forest of three trees (Fawagreh et al., 2014)*

Breiman (2001) described that RFs for regression create trees based on a random vector, resulting in tree predictors $h(x, \theta)$ that yield numerical outputs instead of class labels. He explained that these numerical outputs assume the training set is independently sampled from the distribution of the random vector $(Y, X)$. Additionally, Breiman (2001) stated that the mean-squared generalisation error of any numerical predictor $h(x)$ is given by:

$$E_{X,Y}\left[\left(Y - h(X)\right)^2\right]$$

and that the random forest predictor is obtained by averaging the outputs of the individual trees $\{h(x, \theta)\}$.

Random forests can provide value to our thesis as it is a robust algorithm that does not tend to overfit while providing competitive results. RF also reduces the bias and can operate effectively on regression problems.

### *3.2.4 eXtreme Gradient Boosting*

eXtreme Gradient Boosting (XGBoost), is a supervised learning method that uses the boosting technique to produce accurate models. Boosting is an ensemble learning technique where multiple models are created sequentially, with each successive model aiming to correct the errors of the previous one. In the context of tree boosting, each new model added to the ensemble represents a decision tree (Chen and Guestrin, 2016).

XGBoost is an implementation of a generalized gradient boosting algorithm that has gained prominence due to its predictive performance, optimised multicore and distributed processing capabilities, and capacity to handle sparse data efficiently (R. Mitchell and Frank, 2017).

Based on the interpretation by R. Mitchell and Frank (2017) of the classic paper by Chen and Guestrin (2016), XGBoost is a generalised gradient boosting approach that incorporates a regularisation term to prevent overfitting and can handle any differentiable loss function. They also stated that instead of optimising the plain squared error loss, XGBoost defines an objective function with two components: a loss function over the training set and a regularisation term that penalises model complexity.

$$Obj = \sum_i L(y_i, \hat{y}_i) + \sum_k \omega(f_k)$$

$L(y_i, \widehat{y_i})$ is a convex differentiable loss function that calculates the difference between the predicted and true labels for each training event. The XGBoost algorithm defines $\omega(f_k)$ as the complexity of the tree $f_k$ (Chen and Guestrin, 2016).

$$\omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_j w_j^2$$

Where $T$ is the number of leaves in the tree; $f_k$ and $w_j$ are the leaf weights (i.e., the projected values stored at the leaf nodes). Incorporating $\omega(f_k)$ into the objective function optimises fewer complex trees that minimise $L(y_i, \widehat{y_i})$, leading to less overfitting. The user-configurable parameters $\gamma$ and $\lambda$ apply a constant penalty for each additional tree leaf and penalize excessive weights (R. Mitchell and Frank, 2017).

XGBoost is used in this thesis because of its robust performance and capacity to handle a diverse set of features. XGBoost's ability to handle sparse data, combined with its advanced regularisation methods, make it a good fit for our feature list. It attempts to maintain high predictive accuracy while avoiding overfitting, resulting in better estimates than previous models despite complex and variable data (Dhaliwal et al., 2018).

### *3.2.5 Support Vector Regression*

Support Vector Regression (SVR) is a type of Support Vector Machine (SVM), a supervised learning method that trains the model using a high-dimensional feature space (Basak et al., 2007a). SVR generates a model based on a subset of training data, as the cost function disregards training data close to the prediction within a threshold ε. It applies SVM to regression by introducing an insensitive tube around the function and optimising to find the flattest tube that balances model complexity and prediction error (Awad and Khanna, 2015). The hyperplane is defined by the model using support vectors (training samples from outside the tube), ensuring strong generalisation and accuracy (Awad and Khanna, 2015).

The goal of SVR is to find a function $f(x)$that deviates from the target values $y$ by no more than $\epsilon$ for all training data, while keeping the function as flat as possible. The linear function $f(x)$ is represented as:

$$f(x) = \omega \cdot x + b$$

Where $\omega \in \{R\}^n$ and $b \in \{R\}$, and $(\cdot)$ indicates the dot product in $\{R\}^n$ (Basak et al., 2007b).

To achieve this, it is necessary to minimise the Euclidean norm $||\omega||^2$. Formally, this can be expressed as a convex optimisation problem:

$$min \frac{1}{2} ||\omega||^2$$

Subject to the constraints:

$$y_i - (\omega \cdot x_i + b) \leq \epsilon$$

$$(\omega \cdot x_i + b) - y_i \leq \epsilon$$

This optimisation problem is feasible when $f$ exists and can approximate all pairs $(x_i, y_i)$ with $\epsilon$ precision.

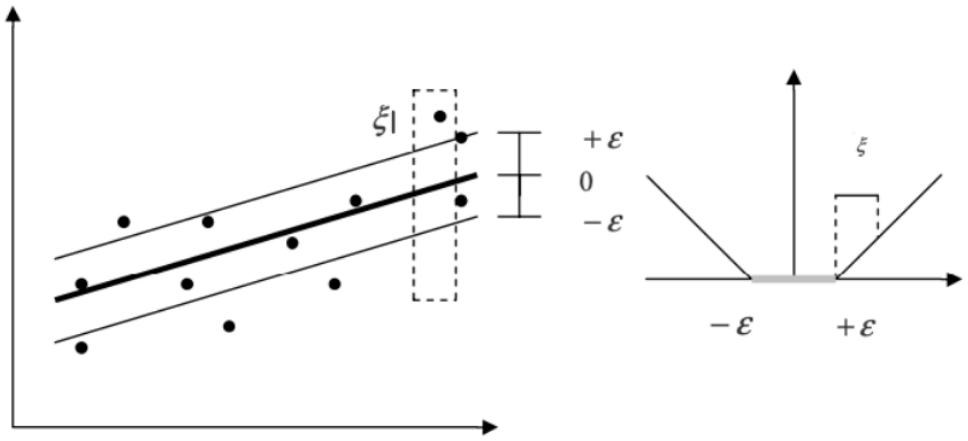



*Figure 4 SVR model trying to fit data (block line), with the thresholds of $+\epsilon$ and $-\epsilon$. The $\xi$ shows the slack variable deviating outside the epsilon margin. (Basak et al., 2007c)*

SVR is particularly useful for predicting stop delivery times when a non-linear relationship is suspected between inputs and the output. It effectively handles multiple continuous and categorical variables, making it suitable for complex environments such as varying weather conditions and diverse delivery routes.

### *3.3 Model Training*

We use scikit-learn, a public library available in python for model training and preprocessing. For both hyperparameter optimisation and model training, we split the data into a training set (80%) and a test set (20%). Additionally, all models are based on the same seed to ensure consistency across the models. Models at the unit level undergo their own hyperparameter optimisation and model training, based on datasets filtered by unit code. e focus on units 101792 and 102504 for a more in-depth analysis. These units are chosen due to the substantial amount of data available for each (see appendix D4). Additionally, they represent contrasting environments: unit 101792 is located in an urban area (Nydalen - Oslo), while unit 102504 is in a more suburban/rural area (Follo region).

#### *3.3.1 Data preprocessing for modelling*

Pedregosa et al. (2011) describe several important preprocessing tools for ML within the scikit-learn library. The ‘OrdinalEncoder’ converts categorical features into ordinal integers by taking an array-like input of integers or strings and transforming each feature into a single column of integers, ranging from 0 to number of categories – 1. The ‘OneHotEncoder’ converts categorical variables into binary (0 or 1) indicators for each category. The ‘StandardScaler’ standardises numerical features by removing the mean and scaling the values to unit variance.

Furthermore, the ‘ColumnTransformer’ allows different preprocessing steps to be applied to different subsets of features within the dataset (Pedregosa et al., 2011).

To ensure consistency in evaluation and comparison, each model is trained using the same data preparation and training process. Data preprocessing involves converting all categorical features into numerical codes using ‘OrdinalEncoder’ and ‘OneHotEncoder’, while continuous numerical features are standardised using ‘StandardScaler’. This preprocessing pipeline is implemented using the ‘ColumnTransformer’ from the scikit-learn library.

### *3.3.2 Hyperparameter optimisation*

Hyperparameter optimisation is a process to change the hyperparameters of a ML model to get the most optimal results. We perform hyperparameter optimisation by using random combinations of hyperparameters based on a set parameter grid to find optimal results. We compare the performance of the combinations using ‘cross_val_score’ from the scikit-learn library. The specific values tested for each parameter are detailed in appendix A.

Due to the computational intensity of the process, hyperparameter optimisation is done using a subset of the data to allow for testing a larger number of combinations (iterations). XGBoost, RF and DT use 10% of the training set, while SVR uses 3%.

We optimise the Decision Tree on parameters such as 'max_depth', 'min_samples_split', 'min_samples_leaf', and 'max_features'. According to Pedregosa et al. (2011), 'max_depth' refers to the maximum depth of the decision tree and will continue to expand until all leaves are pure or have fewer than the minimum samples required to split, unless specified otherwise. Without setting depth levels, the tree can expand excessively, leading to overfitting. They explain that 'min_samples_split' determines the minimum number of samples required to split a node, providing balanced splits, and reducing overfitting. Additionally, 'min_samples_leaf' specifies the minimum number of samples needed at a leaf node to ensure each leaf has sufficient data for reliable predictions (Pedregosa et al., 2011). Finally, 'max_features' refers to the number of features considered when looking for the best split, making the model less sensitive to noise and more robust (Pedregosa et al., 2011).

RF is a collection of DTs, meaning that the hyperparameters we tune are similar to those for decision trees. For parameters like 'max_depth', 'min_samples_split', 'min_samples_leaf', and 'max_features', we set values that reduce overfitting and ensure reliable predictions. Additionally, we tune two new hyperparameters: 'n_estimators' and 'bootstrap'. 'N_estimators' determine the number of trees in the forest, where trees generally make the model more stable and accurate by reducing variance (Pedregosa et al., 2011). 'Bootstrap' determines if bootstrap samples are used when creating trees. Using bootstrap sampling trains each tree on slightly different subsets of data, reducing variance and improving accuracy (Pedregosa et al., 2011).

XGBoost has various parameters classified into three major categories: general parameters, booster parameters, and task parameters. For our hyperparameter tuning, we focus on general parameters within the tree booster category. Pedregosa et al. (2011) outline several hyperparameters for tuning, including 'n_estimators', ‘max_depth’, 'learning_rate', 'min_child_weight', 'gamma', 'subsample', 'colsample_bytree', 'reg_alpha', and 'reg_lambda'.

According to Pedregosa et al. (2011), 'learning_rate' controls the contribution of each tree, with smaller values requiring more trees for the same performance level but increasing model robustness. They also define 'min_child_weight' as the minimum sum of instance weight needed in a child node, corresponding to the minimum number of instances in each node.

'Gamma' specifies the minimum reduction in loss necessary to create an additional partition on a leaf node of the tree, while 'subsample' denotes the fraction of samples used for training each tree, helping with generalization by reducing reliance on any single subset of training data. Additionally, 'colsample_bytree' is the fraction of features used for each tree, helping to reduce correlated trees and overfitting (Pedregosa et al., 2011). Lastly, 'reg_alpha' and 'reg_lambda' add L1 and L2 regularization terms on weights, respectively, which help make the model more conservative and interpretable, reducing overfitting (Pedregosa et al., 2011).

For SVR, we tune four key hyperparameters: C, epsilon, kernel, and gamma. The kernel parameter determines the type of kernel used in the method, with the default being 'rbf' (Pedregosa et al., 2011). They describe C as a regularisation parameter where lower values can cause underfitting, while higher values can cause

overfitting. Epsilon defines the epsilon tube where no penalty is associated in the training loss function, providing a margin of acceptance, where a larger epsilon can lead to underfitting and a smaller epsilon to overfitting (Pedregosa et al., 2011)

Gamma is the kernel coefficient for 'rbf', 'poly', and 'sigmoid' kernels, with a default value of 'scale', which adjusts gamma according to the data to make it more robust (Pedregosa et al., 2011) .

To assess the generalisation ability of the predictive models and to prevent overfitting, we use *k*-fold Cross-validation for the hyperparameter optimisation. In *k*-fold cross-validation, the training dataset is divided into *k* subsets of roughly equal size. The model is trained on $k - 1$ subsets, which, together, serve as the training set (Berrar, 2018). Then the model is tested on the remaining subset and the performance is evaluated. This is repeated for each split. The average performance of all splits equals the final performance measure of the model. To ensure that no groups (ie. same trips) are present in both the training and test sets within each split (iteration), we use the Group K-Fold version from the scikit-learn library.

To balance handling computational demand and achieving robust hyperparameter optimisation, we split the training data into five folds (see figure 5).

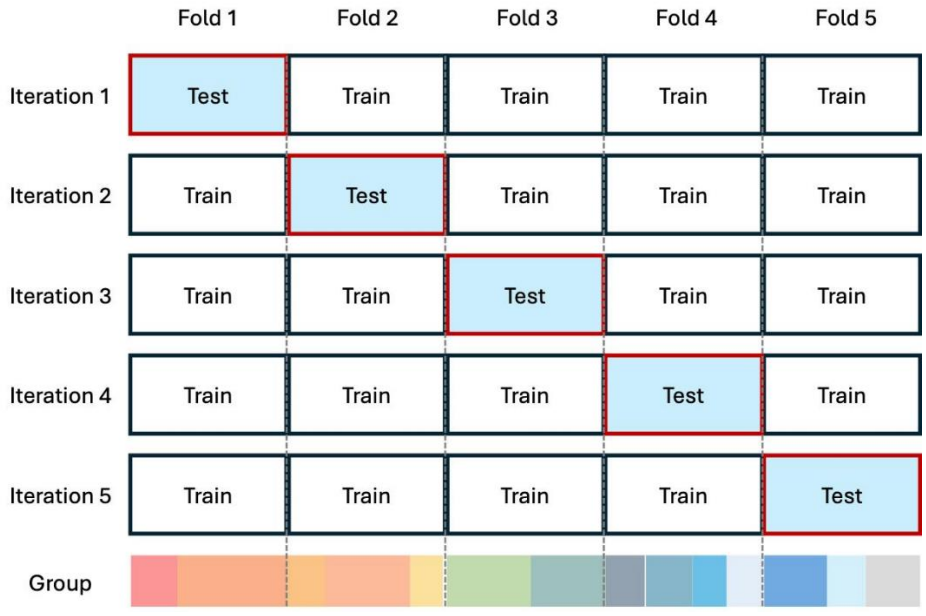



*Figure 5 Group K-Fold Cross-Validation*

In this thesis, we iterate through 200 random combinations for all ML models except for SVR. We limit the SVR model to 60 iterations due to the model being far more computationally expensive.

### *3.3.3 Training Procedure*

For training the models, we use the optimised hyperparameters for each algorithm and maintain the same 80/20 train-test split for consistency. The models are implemented as follows: XGBoost using xgboost. XGBRegressor, Decision Tree Regressor using sklearn.tree.DecisionTreeRegressor, Random Forest Regressor using sklearn.ensemble.RandomForestRegressor, and Support Vector Regression (SVR) using sklearn.svm.SVR. Each model is trained on the same preprocessed data, except for SVR, where we need to limit the training data to 15% of the training set due to computational constraints.

## ***3.4 Model Evaluation***

### *3.4.1 Performance Metrics*

Efficiency and accuracy are central to evaluating ML algorithms, as pointed out by Mohri et al. (2018). Besides computational considerations such as time and space, ML also focuses on sample complexity, which determines the amount of data needed for algorithms to effectively learn and perform. This emphasis on performance metrics reflects ML's integral role in enhancing computational applications, driving forward innovations across diverse industries. It provides a dynamic computational approach that bridges data-driven insights with practical problem-solving capabilities across various domains.

There are numerous ways to measure the performance of ML algorithms, particularly for regression tasks. Botchkarev (2019) summarised various surveys and identifies the most used metrics, which included Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

MSE measures the average squared difference between actual and predicted values, providing a sense of overall error magnitude. RMSE, the square root of MSE, expresses this error in the same units as the target variable, making it more interpretable. MAE calculates the average of the absolute differences between predicted and target values, highlighting the average error magnitude. MAPE, on the other hand, computes the average absolute percentage differences between actual and predicted values, offering a perspective on relative error.

While these metrics are widely used, they have all been scrutinised. The prevailing view is that no single metric is universally best for all situations (Botchkarev, 2019). This highlights the importance of considering the specific context when selecting appropriate performance measures.

We consider MSE, MAE and the coefficient of determination $R^2$ in our model evaluations. $R^2$ is a commonly used metric that is used to measure goodness of fit of any given model by interpreting the percent of the variance in the dependent variable accounted for by the independent variables (Gao, 2023). $R^2$ values range from 0 to 1, where $R^2 = 1$ implies a perfect a perfect relationship between the model and the data (Saunders et al., 2012), while $R^2 = 0$ indicates that the independent variable(s) cannot explain much variability for the dependent variable, making the model a poor fit.

### *3.4.2 Model Interpretation*

To better understand the ML models and how they reach their predictions, we use permutation feature importance to assess the overall impact a feature has on the model, and Shapley Additive exPlanations (SHAP) to get a better understanding of how each feature contributes to the model outputs. Galli (2024) explain that permutation feature importance assesses the decline in a model's performance when the values of a single feature are randomly shuffled, thereby breaking its relationship with the target. This method provides insights into how each feature contributes to the model's predictive performance (Pedregosa et al., 2011). SHAP on the other hand explains the feature impacts by giving each feature an importance value based on its contribution to a specific prediction (Lundberg and Lee, 2017).

# 4 Data

In this thesis, we analyse two years of data sourced from multiple providers, including Posten Bring AS, api.uvar.no, and OpenStreetMap.org. By integrating these data sources, we construct a comprehensive dataset that encompasses various factors relevant to LMDs. To gain a better understanding of last-mile logistics and relevant variables that may affect stop and driving times, we also conduct interviews with couriers at one of the distribution units within our scope.

## *4.1 Data Sources and Description*

### *4.1.1 Delivery data from Posten Bring AS*

We use six primary data sources from Posten Bring for a comprehensive analysis of delivery operations. These include consignment records, item-level data, delivery details, event logs, distribution routes, and stop information.

Two of these datasets, "consignment item" and "consignment" are very similar. The main difference is that "consignment item" provides specific information about parcel metrics, while "consignment" shows whether a consignment item is part of a larger consignment. According to the Cambridge Dictionary, a consignment is defined as "a quantity of goods that are sent together" (Cambridge University Press, n.d.). In this thesis, we focus on consignment items, as deliveries to a mailbox usually consist of a single delivery item rather than a batch of multiple delivery items. We use this data to obtain parcel metrics and filter by service type, ensuring we only look at similar delivery services.

The "delivery details" dataset contains the addresses where consignment items are to be delivered. These addresses are used to match the delivery data with the stop information data, connecting each address to a specific stop location. The addresses in these fields are filled in by the customer and are therefore prone to inaccuracies.

Event logs track the sequence of events associated with each consignment and consignment item. This includes detailed records from scanning events, such as the time and location of each scanning event, the employee handling the parcel, and the equipment used. Event logs also record the type of delivery, capturing whether it is a bag-on-door or a parcel-in-mailbox delivery.

Distribution routes provide data on which postcodes receive specific types of services on particular dates. This helps determine if a delivery trip includes letters, which are untraced and could affect stop delivery times. To ensure that deliveries only include traceable events such as parcel-in-mailbox or bag-on-door services, we use this dataset to filter out route and date combinations that include letters.

Stop information contains data about each relevant delivery location, such as connected addresses, the number of mailboxes, and estimates of rack types. The rack type is estimated to be a block if there are more than four mailboxes at a single address. If there are multiple unique addresses or four or fewer mailboxes, the rack type is estimated to be a stand. This estimation is not always accurate, with a typical error being a block wrongly estimated as a stand due to another address appearing at the same stop location.

### *4.1.2 Weather data from the Uvær API*

The weather data for this study is provided by Uvær, based on data from the Meteorological Institute and accessed via the api.uvar.no API.

Uvær aims to offer insights on where and how people should spend their time based on weather data, rather than predicting the weather itself. To provide this information, Uvær refines historical data from the Meteorological Institute, making it more accessible for specific locations.

The process of obtaining the data involves two steps. First, we retrieve location IDs by matching the distribution unit coordinates with the nearest weather measurement coordinates. Then, using these IDs, we obtain hourly data on temperature (in Celsius), precipitation amount (in mm), and weather type (e.g., rain, snow, cloudy). Additionally, we receive daily data on snow depth, measured using the snow map model by Saloranta (2014). To better understand recent snow conditions that could affect LMD times, we use the sum of precipitation from the current and previous two hours (when the weather type is snow) to create a new column named "recent snow."

Furthermore, we categorise all weather features into respective bins. This helps group similar levels of temperature, precipitation, and snow together to better understand their effects on delivery operations.

### *4.1.3 Travel time data from Posten Bring AS and Openstreetmap.org*

OpenStreetMap is an open-source, editable map covering the entire globe. Initiated in 2004 in the UK, the platform includes comprehensive data on roads, buildings, businesses, addresses, natural features and more (*What Is OpenStreetMap?*, n.d.). The project operates under the governance of the OpenStreetMap Foundation, managed entirely by volunteers (*What Is OpenStreetMap?*, n.d.).

We use modified data from OpenStreetMap.org, where Posten Bring AS has enhanced the dataset by adding extra measures, such as acceleration and deceleration, to improve travel time predictions. Unfortunately, we do not have access to historical traffic data to further improve predictions. The estimated travel times provided by Posten Bring AS are based on this enriched OpenStreetMap data.

The data consists of two datasets (see figure 6). The first dataset is an array that lists all the stop locations in a specific order. The second dataset is a many-to-many matrix that contains the estimated travel times, with each cell representing the travel time between two stop locations. In this matrix, the rows and columns correspond to the stop locations in the same order as they appear in the first dataset. This ensures that the x and y axes of the matrix align correctly with the row numbers in the first dataset. Using these two datasets, we incorporate the estimated driving times into the delivery data by referencing the previous and current delivery stop locations.

**Dataset 1**

| Location |
|---|
| Stop ID 1 |
| Stop ID 2 |
| Stop ID 3 |

**Dataset 2**

| | Stop ID 1 | Stop ID 2 | Stop ID 3 |
|---|---|---|---|
| **Stop ID 1** | 0 | 25 | 40 |
| **Stop ID 2** | 25 | 0 | 17 |
| **Stop ID 3** | 40 | 17 | 0 |

*Figure 6 Dataset 1: Array of locations, Dataset 2: Matrix of estimated driving times between locations.*

### *4.1.4 Interviews with Posten Bring AS Last-Mile Delivery Couriers*

To complement the quantitative data, we conducted qualitative interviews with couriers at one of the distribution units within our scope. These interviews aimed to gather insights into the practical challenges and variables that influence delivery driving and stop times. By asking drivers the general question, “What factors do

you believe affect your delivery driving and/or stop delivery times?", we identified common themes and factors that may not be evident from the quantitative data alone. The insights gained from these interviews help understanding the human and operational elements of last-mile logistics.

We asked 10 couriers and summarised their responses. The feedback from the couriers revealed several key factors impacting delivery efficiency. Weather conditions, particularly snow and cold temperatures, were frequently mentioned. Snow generally slows down delivery processes, especially for less experienced couriers, and reduces available parking spaces, leading to longer walks from the vehicle to the delivery point. Cold weather can affect electronic entrance doors, slow down the use of equipment, and necessitate frequent glove changes, further impacting efficiency. Additionally, devices may operate more slowly in cold temperatures, and extended exposure to cold, especially in villa areas, can cause couriers to become cold, slowing their pace.

Seasonality also plays a significant role. During summer, couriers often cover more routes in a day and are generally more motivated, working faster due to longer daylight hours and better weather conditions. Conversely, in winter, they tend to work slower due to shorter daylight hours and harsher working conditions.

Weekday variations were noted, with Mondays involving more advertisements, impacting the workload, while Fridays typically have fewer parcels, possibly due to more orders being placed over the weekend. Physical conditions of the couriers were also considered, with some reporting increased fatigue throughout the day, though most stated that their efficiency remains consistent.

Experience significantly affects delivery efficiency. Experienced couriers are more efficient, often knowing which parcels to bring first based on knowledge of mailbox capacities. They rarely change routes within a year, leading to better familiarity and efficiency. One courier noted that experience could mean a difference of hours in total route time for a day.

Traffic impacts are most significant when traveling to and from assigned routes, while it is not that significant in-between deliveries while on a route. Parcel characteristics also influence stop times. Large or numerous parcels at a single stop can extend stop times, particularly when multiple trips are needed. The "bag-on-door" service was repeatedly mentioned as causing longer stop times, with issues

such as missing identification on doors resulting in packages being redirected to distribution locations.

The type of vehicle used also impacts efficiency. The Paxster vehicle (a small electric ATV-like delivery vehicle) is most efficient in urban areas but can be challenging in snow. The vehicle is cold, which affects the courier's comfort and efficiency. Efficient parking is crucial and often facilitated by the Paxter, though snow and random factors, such as other cars parked in front of delivery points, can create additional challenges.

Delivery location further influences stop times. Outdoor deliveries are generally faster but may involve more walking, while indoor deliveries likely contain more mailboxes, potentially reducing stop times per package. Additionally, the floor of delivery, particularly in cases involving the "bag-on-door" service, impacts the time required, with higher floors typically taking more time.

Customer and recipient factors also play a role. Challenges include finding names on mailboxes, especially in large apartments, and dealing with incorrect or missing mailbox namings. Multiple packages for the same customer can lead to capacity issues with mailboxes.

Lastly, couriers typically take lunch halfway through their route, often driving back to the distribution point. These insights from the interviews provide a detailed understanding of the various factors that affect LMD times.

### *4.2 Data preprocessing steps*

In this section, we detail our data processing steps.

1. Primary Collection

We first gather primary delivery data from Posten Bring AS, covering “consignment”, “consignment item”, “consignment delivery detail”, and “consignment event data”. These include data from 01.05.2022 to 31.04.2024 to ensure seasonal representation and it being recent. Consignment and consignment item data, containing parcel-specific details, are filtered by service code to include only basic parcel-in-mailbox deliveries. Consignment delivery detail data is filtered by country to include only Norwegian deliveries. Consignment event data is filtered by event code for final delivery events and by unit code to include only relevant distribution units. User IDs are anonymised at this stage.

2. Data Filtering and Joining

Data is filtered prior to joining the tables to enhance process efficiency. The tables are joined using a unique consignment ID that is present in each table. Trips are identified, with each trip defined as all deliveries completed by a single courier in a day. Deliveries within each trip are sequentially numbered, and the total number of deliveries per trip is recorded.

3. Incorporation Distribution Routes Data

We incorporate Distribution Routes data to identify postcode and date combinations that include letters, which are untraced and could affect delivery times. We filter out all such deliveries, resulting in 41.7% of the total deliveries remaining.

4. Joining Stop Information

For the remaining deliveries, we join in the stop information, which is necessary for estimating travel times as the coordinates from stop info are used to identify the to and from locations. This process involves progressively modifying the delivery address. In the first step, we match delivery addresses with stop ID addresses, successfully matching 95.2% of all deliveries. When joining the tables, we match delivery addresses with addresses in the stop information table, requiring matching postcodes to avoid duplicates. This might not eliminate all duplicates, as some stop IDs share the exact same address. To avoid this, we only keep the first match. Since delivery addresses can be imprecise due to manual input by customers, we take further steps to match them with stop IDs.

5. Address Standardisation

For the remaining 4.8% of deliveries, we handle address variations by removing characters following a number in both the delivery and stop information tables. This standardises addresses like “Bygaten 12 H0303” or “Bygaten 15 0858 Oslo” to “Bygaten 12” or “Bygaten 15.” It also matches addresses where the official entry is “Bygaten 21”, but the customer input is “Bygaten 21A,” and vice versa. These adjustments help match another 1.3% of the remaining deliveries.

6. Distance-Based Matching

We then compare distances between coordinates using the Euclidean distance formula, which is sufficient for comparing shorter distances:

$$\sqrt{(delivery.coordX - stopID.coordX)^2 + (delivery.coordY - stopID.coordY)^2}$$

Since this distance comparison requires significant data storage, we initially restrict comparisons to delivery coordinates and stop IDs within the same postcodes. We calculate all distance combinations for each delivery and select the lowest Euclidean distance, only keeping the match if the Euclidean distance is 0.001 or less, equivalent to about 60 meters. This matches 1.4% of the remaining deliveries. For the final matching attempt, we remove the postcode requirement and instead match based on the same postal city, resulting in another 0.3% of all deliveries matched. Since stop information is essential for estimating travel times, the remaining 1.8% deliveries that are left unmatched are filtered out.

7. Adding Travel Times

Travel times are added by identifying previous stop locations for each delivery and finding travel times between stops using the travel time matrices provided by Posten Bring AS, based on their altered version of OpenStreetMap. We index the stop IDs from the arrays that list all stop locations in a specific order and add these indexes to the delivery data based on the previous and current stop IDs. These indexes, named “row_index” and “column_index”, are then used to find the corresponding travel time from the travel time matrix for the relevant distribution unit and stops.

8. Integrating Weather Data

Weather data is then joined with delivery data by time and location. We join in hourly weather data by the closest hour to each delivery event. Daily weather data (snow depth) is joined based on the day of delivery. Locations in the weather data match the distribution unit locations.

9. Adding Additional Features

Next, we add seasons, weekdays and stop delivery time assumptions. The first three are based on the delivery time. The stop delivery time assumption is calculated by taking the time between two scan events in a trip and subtracting the estimated travel time, as shown in the following equation:

$$A_i = T_i - E_i$$

Where A is the assumed stop delivery time, T is the actual time between two scan events, E is the estimated driving time and i is the delivery index.

10. Handling Delivery Gaps

We handle delivery gaps resulting from unmatched deliveries by adjusting the trip sequence and splitting trips with gaps into separate trips. We also remove trip IDs based on the combination of anonymised courier IDs and days by resequencing the trip IDs from 1 and onwards based on the final cleaned data.

11. Grouping deliveries

To analyse stop delivery times, we group sequential deliveries to the same stop ‘stop_id’ within each trip ‘trip_id’. For each group, ‘event_ts’ represents the timestamp of the last delivery event, ensuring consistency by marking the stop's end time. Weather data is based on the timestamp of the first delivery at a stop, as weather impacts the stop as it happens.

The ‘estimated_driving_time’ is calculated from the previous stop to the first delivery at the current stop. The ‘total_time_spent’ is the duration from the last delivery scan at the previous stop until the last delivery at the current stop. The ‘assumed_stop_time’ is derived by subtracting the estimated driving time from the total time spent at the stop. We also count deliveries at each stop with event codes ‘I6’ (bag on door) and ‘I9’ (parcel in a mailbox).

Additionally, we aggregate parcel volume by taking the maximum ‘max_volume_cm3’ volume of a delivery. Using the maximum volume highlights the impact of larger parcels, as averaging would diminish the significance of a single large parcel among multiple smaller ones at a stop.

12. Removing potentially unconventional trips

To avoid including unconventional trips (non-standard trips), we filter out trips that have less than 10 deliveries. We also filter out deliveries where the assumed stop time is less than 10 seconds or greater than 300 seconds. Very low values in assumed stop time can occur if the estimated driving time is significantly higher than the actual time it takes to drive between two locations, making the assumed stop time appear lower. We allow low values to indicate very low stop delivery times but limit it to 10 seconds to avoid including too many potential errors. Similarly, stops exceeding 300 seconds are likely data errors or due to unusual events during delivery. We also remove data for Saturdays and Sundays.

13. Creating Ordinal Features

For several of our features, such as estimated driving time, volume, temperature, precipitation, recent snow, and snow depth, we create ordinal equivalents (see appendix F). This involves grouping similar values into bins, like classifying rainfall as no rain, light, moderate, or heavy rain, and parcel volumes as small, medium, or large.

14. Incorporating Traffic Peak Time Indicators

We create a new feature called ‘rush_time’ to indicate whether a delivery was made during peak traffic times. We define rush time as 06:30 to 09:00 and 15:00 to 17:00, based on definitions from Norwegian toll authorities operating in cities such as Oslo, Bergen, Trondheim, Stavanger, and Tromsø (*Bommene i Trondheim*, n.d.; *Bypakke Bergen - Hva betaler du i bompenger?*, n.d.; *Takster og bomstasjoner*, n.d.; *This is Tenk Tromsø*, n.d.). We use this feature to help account for potential delays associated with higher traffic volumes during these periods.

### *4.3 Data Summary*

Using this combined dataset, we aim to cover as much relevant data as possible, where the relevant data we consider are primarily based on findings from Serkan Özarık et al. (2022), Wolter and Hanne (2023), Muñoz-Villamizar et al. (2021), and Jenelius and Koutsopoulos (2013) in combination with what we learn from our interviews with Posten Bring AS couriers and domain knowledge from our Posten Bring AS contacts.

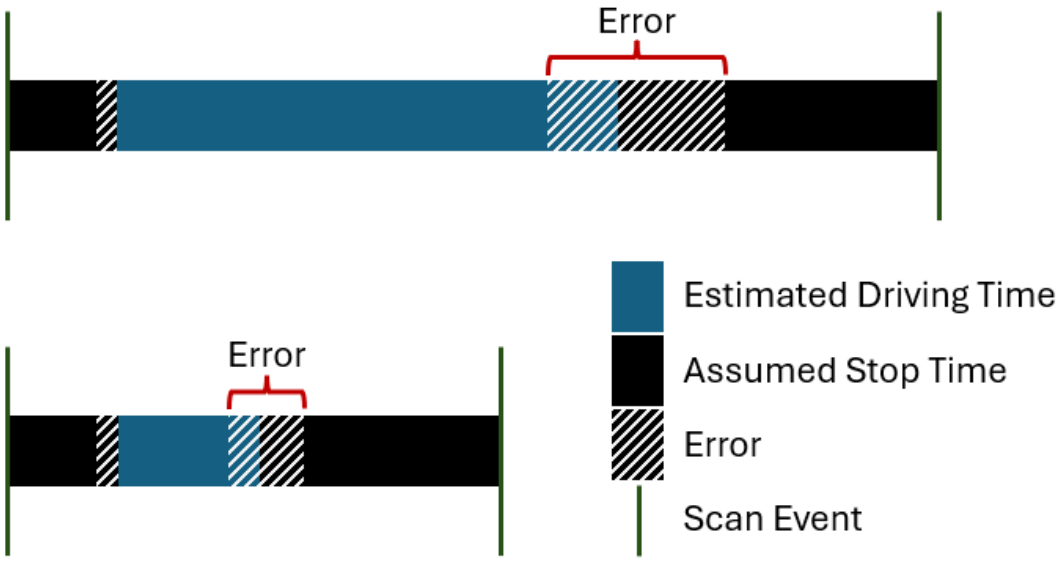



*Figure 7 Filtering Estimated Driving Time to reduce errors from driving*

Figure 7 illustrates the steps between each scan event. The total time between scans is the duration we analyse. We calculate our target value (assumed stop time) by subtracting the estimated driving time from this total time. This means that the assumed stop time for stop X is calculated as the time from leaving stop X-1 to the end of the delivery portion of stop X minus the estimated driving time. This means that our assumed stop time includes the exit time of the previous stop which we consider a fixed time in this thesis. We filter out deliveries where the estimated driving time is longer than one minute to minimise driving-related errors that can affect the stop delivery time estimates. After all data pre-processing, the data consists of 2,398,672 rows, each comprising of one stop delivery event.

# 5 Results and Discussion

## *5.1 Multiple Linear Regression*

| | | | |
|---|---|---|---|
| Dep. Variable: | assumed_stop_time | R-squared: | 0.136 |
| Model: | OLS | Adj. R-squared: | 0.136 |
| Method: | Least Squares | F-statistic: | 6014. |
| Date: | Sat, 29 Jun 2024 | Prob (F-statistic): | 0.00 |
| Time: | 00:12:34 | Log-Likelihood: | -1.0375e+07 |
| No. Observations: | 1913810 | AIC: | 2.075e+07 |
| Df Residuals: | 1913769 | BIC: | 2.075e+07 |
| Df Model: | 40 | | |
| Covariance Type: | HC0 | | |

| | coef | std err | z | P>\|z\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| const | 74.8227 | 0.302 | 247.540 | 0.000 | 74.230 | 75.415 |
| Estimated Driving Time | 13.7892 | 0.067 | 205.762 | 0.000 | 13.658 | 13.921 |
| Bag-on-Door deliveries | 36.3412 | 0.182 | 199.315 | 0.000 | 35.984 | 36.699 |
| Parcel in a mailbox deliveries | 15.9564 | 0.090 | 176.526 | 0.000 | 15.779 | 16.134 |
| Rush Time | 1.2071 | 0.117 | 10.347 | 0.000 | 0.978 | 1.436 |
| Block Stop | 17.2098 | 0.124 | 139.057 | 0.000 | 16.967 | 17.452 |
| Number of Mailboxes | 2.4507 | 0.034 | 72.239 | 0.000 | 2.384 | 2.517 |
| Temperature | -0.9280 | 0.098 | -9.499 | 0.000 | -1.119 | -0.737 |
| Precipitation | 1.0444 | 0.084 | 12.384 | 0.000 | 0.879 | 1.210 |
| Recent Snow | 2.2219 | 0.185 | 12.011 | 0.000 | 1.859 | 2.584 |
| Snow Depth | 0.6712 | 0.053 | 12.668 | 0.000 | 0.567 | 0.775 |
| Max Volume | 4.4510 | 0.073 | 60.697 | 0.000 | 4.307 | 4.595 |
| Unit 101367 | 2.6332 | 0.326 | 8.076 | 0.000 | 1.994 | 3.272 |
| Unit 101377 | -2.4532 | 0.303 | -8.087 | 0.000 | -3.048 | -1.859 |
| Unit 101434 | -0.9339 | 0.386 | -2.420 | 0.016 | -1.690 | -0.178 |
| Unit 101461 | -2.6227 | 0.335 | -7.840 | 0.000 | -3.278 | -1.967 |
| Unit 101609 | 3.9462 | 0.321 | 12.304 | 0.000 | 3.318 | 4.575 |
| Unit 101656 | 8.0378 | 0.344 | 23.356 | 0.000 | 7.363 | 8.712 |
| Unit 101792 | -10.9798 | 0.302 | -36.358 | 0.000 | -11.572 | -10.388 |
| Unit 101793 | -18.6968 | 0.293 | -63.764 | 0.000 | -19.272 | -18.122 |
| Unit 101799 | 1.3953 | 0.413 | 3.381 | 0.001 | 0.586 | 2.204 |
| Unit 101881 | 18.4947 | 0.407 | 45.434 | 0.000 | 17.697 | 19.293 |
| Unit 101882 | 16.1356 | 0.816 | 19.768 | 0.000 | 14.536 | 17.735 |
| Unit 101890 | 0.8970 | 0.383 | 2.343 | 0.019 | 0.147 | 1.648 |
| Unit 101892 | 8.0748 | 0.437 | 18.495 | 0.000 | 7.219 | 8.931 |
| Unit 101915 | 17.1785 | 0.359 | 47.882 | 0.000 | 16.475 | 17.882 |
| Unit 101930 | 3.8396 | 0.415 | 9.252 | 0.000 | 3.026 | 4.653 |
| Unit 102504 | 1.1577 | 0.325 | 3.561 | 0.000 | 0.521 | 1.795 |
| Unit 102666 | -13.2304 | 0.310 | -42.629 | 0.000 | -13.839 | -12.622 |
| Unit 106245 | 0.4493 | 0.348 | 1.290 | 0.197 | -0.234 | 1.132 |
| Unit 106530 | -6.4255 | 0.309 | -20.827 | 0.000 | -7.030 | -5.821 |
| Unit 107480 | 1.1743 | 0.353 | 3.329 | 0.001 | 0.483 | 1.866 |
| Unit 413706 | 5.3289 | 0.374 | 14.236 | 0.000 | 4.595 | 6.063 |
| Unit 413715 | -4.0054 | 0.326 | -12.290 | 0.000 | -4.644 | -3.367 |
| Tuesday | -7.0621 | 0.132 | -53.318 | 0.000 | -7.322 | -6.802 |
| Wednesday | -6.4458 | 0.132 | -48.854 | 0.000 | -6.704 | -6.187 |
| Thursday | -6.3734 | 0.135 | -47.312 | 0.000 | -6.637 | -6.109 |
| Friday | -6.6556 | 0.137 | -48.736 | 0.000 | -6.923 | -6.388 |
| Spring | -1.4279 | 0.113 | -12.631 | 0.000 | -1.649 | -1.206 |
| Summer | 0.0210 | 0.115 | 0.182 | 0.855 | -0.205 | 0.247 |
| Winter | 2.7822 | 0.137 | 20.319 | 0.000 | 2.514 | 3.051 |

*Figure 8 OLS Regression Results on the training set*

Figure 8 presents the baseline results of a traditional OLS Regression, serving as a benchmark for comparing our ML model outcomes. The resulting coefficients are based on standardised data, meaning that they should be interpreted accordingly. Based on the $R^2$ in the output, the model explains approximately 13.6% of the

variance in stop times, indicating a moderate fit. The F-statistic p-value of 0.00 confirms the model's statistical significance.

P-values are generally low, indicating the significance of most features. Notably, there are shorter stop delivery times on Tuesdays, Wednesdays, Thursdays, and Fridays, compared to Mondays. There is also substantial variation between unit codes, with up to 37 seconds expected difference between Unit 101793 and Unit 101881.

Winter has the longest expected stop delivery times, while Autumn and Summer are similar, though Summer is not statistically different from Autumn due to a high p-value. Spring is associated with the shortest stop delivery times.

Deliveries with larger parcels tend to take longer. The number of mailboxes extends the expected stop delivery time by 2.45 seconds for each standard deviation increase. Bag-on-Door deliveries significantly increase stop delivery time by about 36 seconds per standard deviation increase, while Parcel in a Mailbox deliveries add approximately 16 seconds per standard deviation increase. Block stops are estimated to take almost 14 seconds longer than non-block stops.

Estimated driving time primarily accounts for errors related to driving time between deliveries, indicating that errors increase with longer drives. Rush time has a low but significant effect. Recent snow significantly increases expected stop delivery times. Snow depth, precipitation, and temperature have low yet significant impacts, with rain, cold temperatures, and snow all contributing to longer expected stop delivery times. Overall, most predictors are statistically significant, underscoring their meaningful impact on stop delivery times.

### *5.2 Baseline Model Performance Comparison*

*Table 1 Baseline models' result on test set*

| Model | Mean Squared Error | Mean Absolute Error | $R^2$ |
|---|---|---|---|
| **XGBoost** | 2883.13 | 40.17 | 0.164 |
| **DT** | 2962.31 | 40.83 | 0.141 |
| **OLS Regression** | 2990.5 | 41.06 | 0.136 |
| **RF** | 2996.97 | 41.40 | 0.131 |
| **SVR (15%)** | 3151.41 | 38.55 | 0.097 |

Table 1 presents the baseline model comparison results on the test set, illustrating the performance of various models in predicting stop delivery times. The metrics evaluated include Mean Squared Error (MSE), Mean Absolute Error (MAE), and the R-squared ($R^2$) value.

The XGBoost model achieves the best performance among the models compared, with the lowest MSE of 2883.13, an MAE of 40.17, and an $R^2$ value of 0.164, indicating that it explains 16.4% of the variance in stop delivery times. The DT model follows, with an MSE of 2962.31, an MAE of 40.83, and an $R^2$ value of 0.141.

The OLS Regression model, serving as a traditional benchmark, records an MSE of 2990.5, an MAE of 41.06, and an $R^2$ value of 0.136. The RF model exhibits similar performance, with an MSE of 2996.97, an MAE of 41.40, and an $R^2$ value of 0.131.

The SVR model displays a higher MSE of 3151.41 but the lowest MAE of 38.55 among all models. This discrepancy suggests that the 15% sample likely had a slightly higher average stop delivery time (84.55) compared to the full dataset (84.31), affecting the model's overall performance. However, it has the lowest $R^2$ value of 0.097, indicating that it accounts for only 9.7% of the variance in stop delivery times.

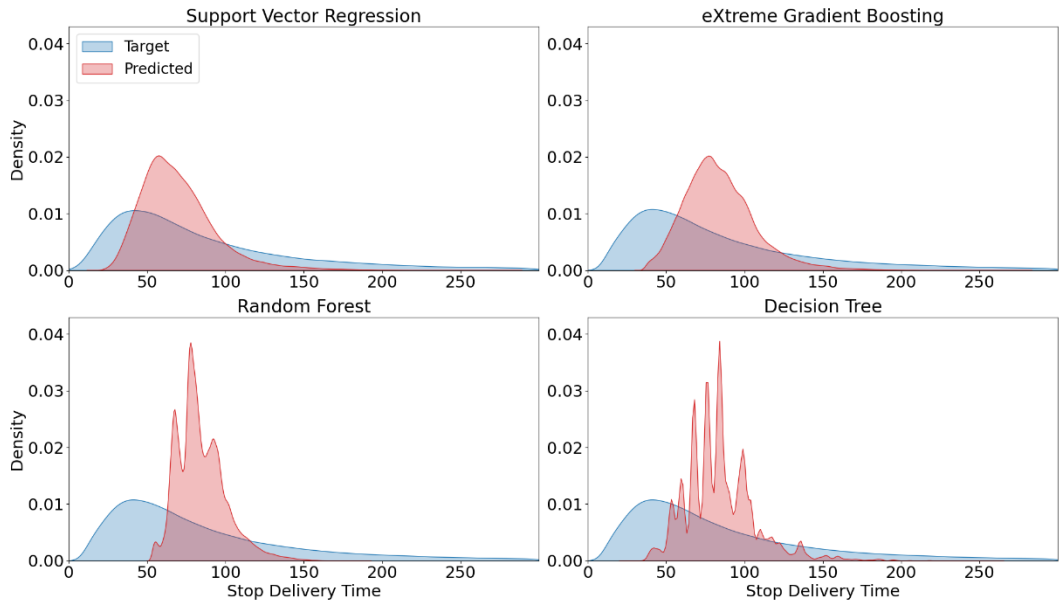



*Figure 9 Density plots of Target vs Predicted Stop Delivery Times*

Figure 9 presents the density plots comparing the target versus predicted stop delivery times for all four ML models. These plots provide a visual assessment of how well each model captures the distribution of stop delivery times.

As is evident for all four models, they all struggle with extreme target values at both ends of the spectrum. This can be expected based on the low $R^2$ values seen in Table 1, where low $R^2$ values can suggest that predictions will be more conservative, as the combinations of available features are not strong enough predictors to significantly influence the predictions. While SVR and XGBoost generally show a better overlap between target and predicted values, Random Forest (RF) and Decision Tree (DT) display a tendency to cluster predictions in certain regions. This clustering indicates that these models are fitting to a smaller set of features, with DT being the most extreme example.

It is important to note that a better overlap does not necessarily indicate a superior model. It could also mean that the model is underfitting the data, simply distributing predictions around the mean target values. Therefore, while XGBoost and SVR show closer alignment with the actual stop delivery times, this could partly be due to underfitting. Meanwhile, the distinct clustering observed in RF and DT suggests these models might be capturing specific patterns within a limited range of features.

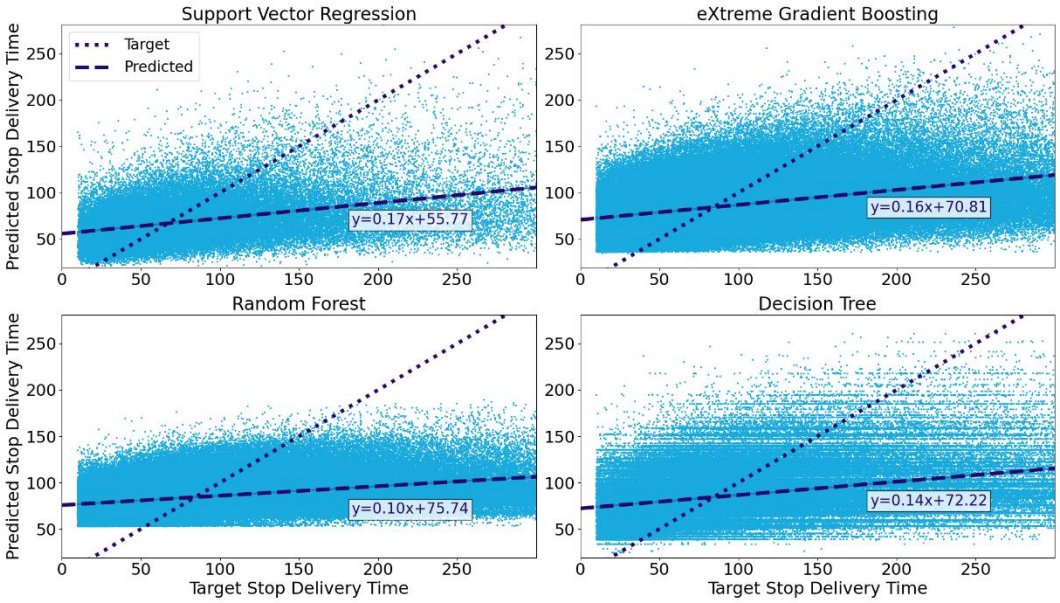



*Figure 10 Comparing Predicted vs Target Stop Delivery Times*

Figure 10 compares the predicted versus target stop delivery times for the four machine learning models. These scatter plots illustrate the relationship between the target and predicted values, providing insight into the models' predictive accuracy. It should be noted that SVR has fewer plots as it uses only 15% of the data due to computational constraints.

In all plots, the dotted line represents the perfect scenario where the predicted values align 100% with the target values. The solid line indicates the trend line of the model's predictions. A closer alignment of the trend line with the ideal line suggests better model performance.

Across all models, there is a noticeable deviation from the ideal line, particularly for higher stop delivery times. This deviation reflects the models' struggle to accurately predict extreme values, aligning with the low $R^2$ values previously observed. The trend lines for XGBoost and SVR show a slightly better fit compared to RF and DT, indicating their predictions are somewhat closer to the actual values. However, the slopes of these lines (0.16 for XGBoost and 0.17 for SVR) are still significantly below 1, suggesting underprediction of higher stop delivery times.

RF and DT models show the most substantial deviations, with slopes of 0.10 and 0.14, respectively. This indicates a higher tendency to underpredict, particularly at the higher end of stop delivery times. The consistent underprediction across all models highlights the challenge in capturing the variability and complexity of stop delivery times with the given feature set. The scatter plot for DT also shows some horizontal lines, confirming what Figure 9 indicated regarding distinct clustering at specific levels.

Despite these deviations, all models seem to capture some tendencies regarding longer stop delivery times. XGBoost is the most notable in terms of consistent increases in its predictions for deliveries with longer stop delivery times, showing a gradual rise in predicted values as the actual stop delivery times increase. This suggests that, while still underpredicting, XGBoost has a better grasp on the general trend of longer stop delivery times compared to the other models.

Overall, while all models demonstrate a reasonable alignment for lower stop delivery times, they consistently underpredict higher values, underscoring the need for a wider set of potential features, improved feature engineering or alternative modelling approaches to better capture the factors influencing stop delivery times.

## *5.3 Model interpretation*

### *5.3.1 Baseline level*

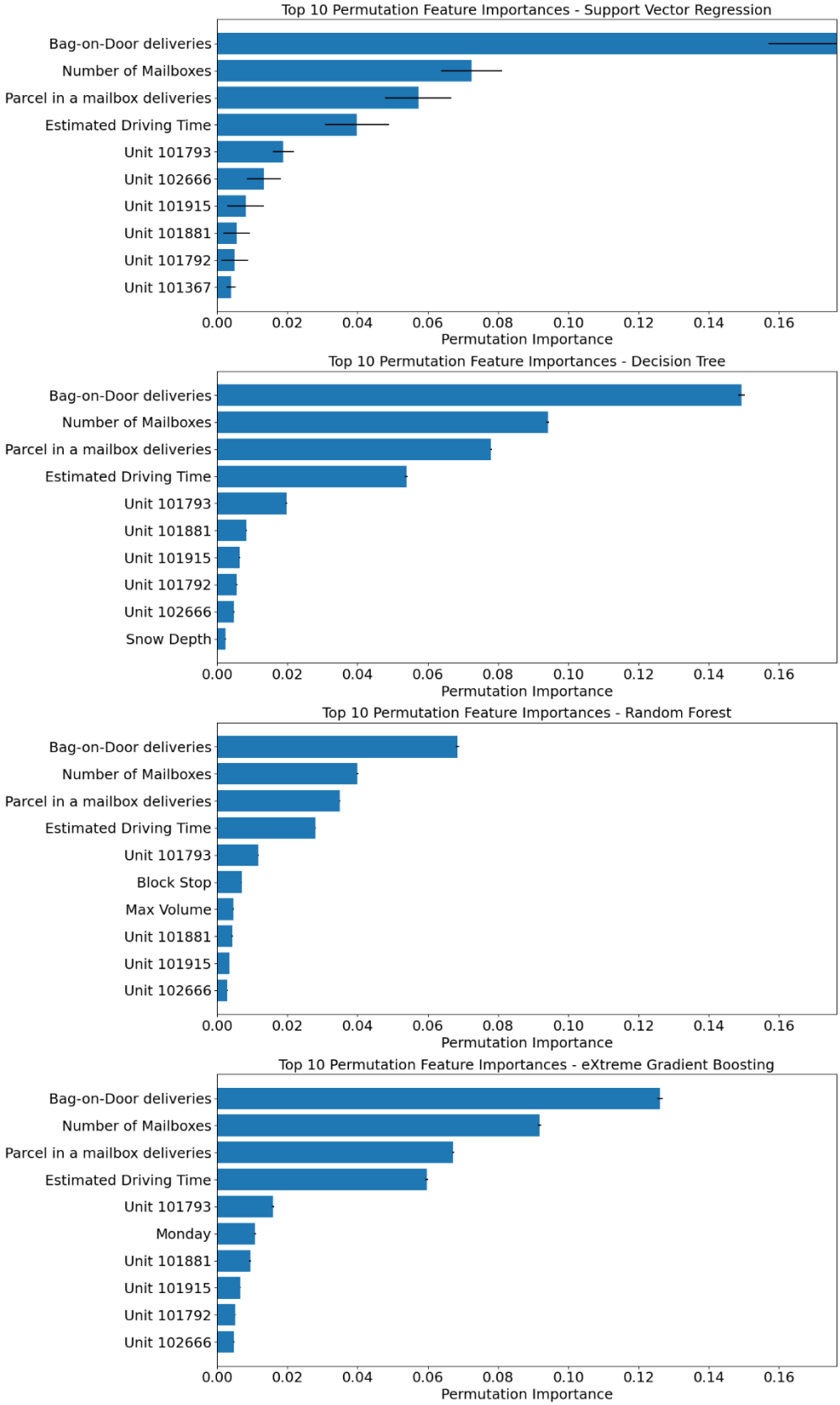



*Figure 11 Top 10 Permutation Feature importances for all machine learning models*

To better understand the models, we examine Figure 11, which plots the top 10 permutation feature importances for each machine learning model. Due to computational constraints, the SVR model is limited to 1000 samples. Across all four models, the number of Bag-on-Door deliveries consistently stands out as the most important feature. Following this, the number of mailboxes at a stop and the number of parcels in a mailbox are the second and third most important features in all models. This indicates that the number of mailboxes significantly affects stop delivery times and that the volume of deliveries at a stop has a substantial impact.

Estimated driving time ranks fourth in each model, suggesting some potential errors due to inaccuracies in driving time estimates. Alternatively, it could imply that estimated driving time, which correlates with the distance between stops, actually influences stop delivery time. This might be due to productivity changes from getting a rest while driving, or the other way around, where prolonged sitting reduces efficiency.

Unit codes also show some effect according to the feature importance. Notably, unit 101793 appears consistently across all models as the most impactful unit. Examining the summary statistics for units in appendix B2, we find that unit 101793 has the lowest average assumed stop time, which may explain why the models assign more importance to this unit.

Finally, the presence of Monday as a feature in the XGBoost models aligns with the findings from the baseline multiple regression model, where Monday was an outlier among weekdays. This supports the observations from interviews with Posten Bring AS couriers, who noted that Mondays tend to involve more advertisement deliveries and generally have a higher volume of deliveries, potentially leading to longer stop times.

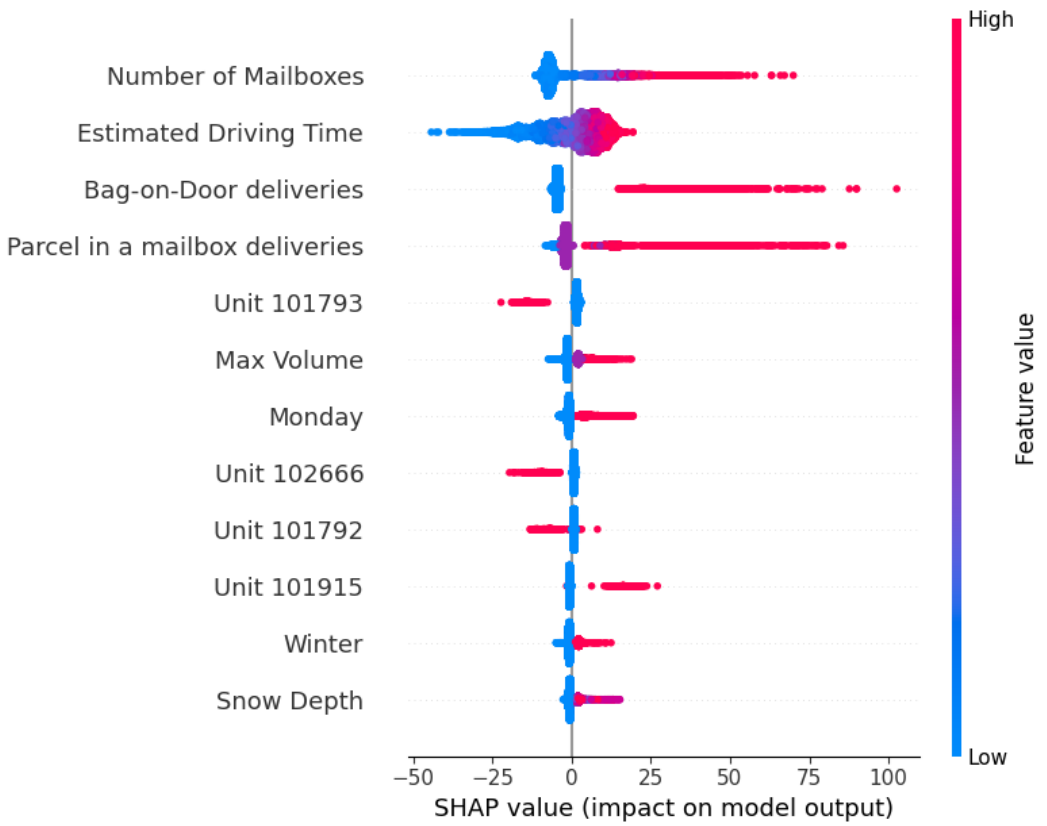



*Figure 12 Top 12 SHAP values for eXtreme Gradient Boosting (30 000 deliveries considered), sorted by impact on the model*

To understand in what way a feature potentially affects stop delivery times, we consider the best performing model, XGBoost, by assessing the SHAP output shown in Figure 12. Each dot in the figure represents an individual delivery, with SHAP analysing a total of 30,000 deliveries. The top 12 features are ordered by their mean absolute impact on the model. The colour gradient goes from blue (low value) to red (high value), indicating the feature values. See appendix G for an example of SHAP's interpretation of a single delivery.

The number of mailboxes and estimated driving times appear to be the top two features impacting the model's output. A low number of mailboxes at a stop generally seems to reduce the estimated stop delivery time by a small but consistent amount, while a high number of mailboxes has a much larger impact on the model's prediction. Similarly, low estimated driving times tend to reduce the model's output prediction, with extremely low driving times having a more significant negative impact than longer driving times. This could be partly due to the filtering of estimated driving times where we only include those up to one minute.

Next, it appears that delivery types, such as Bag-on-Door deliveries and Parcel in a Mailbox deliveries, both increase the predicted stop delivery time. This observation aligns with practical expectations. A higher number of Bag-on-Door deliveries seems to have a stronger effect on stop delivery times compared to a higher number of Parcel in a Mailbox deliveries. Additionally, the feature indicating Monday (shown in red when its value is 1) is associated with longer predicted stop delivery times, which aligns with feedback from Posten Bring AS couriers.

Furthermore, the presence of Unit 101793 in the data tends to lead to lower stop delivery times, which corresponds with the summary statistics in appendix B2, where deliveries from this unit have the shortest average assumed stop times. This pattern appears to apply to other units as well; units associated with shorter estimates typically have shorter average assumed stop times, whereas Unit 101915 has one of the longest average assumed stop times.

The SHAP analysis also suggests that larger parcels lead to longer predicted assumed stop times. However, this feature is strongly correlated with Bag-on-Door deliveries (see appendix E), a service that is primarily performed when parcels do not fit in the mailbox, indicating they are large. Regarding seasonality, winter is associated with longer predicted stop delivery times, similar to deeper snow, but both effects seem relatively low.

One possible explanation for why the number of mailboxes at a stop and the estimated driving time are considered more important in the SHAP analysis compared to Permutation Feature Importance could be that these features have high variability and uncertainty compared to features like the number of Bag-on-Door deliveries and Parcel in a Mailbox deliveries. Baudeu et al. (2023) found that SHAP values can be biased towards giving features with high entropy more weight, especially in low signal-to-noise ratio (SNR) conditions. This might be relevant in our thesis, where the top performing model only reaches an $R^2$ of 0.164, indicating the model does not capture much variability for the dependent variable. Soch et al. (2024) demonstrated that SNR can be derived from $R^2$ using the following formula:

$$SNR = \frac{R^2}{1 - R^2}$$

For the XGBoost model, this results in:

$$SNR = \frac{0.164}{1 - 0.164} = 0.196$$

Which can be considered a low SNR. Therefore, while these interpretations provide valuable insights into the factors influencing stop delivery times, they should be approached with caution. The low SNR indicates that the model struggles to distinguish the signal from the noise, potentially leading to more focus on features with higher entropy.

### *5.3.2 Unit level*

*Table 2 eXtreme Gradient Boosting performance on unit 102504*

| Model | Mean Squared Error | Mean Absolute Error | $R^2$ |
|---|---|---|---|
| **XGBoost** | 2553.70 | 38.65 | 0.122 |

*Table 3 eXtreme Gradient Boosting performance on unit 101792*

| Model | Mean Squared Error | Mean Absolute Error | $R^2$ |
|---|---|---|---|
| **XGBoost** | 2528.78 | 37.10 | 0.206 |

As a final analysis, we use XGBoost to train models with the same steps as those used for the baseline data, but with data filtered by units. Table 1 presents results for the less central distribution unit 102504, which covers the Follo region, while Table 2 shows results for the more central distribution unit located in Nydalen, Oslo.

We find that filtering by units leads to more accurate predictions and reduction in the variance of the data, especially for unit 101792, which has a MAE of 37.10 and $R^2$ of 0.206. This could partially be explained by the variance between different units, as well as the variance in the stop delivery times for the units.

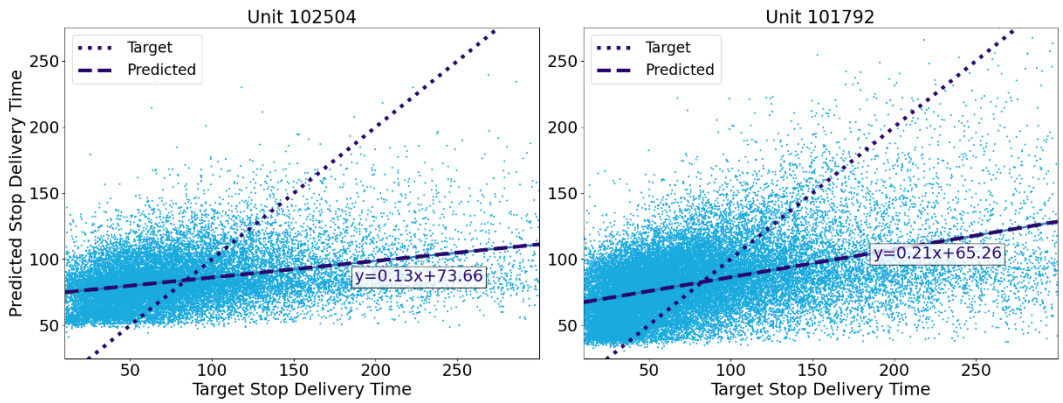



*Figure 13 Predicted vs Target Stop Delivery Times for unit 102504 and 101792*

Looking at Figure 13, we observe that the bulk of target stop delivery times for unit 101792 are generally low, in the region of 10-100 seconds. This reduced variance in the data could be one reason as to might explain why the model for this unit is performing better.

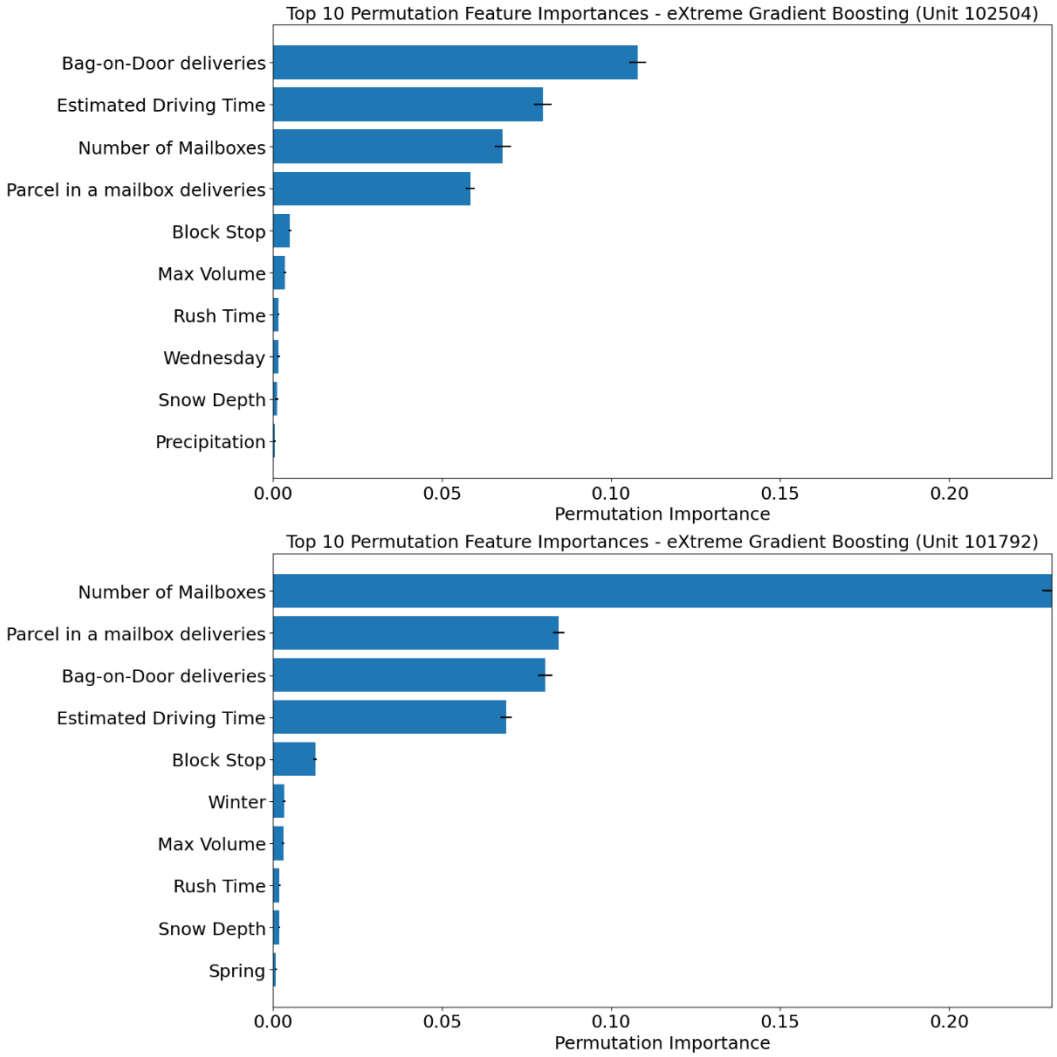



*Figure 14 Top 10 Permutation Feature Importances at a unit level*

In Figure 14, we see much of the same as those observed in the baseline models. A notable difference is that the number of mailboxes at a stop has a very high permutation importance for unit 101792. This unit, as seen in appendix B2, has the second highest mean number of mailboxes, which may explain the increased importance of this feature.

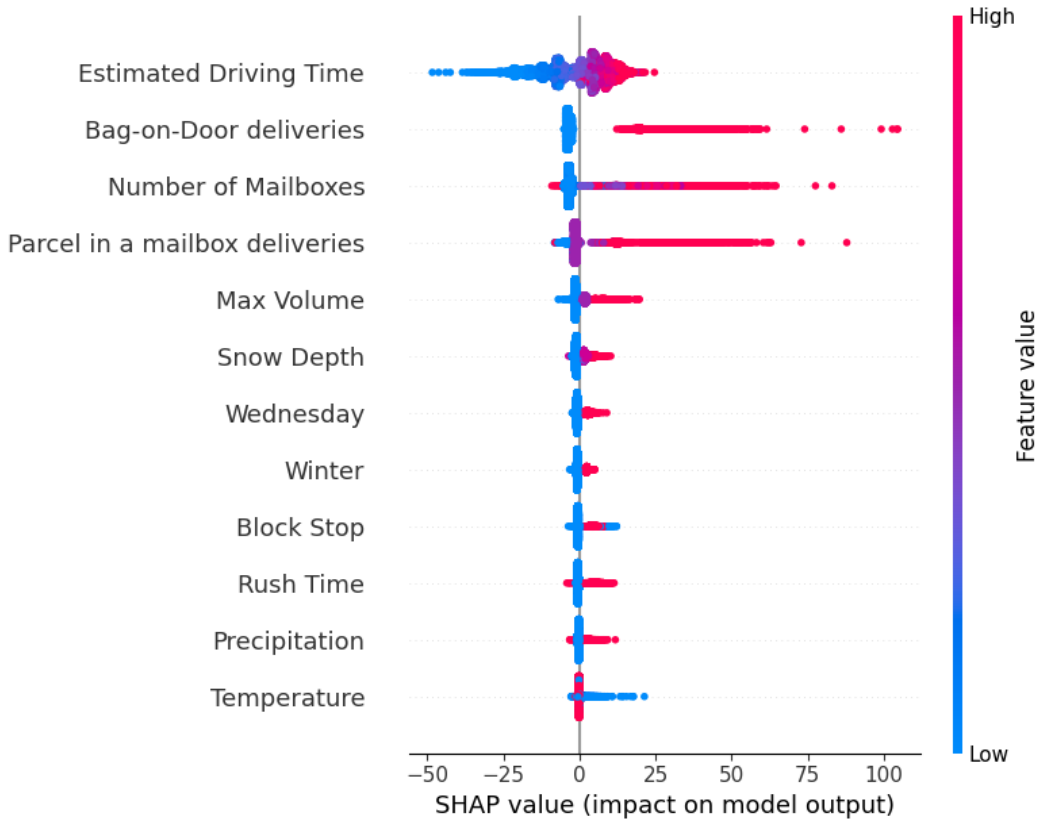



*Figure 15 Top 12 Features - eXtreme Gradient Boosting (unit = 102504)*

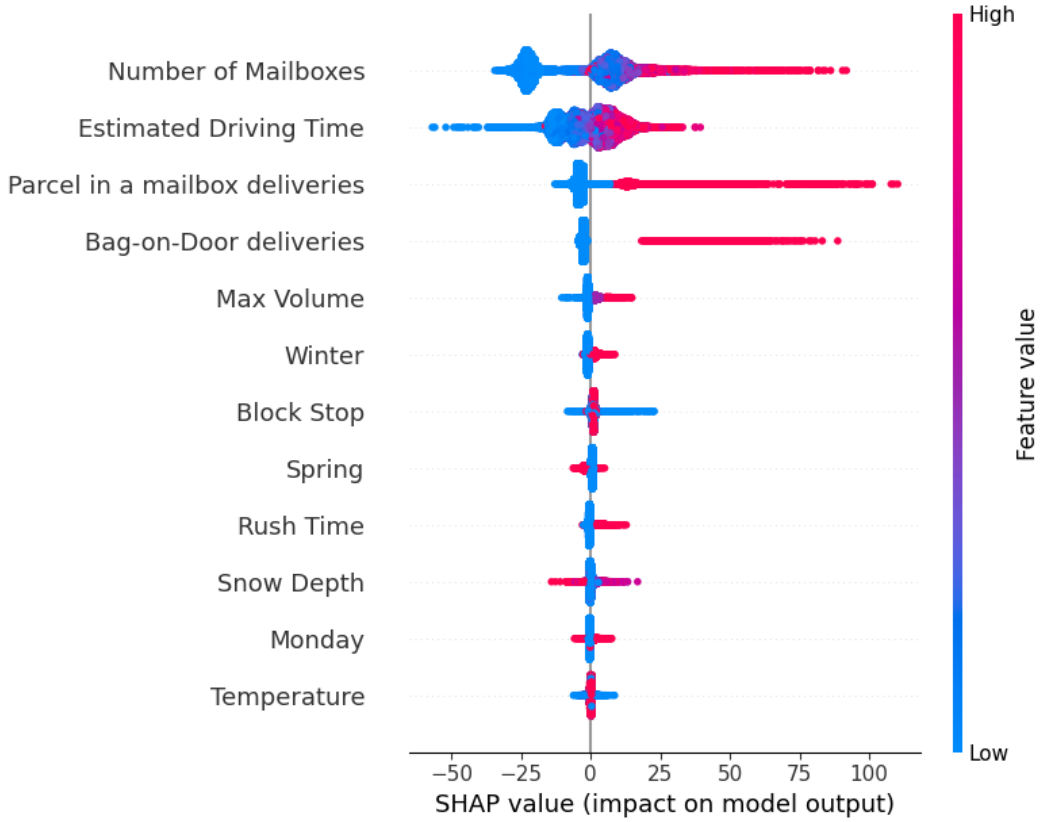



*Figure 16 Top 12 Features - eXtreme Gradient Boosting (unit = 101792)*

Figures 15 and 16 display the SHAP values for the units. Although the results are similar to the baseline model, they provide additional insights into features that might be overlooked in the baseline model. Interestingly, the output for unit 102504 shows that stop delivery times on Wednesday are predicted to be longer, in contrast to the consistent Mondays seen in previous results. This suggests that there might be unit-specific factors affecting delivery times on Wednesdays, such as different distribution schedules for advertisements.

For features with both high and low values on either side of the impact-neutral line, it indicates that the model struggles to accurately determine their effect on the predicted value. A concern regarding the top two features at the unit level is the potential for data leakage. This can occur if stop locations have a unique number of mailboxes or if there are consistent stop locations prior to them, leading to repetitive estimated driving times. While the general trend of the impacts of these features is clear, the overlapping low and high feature values on either side of the neutral SHAP impact line might suggest this issue. Another issue with using the number of mailboxes is how there could be a causal effect in which a stop with many mailboxes generally has a higher chance of getting multiple parcels during one delivery, we do however not see a strong correlation between the number of mailboxes at a stop and the number of deliveries of either parcel service.

# 6 Conclusion

In this thesis, we set out to predict stop delivery times using ML and to better understand how various factors influence these times. While we are unable to accurately predict stop delivery times, our analysis highlights several features with strong effects on them. Conversely, several factors included in the models have seemingly negligible impacts.

Our findings indicate that it is possible to predict stop delivery times without having true target values. However, there is significant variation in LMD stop delivery times, influenced by factors beyond the data we analyse in this thesis.

Notably, the number of mailboxes at a stop, the number of bag-on-door deliveries, and the number of parcel in a mailbox deliveries strongly increase the estimated stop delivery time. This suggests that increasing delivery volumes will likely lead to significantly longer stop delivery times, meaning trips will take longer, even if the number of stops remains the same. The findings also indicate that stop delivery times on Mondays tend to be longer, implying that advertisements included in deliveries on this day have a significant effect. This means that trips involving advertisements might require allocating more time than trips that do not.

Additionally, we observe considerable variation between different units, which may be related to factors such as the general accessibility of stop locations, vehicle standards, and delivery practices. This shows that for modelling purposes, units could be clustered together in similar groups based on different metrics such as practices, vehicle bay, geographical locations, and the composition of stop locations that they work with.

Factors such as temperature and precipitation were found to have negligible impacts on stop delivery times. This suggests that while weather conditions might intuitively seem important, their actual effect on the time it takes to complete deliveries appear to be minimal within the scope of our data.

We also find that ML models, particularly XGBoost, can outperform traditional multiple linear regression in predicting stop delivery times. Among the ML models tested, SVR is the least efficient due to its computational expense compared to tree-based models. However, the overall takeaway is that obtaining high-quality,

comprehensive data is more critical than choosing a particular model. Better data coverage could substantially help capture more of the variance in LMDs.

A significant limitation in this study is the lack of detailed and accurate data that could help explain the variance in stop delivery times. Estimated driving time consistently appears as a significant feature in the models, likely because the target value is generated based on the total time between two deliveries minus estimated driving time. This approach means that potential travel delays due to traffic, or other factors will contribute to inaccuracies in the target values.

Moreover, our data does not account for various important factors such as the floor of a delivery, whether the location is indoors or outdoors, courier experience, vehicle type, and walking distances. The absence of these data points limits the models' ability to capture the full range of variables that influence stop delivery times, resulting in a substantial amount of unexplained variance.

Future research could focus on collecting more precise and comprehensive data on driving and stop times to improve model accuracy. Developing a case study at one distribution unit, where there is a closer focus on potential variables that may affect stop delivery times, might be beneficial. This could also include more accurate data through methods of tracking true stop delivery times. Additionally, variations such as courier experience with a delivery route, parking availability, interactions with people, access to delivery points, clear delivery instructions, and walking distance are some factors we suggest that have the potential to affect stop delivery times. Addressing these and other factors in future studies could help enhance the predictability and reliability of stop delivery time models, providing more robust and actionable insights.

# References


Awad, M., & Khanna, R. (2015). Support Vector Regression. In M. Awad & R. Khanna (Eds.), *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers* (pp. 67–80). Apress. https://doi.org/10.1007/978-1-4302-5990-9_4

Bányai, T. (2018). Real-Time Decision Making in First Mile and Last Mile Logistics: How Smart Scheduling Affects Energy Efficiency of Hyperconnected Supply Chain Solutions. *Energies*, *11*(7), Article 7. https://doi.org/10.3390/en11071833

Basak, D., Pal, S., & Patranabis, D. (2007a). Support Vector Regression. *Neural Information Processing – Letters and Reviews*, *11*, 204.

Basak, D., Pal, S., & Patranabis, D. (2007b). Support Vector Regression. *Neural Information Processing – Letters and Reviews*, *11*.

Basak, D., Pal, S., & Patranabis, D. (2007c). Support Vector Regression. *Neural Information Processing – Letters and Reviews*, *11*, 206.

Berrar, D. (2018). *Cross-Validation*. https://doi.org/10.1016/B978-0-12-809633-8.20349-X

*Bommene i Trondheim*. (n.d.). Miljøpakken. Retrieved 15 June 2024, from https://miljopakken.no/bomsystemet

Bosona, T. (2020). Urban Freight Last Mile Logistics—Challenges and Opportunities to Improve Sustainability: A Literature Review. *Sustainability*, *12*(21), Article 21. https://doi.org/10.3390/su12218769

Botchkarev, A. (2019). A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management*, *14*, 045–076. https://doi.org/10.28945/4184

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. https://doi.org/10.1007/BF00058655

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Burkov, A. (2019). *The Hundred-Page Machine Learning Book*. Andriy Burkov.

*Bypakke Bergen—Hva betaler du i bompenger?* (n.d.). Ferde.no. Retrieved 15 June 2024, from https://ferde.no/bomanlegg-og-priser/bergen

Cambridge University Press. (n.d.). *Consignment*. Cambridge Dictionary. Retrieved 24 May 2024, from https://dictionary.cambridge.org/dictionary/english/consignment

Capgemini. (2019). *The last-mile delivery challenge*. Capgemini. https://www.capgemini.com/wp-content/uploads/2019/01/Report-Digital-%E2%80%93-Last-Mile-Delivery-Challenge1.pdf

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Cunningham, P., Cord, M., & Delany, S. (2008). *Supervised Learning* (pp. 21–49). https://doi.org/10.1007/978-3-540-75171-7_2

Dhaliwal, S. S., Nahid, A.-A., & Abbas, R. (2018). Effective Intrusion Detection System Using XGBoost. *Information*, *9*(7), Article 7. https://doi.org/10.3390/info9070149

Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, *10*(7), 1895–1923. https://doi.org/10.1162/089976698300017197

Fawagreh, K., Gaber, M. M., & Elyan, E. (2014). Random forests: From early developments to recent advancements. *Systems Science & Control Engineering*, *2*(1), 602–609. https://doi.org/10.1080/21642583.2014.956265

From, K., & Mangan, K. (2023). *Eliminating Last-Mile Inefficiencies in the Trucking Industry* [Master's Thesis, MIT]. DSpace@MIT. https://dspace.mit.edu/bitstream/handle/1721.1/126494/scm2020-from-eliminating-last-mile-inefficiencies-in-the-trucking-industry-capstone.pdf?sequence=1

Galli, S. (2024, January 11). *Understanding Permutation Feature Importance for Model Interpretation*. Train in Data's Blog. https://www.blog.trainindata.com/permutation-feature-importance/

Gao, J. (2023). R-Squared (R2) – How much variation is explained? *Research Methods in Medicine & Health Sciences*, 26320843231186398. https://doi.org/10.1177/26320843231186398

Gevaers, R., Van de Voorde, E., & Vanelslander, T. (2014). Cost Modelling and Simulation of Last-mile Characteristics in an Innovative B2C Supply Chain Environment with Implications on Urban Areas and Cities. *Procedia - Social and Behavioral Sciences*, *125*, 398–411. https://doi.org/10.1016/j.sbspro.2014.01.1483

Ghosh, M., Kuiper, A., Mahes, R., & Maragno, D. (2023). Learn global and optimize local: A data-driven methodology for last-mile routing. *Computers & Operations Research*, *159*, 106312. https://doi.org/10.1016/j.cor.2023.106312

Giuffrida, N., Fajardo-Calderin, J., Masegosa, A. D., Werner, F., Steudter, M., & Pilla, F. (2022). Optimization and Machine Learning Applied to Last-Mile

Logistics: A Review. *Sustainability*, *14*(9), Article 9. https://doi.org/10.3390/su14095329

Harmony, T., Carlier, M. E. M., & Hinojosa-Rodríguez, M. (2022a). *Neuroimaging techniques* (Section 6.2.3; p. 123). https://doi.org/10.1016/B978-0-12-820125-1.00012-9

Harmony, T., Carlier, M. E. M., & Hinojosa-Rodríguez, M. (2022b). *Neuroimaging techniques* (6.2.1.3; p. 115). https://doi.org/10.1016/B978-0-12-820125-1.00012-9

Hughes, S., Moreno, S., Yushimito, W. F., & Huerta-Cánepa, G. (2019). Evaluation of machine learning methodologies to predict stop delivery times from GPS data. *Transportation Research Part C: Emerging Technologies*, *109*, 289–304. https://doi.org/10.1016/j.trc.2019.10.018

Jenelius, E., & Koutsopoulos, H. N. (2013). Travel time estimation for urban road networks using low frequency probe vehicle data. *Transportation Research Part B: Methodological*, *53*, 64–81. https://doi.org/10.1016/j.trb.2013.03.008

Jo, T. (2021). Introduction. In T. Jo (Ed.), *Machine Learning Foundations: Supervised, Unsupervised, and Advanced Learning* (pp. 3–22). Springer International Publishing. https://doi.org/10.1007/978-3-030-65900-4_1

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, *4*, 237–285. https://doi.org/10.1613/jair.301

Li, Y. (2018). *Deep Reinforcement Learning: An Overview* (arXiv:1701.07274). arXiv. http://arxiv.org/abs/1701.07274

Liu, Q., & Wu, Y. (2012). *Supervised Learning*. https://doi.org/10.1007/978-1-4419-1428-6_451

Liu, S., He, L., & Max Shen, Z.-J. (2021). On-Time Last-Mile Delivery: Order Assignment with Travel-Time Predictors. *Management Science*, *67*(7), 4095–4119. https://doi.org/10.1287/mnsc.2020.3741

Lundberg, S., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions* (arXiv:1705.07874). arXiv. http://arxiv.org/abs/1705.07874

Mitchell, R., & Frank, E. (2017). Accelerating the XGBoost algorithm using GPU computing. *PeerJ Computer Science*, *3*, e127. https://doi.org/10.7717/peerj-cs.127

Mitchell, T. M. (1997). *Machine learning* (Nachdr.). McGraw-Hill.

Mo, B., Wang, Q., Guo, X., Winkenbach, M., & Zhao, J. (2023). Predicting drivers' route trajectories in last-mile delivery using a pair-wise attention-based pointer neural network. *Transportation Research Part E: Logistics and Transportation Review*, *175*, 103168. https://doi.org/10.1016/j.tre.2023.103168

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning* (2nd ed.). The MIT Press.

Muñoz-Villamizar, A., Solano-Charris, E. L., Reyes-Rubiano, L., & Faulin, J. (2021). Measuring Disruptions in Last-Mile Delivery Operations. *Logistics*, *5*(1), Article 1. https://doi.org/10.3390/logistics5010017

Nasteski, V. (2017). An overview of the supervised machine learning methods. *HORIZONS.B*, *4*, 51–62. https://doi.org/10.20544/HORIZONS.B.04.1.17.P05

OpenAI. (2024). *ChatGPT*. https://chatgpt.com/

Oxford English Dictionary. (n.d.). *Artificial intelligence| Oxford English Dictionary*. Retrieved 19 June 2024, from https://www.oed.com/dictionary/artificial-intelligence_n

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., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830.

*Posten Bring AS - About us*. (n.d.). Posten Bring. Retrieved 16 May 2024, from https://www.postenbring.no/en/about-us

Puente-Mejia, B., Suárez-Núñez, C., Calahorrano, D., Gavilanes, M., & Masaquiza, D. (2022). Machine Learning Applied to Last Mile Operations: Applying Machine Learning Models for Stops Classification in Urban Logistics. In J. Vargas Florez, I. de Brito Junior, A. Leiras, S. A. Paz Collado, M. D. González Alvarez, C. A. González-Calderón, S. Villa Betancur, M. Rodriguez, & D. Ramirez-Rios (Eds.), *Production and Operations Management* (pp. 501–512). Springer International Publishing. https://doi.org/10.1007/978-3-031-06862-1_38

Rejikumar, G., Krishnaraj, S., & Shini, S. (2020). An Exploration about the Last Mile Logistic Efficiency in Indian E-Commerce Sector- A Text Mining Approach. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3563089

Saloranta, T. M. (2014). Simulating more accurate snow maps for Norway with MCMC parameter estimation method. *The Cryosphere Discussions*, *8*(2), 1973–2003. https://doi.org/10.5194/tcd-8-1973-2014

Saunders, L. J., Russell, R. A., & Crabb, D. P. (2012). The Coefficient of Determination: What Determines a Useful R 2 Statistic? *Investigative Ophthalmology & Visual Science*, *53*(11), 6830–6832. https://doi.org/10.1167/iovs.12-10598

Serkan Özarık, S., da Costa, P., & Florio, A. M. (2022). *Machine Learning for Data-Driven Last-Mile Delivery Optimization* (SSRN Scholarly Paper 4012376). https://doi.org/10.2139/ssrn.4012376

Servos, N., Liu, X., Teucke, M., & Freitag, M. (2020). Travel Time Prediction in a Multimodal Freight Transport Relation Using Machine Learning Algorithms. *Logistics*, *4*(1), Article 1. https://doi.org/10.3390/logistics4010001

Silva, T. C., & Zhao, L. (2016). Machine Learning. In T. Christiano Silva & L. Zhao (Eds.), *Machine Learning in Complex Networks* (pp. 71–91). Springer International Publishing. https://doi.org/10.1007/978-3-319-17290-3_3

Snoeck, A., Merchán, D., & Winkenbach, M. (2020). Route learning: A machine learning-based approach to infer constrained customers in delivery routes. *Transportation Research Procedia*, *46*, 229–236. https://doi.org/10.1016/j.trpro.2020.03.185

Soch, J., Proofs, T. B. of S., Maja, Monticone, P., Faulkenberry, T. J., Kipnis, A., Petrykowski, K., Allefeld, C., Atze, H., Knapp, A., McInerney, C. D., Lo4ding00, & amvosk. (2024). *StatProofBook/StatProofBook.github.io: StatProofBook 2023* (Version 2023) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.10495684

Song, Y., & Lu, Y. (2015). Decision tree methods: Applications for classification and prediction. *Shanghai Archives of Psychiatry*, *27*(2), 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044

Sutton, R., & Barto, A. (1999). Reinforcement learning. *Journal of Cognitive Neuroscience*, *11*, 126–134.

*Takster og bomstasjoner*. (n.d.). Retrieved 15 June 2024, from https://www.fjellinjen.no/bompenger

*This is Tenk Tromsø*. (n.d.). Tenk Tromsø. Retrieved 15 June 2024, from https://tenktromso.no/tenk-tromso

Tyagi, K., Rane, C., Sriram, R., & Manry, M. (2022). *Unsupervised learning* (pp. 33–52). https://doi.org/10.1016/B978-0-12-824054-0.00012-5

Vakulenko, Y., Shams, P., Hellström, D., & Hjort, K. (2019). Online retail experience and customer satisfaction: The mediating role of last mile delivery. *The International Review of Retail, Distribution and Consumer Research*, *29*(3), 306–320. https://doi.org/10.1080/09593969.2019.1598466

*What is OpenStreetMap? | OpenStreetMap*. (n.d.). Retrieved 18 May 2024, from https://welcome.openstreetmap.org/what-is-openstreetmap/

*Wise Systems*. (n.d.). Wise Systems. Retrieved 20 May 2024, from https://www.wisesystems.com/blog/how-customer-expectations-in-last-mile-delivery-have-evolved/

Wolter, J., & Hanne, T. (2023). Prediction of service time for home delivery services using machine learning. *Soft Computing*. https://doi.org/10.1007/s00500-023-09220-7

# Appendices:

## *Appendix A – Hyperparameters*

*Table A.1: Support Vector Regression (3% data, 60 iterations)*

| Parameter | Options | | | | | Optimal Value |
|---|---|---|---|---|---|---|
| C | 0.1 | 1 | 10 | 30 | 100 | 100 |
| Epsilon | 0.01 | 0.1 | 0.5 | 1 | | 0.1 |
| Kernel | Linear | Poly | Rbf | Sigmoid | | Rbf |
| Gamma | Scale | Auto | | | | auto |

*Table A.2: Decision Tree (10% data, 200 iterations):*

| Parameter | Options | | | | | Optimal Value |
|---|---|---|---|---|---|---|
| Max_depth | 3 | 5 | 7 | 10 | | 10 |
| Min_sample_split | 2 | 5 | 7 | 9 | | 2 |
| Min_sample_leaf | 1 | 2 | 4 | 7 | 9 | 9 |
| Max_features | None | sqrt | log2 | | | None |

*Table A.3: Random Forest (10% data, 200 iterations)*

| Parameters | Options | | | Optimal value |
|---|---|---|---|---|
| N_estimator | 100 | 200 | | 200 |
| Max_depth | 5 | 7 | 10 | 10 |
| Min_sample_split | 2 | 5 | 7 | 5 |
| Min_samples_leaf | 1 | 2 | 4 | 2 |
| Max_feature | ‘sqrt’ | ‘log2’ | | ‘sqrt’ |
| Bootstrap | True | False | | False |

*Table A.4: eXtreme Gradient Boosting (10% data, 200 iterations):*

| Parameters | Options | | | Optimal value |
|---|---|---|---|---|
| N_estimator | 100 | 200 | 300 | 300 |
| Learning_rate | 0.01 | 0.05 | 0.1 | 0.05 |
| Max_depth | 6 | 8 | 10 | 6 |
| Min_child_weight | 1 | 3 | 5 | 5 |
| Gamma | 0 | 0.1 | 0.3 | 0.1 |

| | | | | |
|---|---|---|---|---|
| **Subsample** | 0.7 | 0.8 | 0.9 | 0.8 |
| **Colsample_bytree** | 0.7 | 0.8 | 0.9 | 0.9 |
| **Reg_alpha** | 0 | 0.05 | 0.1 | 0 |
| **Reg_lambda** | 0.1 | 1 | 10 | 1 |

# *Appendix B – Summary Statistics*

*Table B.1: Summary Statistics of the Delivery Data*

| | assumed_stop_time | estimated_driving_time | I6 | I9 | number_of_mailboxes_at_stop | trip_id |
|---|---|---|---|---|---|---|
| **count** | 2398672 | 2398672 | 2398672 | 2398672 | 2398672 | 2398672 |
| **mean** | 84.3 | 23.9 | 0.1 | 1.1 | 5.9 | 63683.6 |
| **std** | 58.9 | 16.6 | 0.4 | 0.6 | 12.2 | 36516.6 |
| **min** | 10.0 | 0.0 | 0.0 | 0.0 | 1.0 | 20.0 |
| **25%** | 41.6 | 9.9 | 0.0 | 1.0 | 1.0 | 31850.0 |
| **50%** | 67.0 | 21.4 | 0.0 | 1.0 | 1.0 | 62201.0 |
| **75%** | 110.0 | 36.6 | 0.0 | 1.0 | 7.0 | 95053.0 |
| **max** | 300.0 | 60.0 | 17.0 | 25.0 | 394.0 | 125580.0 |

*Table B.2: Aggregated data of continuous numerical values for unit codes*

| Unit Code | Mean Assumed Stop Time | Mean Estimated Driving Time | Mean I6 | Mean I9 | Mean Rush Time | Mean Block Stop | Mean Number of Mailboxes at Stop |
|---|---|---|---|---|---|---|---|
| 101366 | 94.09 | 16.31 | 0.04 | 1.27 | 0.12 | 0.74 | 16.36 |
| 101367 | 94.46 | 20.79 | 0.09 | 1.14 | 0.15 | 0.62 | 10.91 |
| 101377 | 87.57 | 21.46 | 0.13 | 1.12 | 0.16 | 0.44 | 8.49 |
| 101434 | 82.24 | 27.64 | 0.12 | 1.11 | 0.13 | 0.08 | 2.57 |
| 101461 | 83.18 | 26.11 | 0.10 | 1.27 | 0.10 | 0.11 | 3.54 |
| 101609 | 94.00 | 25.14 | 0.13 | 1.18 | 0.07 | 0.30 | 6.74 |
| 101656 | 97.73 | 20.12 | 0.15 | 1.10 | 0.17 | 0.44 | 9.06 |
| 101792 | 82.29 | 20.92 | 0.07 | 1.19 | 0.13 | 0.60 | 14.06 |
| 101793 | 68.50 | 22.98 | 0.14 | 1.07 | 0.18 | 0.35 | 5.44 |
| 101799 | 85.69 | 27.74 | 0.13 | 1.12 | 0.15 | 0.05 | 2.17 |
| 101881 | 102.91 | 25.16 | 0.12 | 1.18 | 0.12 | 0.09 | 2.37 |
| 101882 | 100.49 | 26.56 | 0.11 | 1.18 | 0.11 | 0.06 | 2.19 |
| 101890 | 85.13 | 28.50 | 0.12 | 1.14 | 0.09 | 0.05 | 1.73 |
| 101892 | 92.73 | 27.80 | 0.12 | 1.09 | 0.21 | 0.08 | 2.64 |
| 101915 | 99.99 | 28.07 | 0.09 | 1.06 | 0.16 | 0.08 | 2.29 |
| 101930 | 89.80 | 21.34 | 0.15 | 1.14 | 0.16 | 0.12 | 3.87 |
| 102504 | 83.80 | 25.26 | 0.10 | 1.09 | 0.10 | 0.13 | 3.14 |
| 102666 | 72.44 | 25.69 | 0.15 | 1.04 | 0.11 | 0.17 | 4.43 |
| 106245 | 83.08 | 27.46 | 0.12 | 1.05 | 0.27 | 0.09 | 2.87 |
| 106530 | 78.05 | 24.39 | 0.11 | 1.08 | 0.09 | 0.21 | 3.80 |
| 107480 | 83.45 | 23.84 | 0.10 | 1.11 | 0.05 | 0.12 | 2.79 |
| 413706 | 87.85 | 25.44 | 0.13 | 1.09 | 0.10 | 0.08 | 2.56 |
| 413715 | 77.99 | 22.56 | 0.12 | 1.04 | 0.20 | 0.14 | 4.02 |

# *Appendix C – Data Example*

*Table C.1: Random sample of the delivery data*

| unit_code_nom | assumed_stop_time | estimated_driving_time | l6 | l9 | rush_time_bol | block_stop_bol | number_of_mailboxes_at_stop | temperature_ord | precipitation_ord | recent_snow_ord | snow_depth_ord | weekday_nom | season_nom | max_volume_cm3_ord | trip_id |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 101915 | 128.00 | 51.00 | 0 | 1 | False | False | 1 | no frost | light | none | none | 2 | Summer | small | 71925 |
| 106530 | 89.30 | 49.70 | 0 | 1 | False | False | 1 | cold | none | none | deep | 2 | Winter | small | 34388 |
| 101609 | 47.67 | 1.33 | 0 | 1 | False | True | 18 | no frost | none | none | none | 4 | Spring | small | 117352 |
| 102666 | 152.58 | 20.42 | 0 | 1 | False | True | 11 | cold | light | moderate-heavy | deep | 1 | Winter | medium | 111804 |
| 102504 | 131.93 | 49.07 | 0 | 2 | False | True | 13 | no frost | none | none | none | 1 | Spring | small | 50360 |
| 101793 | 67.61 | 5.39 | 1 | 0 | False | False | 1 | no frost | none | none | none | 4 | Spring | medium | 121735 |
| 101377 | 51.36 | 30.64 | 0 | 1 | False | False | 1 | no frost | none | none | none | 4 | Autumn | small | 86430 |
| 101793 | 28.00 | 0.00 | 0 | 2 | False | False | 1 | no frost | none | none | none | 4 | Autumn | small | 87490 |
| 106530 | 36.35 | 13.65 | 0 | 1 | False | False | 1 | no frost | light | none | deep | 2 | Spring | small | 119365 |
| 101799 | 50.25 | 40.75 | 0 | 1 | False | False | 2 | very cold | none | none | moderate | 3 | Winter | small | 101773 |
| 101792 | 103.85 | 37.15 | 1 | 0 | False | True | 9 | cold | none | none | moderate | 5 | Winter | medium | 103979 |
| 101793 | 83.58 | 6.42 | 0 | 1 | False | False | 4 | no frost | none | none | none | 3 | Spring | large | 116064 |
| 101792 | 107.02 | 12.98 | 0 | 3 | True | True | 66 | no frost | none | none | none | 1 | Autumn | medium | 85810 |
| 101881 | 70.46 | 36.54 | 0 | 1 | False | False | 1 | no frost | none | none | none | 1 | Spring | small | 116665 |
| 101461 | 28.66 | 43.34 | 0 | 2 | False | False | 1 | no frost | none | none | none | 1 | Autumn | small | 90398 |
| 101793 | 17.23 | 58.77 | 0 | 1 | False | False | 4 | cold | none | none | moderate | 1 | Winter | small | 110086 |
| 106530 | 112.09 | 11.91 | 0 | 1 | False | False | 2 | cold | none | none | deep | 2 | Winter | medium | 103263 |
| 101799 | 60.49 | 32.51 | 0 | 1 | False | False | 1 | no frost | none | none | none | 2 | Summer | small | 74464 |
| 101792 | 156.60 | 29.40 | 2 | 1 | False | False | 2 | no frost | light | none | shallow | 5 | Autumn | large | 88846 |
| 102504 | 89.39 | 47.61 | 0 | 1 | False | False | 3 | no frost | none | none | deep | 2 | Winter | medium | 38564 |
| 101377 | 76.58 | 9.42 | 0 | 1 | False | True | 8 | no frost | none | none | none | 4 | Summer | small | 58264 |
| 102666 | 88.06 | 14.94 | 0 | 1 | False | True | 8 | no frost | none | none | deep | 1 | Spring | small | 47698 |
| 102666 | 48.96 | 21.04 | 0 | 2 | True | False | 1 | no frost | none | none | none | 3 | Summer | small | 73419 |
| 101793 | 44.81 | 2.19 | 0 | 1 | False | True | 32 | cold | none | none | none | 3 | Autumn | small | 94477 |
| 101793 | 134.19 | 42.81 | 0 | 1 | False | True | 8 | no frost | moderate | none | none | 1 | Autumn | small | 24216 |
| 101461 | 88.17 | 18.83 | 0 | 1 | False | False | 1 | no frost | none | none | none | 4 | Autumn | small | 20946 |
| 101890 | 100.34 | 13.66 | 0 | 2 | False | True | 11 | cold | none | none | none | 2 | Spring | medium | 44932 |

## *Appendix D – Distribution of data*

*Fig D.1: Distribution of numerical values*

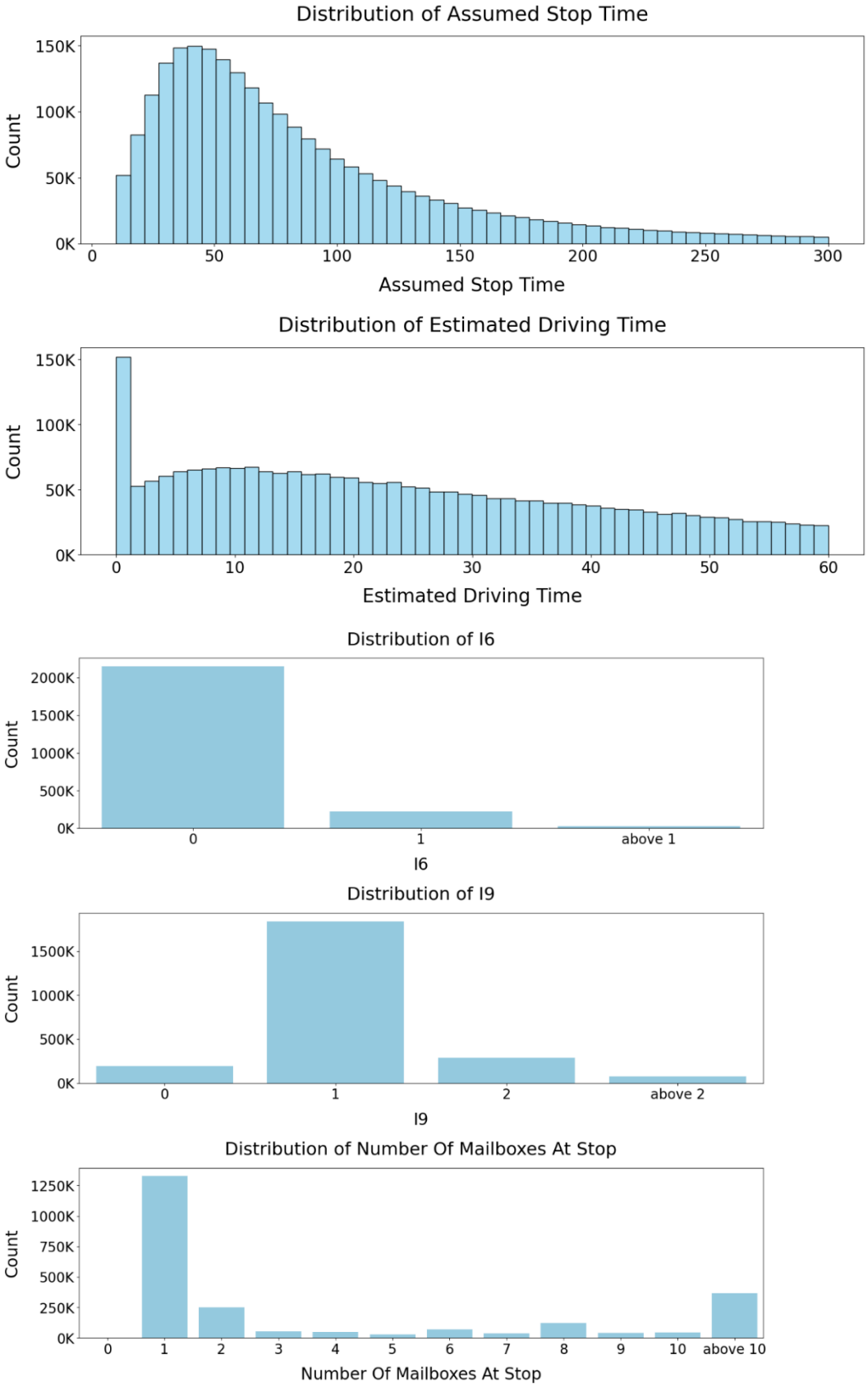


*Fig D.2: Distribution of ordinal values*

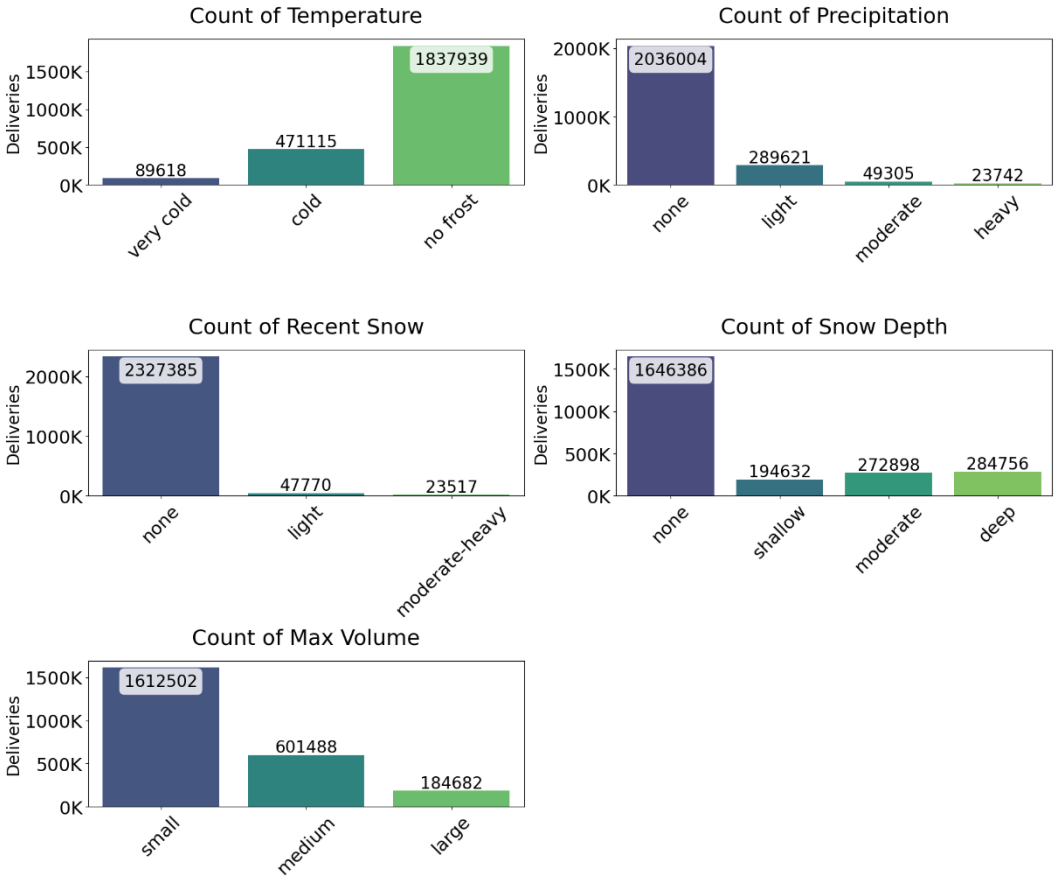



*Fig D.3: Distribution of boolean values*

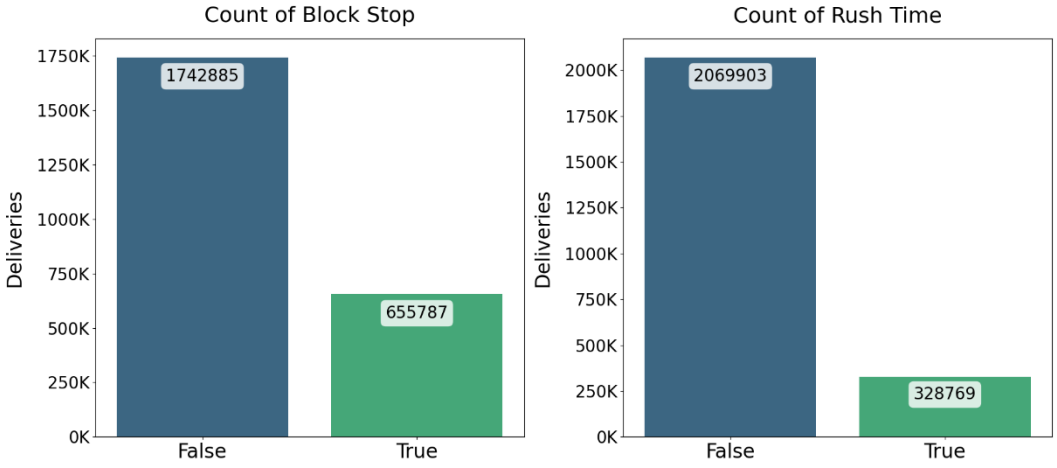
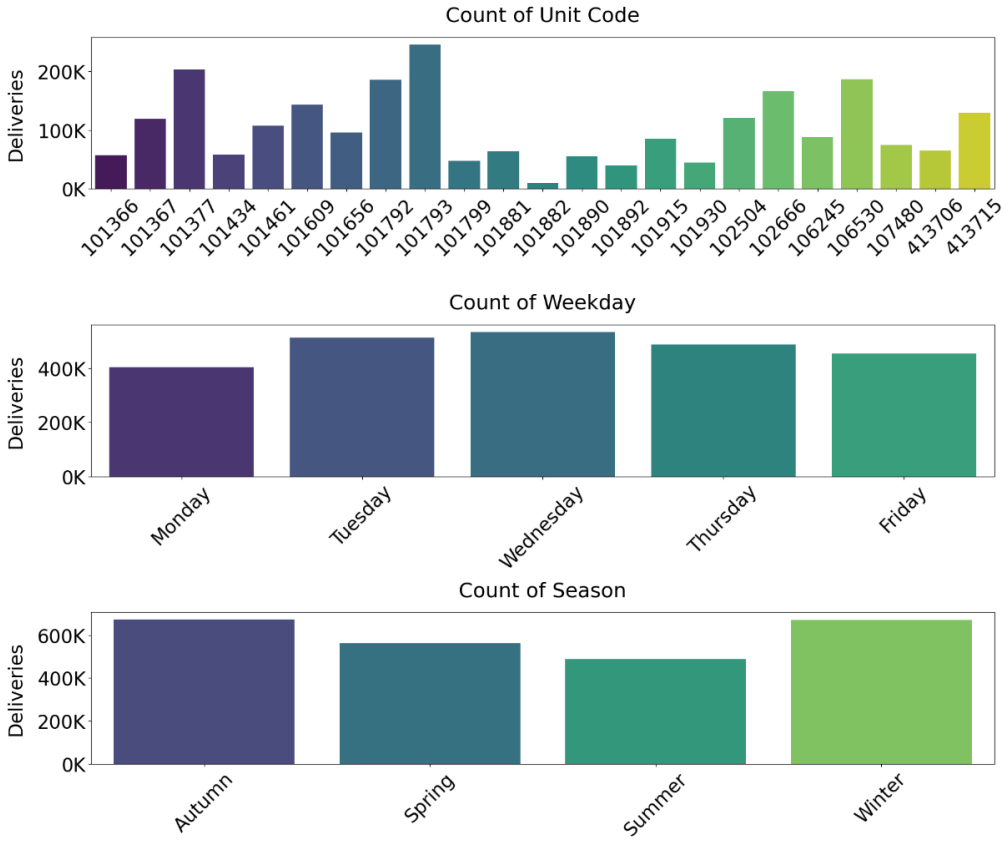
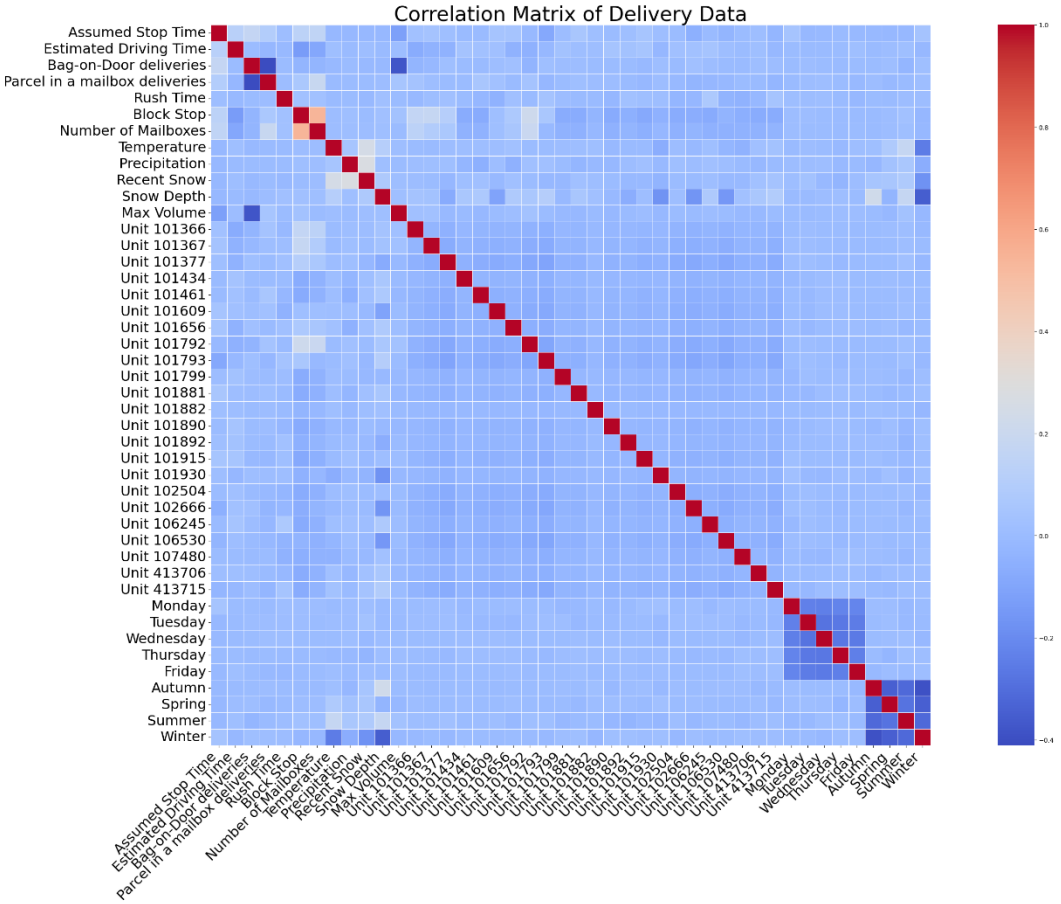


*Fig D.4: Distribution of nominal values*



# Appendix E – Correlation Matrix for the delivery data



## *Appendix F – Nominal value bins for the delivery data*

| Temperature (T) | Very cold | Cold | No frost |
|---|---|---|---|
| | T <= -8 | -8 < T <= 1 | T > 1 |

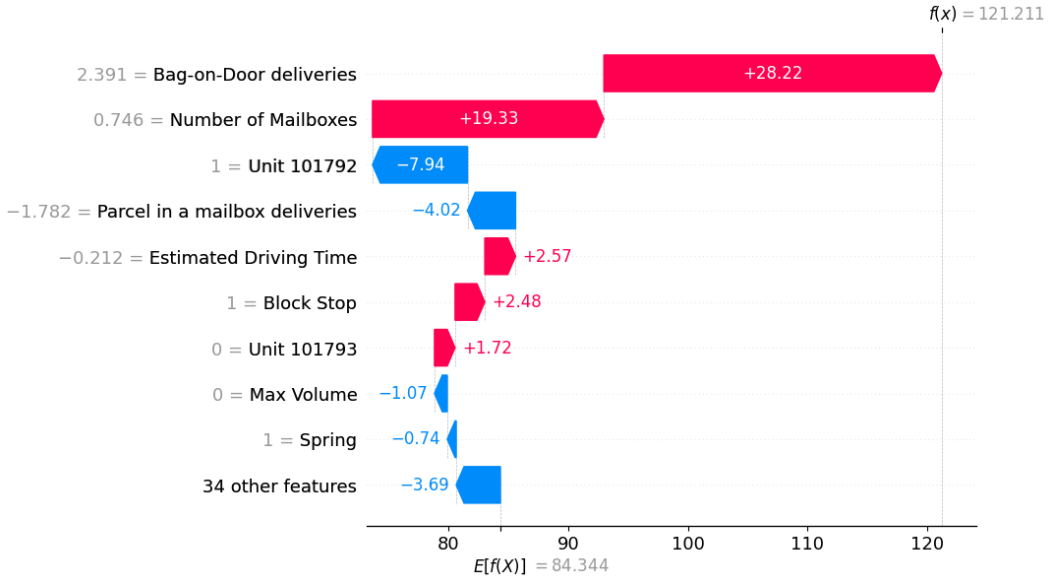
| Precipitation (P) | None | Light | Moderate | Heavy |
|---|---|---|---|---|
| | P = 0 | 0 < P <= 1 | 1 < P = 2 | P > 2 |

| Recent Snow (RS) | None | Light | Moderate-heavy |
|---|---|---|---|
| | RS = 0 | 0 < RS <= 1 | RS > 1 |

| Snow Depth (SD) | None | Shallow | Moderate | Deep |
|---|---|---|---|---|
| | SD = 0 | 0 < SD <= 5 | 5 < SD <= 25 | SD > 25 |

| Max Volume (MV) | Small | Medium | Large |
|---|---|---|---|
| | 0 <= MV <= 4000 | 4000 < MV <= 10000 | MV > 10000 |

## *Appendix G – SHAP on a single delivery*




Example of a single delivery and how SHAP explains the model's prediction. E[f(x)] = Expected model output, f(x) = Prediction.

## *Appendix H – Density plots of unit target vs prediction*