KADIR HAS UNIVERSITY

SCHOOL OF GRADUATE STUDIES

PROGRAM OF INDUSTRIAL ENGINEERING

REAL TIME PREDICTION OF DELIVERY DELAY WITH MACHINE LEARNING

BÜŞRA ÜLKÜ KÜP

MASTER OF SCIENCE THESIS

İSTANBUL, JULY, 2023

REAL TIME PREDICTION OF DELIVERY DELAY WITH MACHINE LEARNING

BÜŞRA ÜLKÜ KÜP

A thesis submitted to
the School of Graduate Studies of Kadir Has University
in partial fulfilment of the requirements for the degree of
Master of Science in
Graduate Programme

İstanbul, July, 2023

APPROVAL

This thesis titled REAL TIME PREDICTION OF DELIVERY DELAY WITH MACHINE LEARNING submitted by BÜŞRA ÜLKÜ KÜP, in partial fulfillment of the requirements for the degree of Master of Science in Graduate Programme is approved by

Asst. Prof. Mustafa Hekimoğlu (Advisor) Kadir Has University	....
Prof. Dr. Ahmet Deniz Yücekaya Kadir Has University	....
Asst. Prof. Emre Çelebi Yeditepe University	....

I confirm that the signatures above belong to the aforementioned faculty members.

....

Prof. Dr. Mehmet Timur Aydemir

Dean of School of Graduate Studies

Date of Approval: 27.07.2023

DECLARATION ON RESEARCH ETHICS AND PUBLISHING METHODS

I, BÜŞRA ÜLKÜ KÜP; hereby declare

- that this Master of Science Thesis that I have submitted is entirely my own work and I have cited and referenced all material and results that are not my own in accordance with the rules;
- that this Master of Science Thesis does not contain any material from any research submitted or accepted to obtain a degree or diploma at another educational institution;
- and that I commit and undertake to follow the "Kadir Has University Academic Codes and Conduct" prepared in accordance with the "Higher Education Council Codes of Conduct".

In addition, I acknowledge that any claim of irregularity that may arise in relation to this work will result in a disciplinary action in accordance with university legislation.

BÜŞRA ÜLKÜ KÜP

. . . .
27.07.2023

Dedicated to my partner in life. His patience, understanding, and encouragement are cherished beyond words.

ACKNOWLEDGEMENT

My academic journey commenced in September 2020, embarking on a master's program that spanned nearly three years of dedicated study at Kadir Has University. As I reflect upon this formidable period of growth and learning, I want to express my gratitude to those who have contributed to my development.

I am deeply grateful to my esteemed thesis advisor, Assist. Prof. Mustafa Hekimoğlu, whose unwavering guidance, insightful feedback, and dedicated mentorship have been invaluable throughout this research journey. His expertise and support have contributed significantly to the shaping and refinement of this thesis.

I extend my heartfelt appreciation to my co-advisor, Prof. Dr. Ahmet Deniz Yücekaya, for their valuable contributions and constructive insights that enriched the quality of my work.

My deepest thanks go to my dedicated and collaborative working partner, Barış Bayram, Ph.D. His collective efforts and shared insights have significantly contributed to the success of this endeavor. His diverse perspectives added depth and clarity to the research process. Also, I would like to sincerely thank Gözde Aydın, who helped me whenever I got stuck during the thesis process.

To my cherished family, their unwavering encouragement, understanding, and patience have given me the strength and motivation to overcome challenges and achieve this significant milestone. And to my friends, thank you for the laughter and moral support.

Büşra Ülkü Küp

İstanbul, July, 2023

REAL TIME PREDICTION OF DELIVERY DELAY WITH MACHINE LEARNING

ABSTRACT

The growth of the Internet has led to a considerable transformation in the e-commerce and logistics industries, resulting in a surge in online shopping and an increased need for efficient delivery operations. This study's impact is significant as its findings offer valuable insights into predicting delivery delays using machine learning, allowing logistics companies to optimize their processes and enhance customer satisfaction. Moreover, the use of real-world data in this study lends credibility to the findings and highlights the advantages of integrating real-time and machine learning in academic research. Four of the most commonly used supervised classification algorithms in the literature - Logistic Regression, XGBoost, CatBoost, and Random Forest - were employed in this study to predict early delivery delays in a e-commerce logistics company using real-world data. To enable continuous prediction throughout the entire process, the delivery process was split into 11 and 15 steps for different delivery types. Prediction models were optimized separately for each step's unique model during the process, using parameter tuning and feature selection. When evaluating final ROC-AUC scores for models created using four classifiers, it was found that the ROC-AUC scores for XGBoost ranged from 71.5% to 99.9%, while the ROC-AUC scores for CatBoost ranged from 72.4% to 99.9%. Although the results of the two classifiers were adjacent in the different steps, CatBoost had slightly better performance metrics overall compared to XGBoost.In future work, a comprehensive range of algorithms will be explored, additional features will be integrated, and deep learning models will be investigated to achieve greater accuracy and robustness. By utilizing larger datasets, even at a big-data scale, proposed models can uncover more advanced insights and improved performance. However, this method does require high computational hardware and power. The challenges associated with model interpretability and computational requirements will be addressed in next steps.

Keywords: logistics, e-commerce, machine learning, delay prediction, real time predictions, classification, catboost, random forest, xgboost

MAKİNE ÖĞRENMESİ İLE TESLİMAT GECİKMESİNİN GERÇEK ZAMANLI TAHMİNİ

ÖZET

İnternetin yaygınlaşması, e-ticaret ve lojistik endüstrilerinde önemli bir dönüşüme yol açmıştır. Bu dönüşüm, çevrimiçi alışverişte önemli bir artışa öncülük etmiş ve rekabetçi ortamda kargo şirketlerinin operasyonel verimliliğini arttırma ihtiyacını ortaya çıkarmıştır. Teslimat süreçlerini optimize etmek ve müşteri memnuniyetini artırmak amacıyla, makine öğrenimi kullanılarak teslimat gecikmelerinin tahmin edilmesi, lojistik şirketlerine önemli katkılar sağlayacaktır. Ayrıca, gerçek dünya verilerinin bu çalışmada kullanılması, elde edilen sonuçların güvenilirliğini artırmakta ve makine öğreniminin lojistik endüstrisi odaklı akademik araştırmalarda kullanılmasının avantajlarını vurgulamaktadır. Bu çalışmada, Logistic Regression, XGBoost, CatBoost ve Random Forest gibi en yaygın kullanılan dört denetimli sınıflandırma algoritması, bir e-ticaret lojistik şirketinde gerçek zamanlı veriler kullanılarak teslimat gecikmelerinin tahmin edilmesi amacıyla uygulanmıştır. Tüm süreç boyunca sürekli gecikme tahmini yapabilmek için, tüm teslimat süreci farklı gönderi türleri için sırasıyla 11 ve 15 adım şeklinde ayrıştırılmış ve her adım için ayrı tahmin modelleri oluşturulmuştur. Bu modellerin performansını artırmak için optimal parametre ve öznitelik seçimi yöntemleri kullanılmıştır. Kullanılan bu optimizasyon teknikleri, modellerin performansları üzerinde önemli bir olumlu etki sağlamıştır. Elde edilen sonuçlara göre, dört farklı sınıflandırıcı kullanılarak oluşturulan modellerin nihai ROC-AUC skoru ile değerlendirildi. XGBoost için ROC-AUC puanları %71,5 ile %99,9 arasında değişmekteyken, CatBoost için ROC-AUC puanları %72,4 ile %99,9 arasında değişim gösterdi. Bu iki sınıflandırıcı farklı adımlarda çok yakın performans göstermiş olsalar da, CatBoost genel olarak XGBoost'a kıyasla biraz daha iyi bir sonuç ortaya koymuştur. Gelecekteki çalışmalarda, daha doğru sonuçlar elde edebilmek için derin öğrenme bazlı sınıflandırma methodlarının denenmesi ve ek özniteliklerin entegre edilmesi üzerine çalışmalar yapılacaktır. Daha büyük veri

kümeleri kullanılması önerilen gecikme tahmini yaklaşımının, daha etkin çıktılar ve performans iyileştirmeleri sağlayacaktır. Ancak, daha büyük veri kümeleri elde edilmesi, işlenmesi ve derin öğrenme modellerinin denenmesi için daha yüksek performanslı donanımsal, işlemci ve hafıza, kaynaklara ihtiyaç duyulacaktır. Bu zorlukların üstesinden gelmek ve daha yüksek performanslı çözümler sunmak için çeşitli stratejiler ve teknikler geliştirilmeye devam edilecektir.
Anahtar Sözcükler: lojistik, e-ticaret ,makine öğrenimi, gecikme tahmini, gerçek zamanlı tahmin, sınıflandırma, catboost, random forest, xgboost

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

ANN	Artificial Neural Network
AUC	Area Under Curve
AVL	Automatic Vehicle Location
CatBoost	Categorical Boosting
CPU	Central Processing Unit
FN	False Negative
FP	False Positive
GBDT	Gradient Boosting Decision Tree
GPS	Global Positioning System
GPU	Graphics Processing Unit
LGB	Light-Gradient Boosting
LR	Logistic Regression
LSTM	Long Short-Term Memory Network
MAE	Mean Absolute Error
OD	Origin-Destination
OLR	Ordinary Logistic Regression
RF	Random Forest
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
SHAP	SHapley Additive exPlanations
TN	True Negative
TP	True Positive
XGBoost	Extreme Gradient Boosting

1. INTRODUCTION

The Internet has become an essential tool for conducting many business transactions and is increasingly central in all aspects of life. With its widespread use worldwide, e-commerce users are constantly increasing. In recent years, Turkey has experienced a positive acceleration in e-commerce volume. However, this increase in e-commerce transactions has also negatively impacted traditional shopping numbers (Açılar, 2016).

The pandemic has substantially impacted online shopping, significantly increasing the number of e-commerce users and order volume. This shift is due to the improved efficiency and accessibility of pre-sales services, such as site and product comparisons. As a result, e-commerce is becoming increasingly favorable compared to physical shopping. Additionally, e-commerce reduces the potential time loss resulting from out-of-stock items in physical shopping. The energy and effort required for online shopping are significantly lower than for physical shopping. E-commerce eliminates the limitations on product and store alternatives caused by transportation in physical shopping, enabling cross-country shopping.

One of the benefits of e-commerce is that it provides quick access to information about many alternatives, making it easy to choose the ideal product. Compared to the past, obtaining knowledge that provides the most suitable conditions for shopping and enables the most efficiency has become straightforward. This phenomenon is best expressed by the term ”Knowledge Economy” (Ozmen, Öner, Khosrowshahi, & Underwood, 2013).

The rise of online shopping has increased the demand for logistics services responsible for storing and distributing shipments. It has become crucial to plan shipment densities efficiently, determine carrier and vehicle numbers, ensure timely delivery to customers, and establish the optimal number of collaborating companies. To survive against changing conditions, increase profits, and perform operations quickly and optimally, logistics companies must take advantage of technological innovations in this sector, as in many other industries. The relationship between logistics and technology has become increasingly critical to meet the sector's needs and keep up with the competitive environment.

Logistics companies utilize various technological methods to increase profitability and efficiency, reduce costs, strengthen their competitive position, and improve customer satisfaction through rapid distribution. For example, they can manage vehicles by accessing their real-time location using vehicle tracking systems. Using package recognition systems, they can also direct packages to their appropriate destination based on their characteristics, such as size, shape, weight, and color. Cloud computing and big data technologies are utilized to process and manage transactional data effectively. Furthermore, the growth of data and easy access to it, the development of more powerful computers and efficient algorithms, the increase in Internet use, and the emergence of cloud computing and big data systems have significantly increased interest in machine learning.

Machine learning methods are frequently preferred for problem-solving across numerous fields, including healthcare, finance, manufacturing, and logistics. Compared to traditional programming approaches, they offer several advantages, such as adapting to new and changing data, processing large amounts of data quickly and efficiently, and learning from data. These advantages make machine learning a powerful tool for solving complex logistics problems, such as predicting delivery demand based on historical data, planning capacity, allocating resources more effectively, detecting fraudulent activity to reduce losses, and predicting shipments that are likely to be delayed in the future.

This thesis explores a significant concern within the logistics industry: the issue of delayed shipment. Such delays can adversely affect a company's operations, reputation, and customer satisfaction. Ensuring timely delivery is paramount in maintaining high levels of customer satisfaction, as frequent delays can erode trust in the company, ultimately resulting in a decline in shipment volume. Logistic companies may opt for approaches such as implementing additional shipping charges or providing coupon codes for delayed shipments to expedite deliveries and enhance customer satisfaction. However, these temporary solutions often lead to increased costs. In light of this, the present study proposes a more efficient and enduring solution: leveraging machine learning techniques to predict potential delivery delays early by using historical transactional data.

The data set used for analysis and modeling is divided into two groups which are delayed and on-time shipments. Successful shipments refer to those delivered within the designated timeframe, while unsuccessful shipments encompass those that have experienced delays. These classifications serve as the labels for the target variable. Notably, shipments undergo multiple stages before reaching their intended recipients. While the number of steps may vary across cargo companies, some stages, such as departure and arrival units, are common. Iterative assessments were conducted for each step, considering various factors, including the time elapsed from order placement to delivery for each stage, to calculate the probability of delays. In this context, supervised classification algorithms such as CatBoost, Extreme Gradient Boosting (XGBoost), Logistic Regression (LR), and Random Forest (RF) are used due to the availability of labeled data. By utilizing the labeled data within the training set, models aimed to discern the underlying patterns and associations between the characteristics of the deliveries and their corresponding outcomes. The main purpose of this study is to estimate whether there is a possibility of delay before a shipment reaches the midpoint of its operational processes. The goal is to inform the appropriate department about potential late shipments and take action to prevent issues before they occur. This approach aims to uphold high levels of customer satisfaction and operational efficiency. In this way, it is aimed to keep customer satisfaction and operational efficiency at the optimum level.

Section 2 offers a literature review conducted for this study. Section 3 defines the research problem and contextualizes it. Furthermore, that section details the algorithms employed in the study, explains the feature selection process, and describes the parameter tuning methodologies applied to optimize the models. Section 4 delves into the practical application of the research. It provides valuable insights into business operations modeling. Additionally, this section highlights the data analysis procedures undertaken and documents the experiments conducted using various machine learning models. Section 5 presents the conclusive findings derived from the study, summarizing the key outcomes and their implications. Moreover, it offers valuable insights into potential avenues for future research, identifying areas that require further exploration to advance the field. These sections provide a structured and comprehensive framework for the study, covering literature review, methodology, application, and concluding remarks.

1.1 Motivation and Aim

Logistics companies play a crucial role in facilitating the delivery of shipments from sellers to recipients in online shopping. Each company employs strategies to fortify its market position in an increasingly competitive environment. Timely delivery of consignments is critical for customer satisfaction, the company's reputation, and profitability. However, several internal and external factors can influence the duration of shipment deliveries. Predicting potential delays in advance is crucial to mitigate the effects of delays and enable timely actions.

The thesis aims to deploy a machine learning model that utilizes transactional data to accurately predict delays at an early stage in deliveries. By leveraging this predictive model, logistics companies can proactively address possible delays and minimize their consequences, enhancing operational efficiency and customer satisfaction.

1.2 Research Questions

The developed machine-learning model will provide information that can be used to answer the following research questions:

1. How can data from a complex business process be effectively modeled to predict shipment delays accurately?
2. How can an early prediction of package delay be estimated by utilizing machine learning algorithms?
3. Which specific features or attributes are critical for achieving reliable and robust prediction of shipment delays?
4. What is the comparative performance of boosting algorithms against traditional algorithms in accurately predicting shipment delays?

The following supportive questions are suggested based on these primary questions, which would address much more detailed aspects of the study.

1. Among various classification algorithms, which algorithm or combination of algorithms exhibit the highest predictive performance in predicting shipment delays?
2. To what extent is parameter optimization essential for the machine learning models used in predicting shipment delays? If optimization is required, which model parameters yield the most optimal results?
3. Which evaluation metrics are the most appropriate for assessing the performance and effectiveness of the developed machine learning models in predicting shipment delays?
4. Which steps and attributes exert the greatest impact on ensuring on-time delivery of shipments?

2. LITERATURE REVIEW

Regarding delivery management, one of the most critical factors is the ability to predict delivery delays. Delivery delays can significantly impact a company's reputation, customer satisfaction, and overall profitability. Therefore, many researchers have focused on developing models and algorithms to predict delivery delays. Salari, Liu, and Shen (2022) proposed a data-driven framework for predicting the distribution of order delivery time and setting a promised delivery time to customers cost-effectively. The proposed machine learning models use tree-based models and asymmetric loss functions to generate distributional forecasts and provide a cost-sensitive decision rule for deciding the promised delivery day from the predicted distribution. Tested on a real-world dataset, the proposed framework demonstrates superior forecasting performance and can potentially improve sales volume by 6.1% over the current policy. Araujo and Etemad (2021) explored the use of deep learning for last-mile parcel delivery time prediction using a large-scale parcel dataset provided by Canada Post. The study presents an origin-destination (OD) formulation. It investigates three categories of convolutional-based neural networks, demonstrating their superior performance compared to classical machine learning models and referenced OD solutions. The study provides an end-to-end neural pipeline that leverages parcel OD data and the weather to predict the delivery duration accurately and can potentially improve user experience and aid last-mile postal logistics. Jonquais and Krempl (2019) investigated machine learning and predictive analytics to improve the estimated arrival time for shipments in the shipping industry. By training the model on historical shipment data and incorporating external factors such as holiday seasons and port congestion levels, a machine learning model was developed with a mean absolute error (MAE) of 3.74 days at the time of booking transportation, outperforming the baseline model which only considers historical average transit times on a shipping lane. However, the performance of the machine learning model was found to be better for long lead times compared to short lead times. Metzger et al. (2015) investigated and compared three main classes of predictive monitoring techniques for business processes: machine learning, constraint satisfaction, and quality-of-service (QoS) aggregation. An empirical analysis using an industrial case study in transport and logistics assesses the techniques based on accuracy indicators and lead time for accurate predictions. The findings suggest that combining specific techniques can improve precision and recall, with evidence showing improvements of 14% in precision when combining constraint satisfaction with QoS aggregation and 23% in recall when combining machine learning with constraint satisfaction. Khiari and Olaverri-Monreal (2020) discussed the challenges in long-term delivery time prediction for transportation and postal services. The study investigated the effectiveness of several machine-learning techniques, including linear regression and tree-based ensembles. It demonstrated the applicability of travel time prediction to mitigate high delays in postal services, highlighting the superior performance of boosting algorithms such as light-gradient boosting (LGB) and CatBoost.

Delays in bus and plane schedules can impact the timely arrival of goods and products, which can subsequently lead to delivery delays. For example, if a bus carrying packages or a plane carrying cargo experiences delays, it can disrupt the logistics chain and result in delayed deliveries. Vernimmen, Dullaert, and Engelen (2007) highlighted the low schedule reliability in the shipping industry, particularly concerning container delays and their implications for various actors in the supply chain. By focusing on shippers (consignees), the paper presented a case study demonstrating the impact of the schedule. It emphasized the potential cost savings achieved through improved schedule performance. Therefore, accurate scheduling and prediction of bus and plane delays are essential factors in minimizing the risk of delivery delays and ensuring efficient transportation operations. Machine learning algorithms have become frequently employed in logistics recently to forecast bus or flight delays. Taparia and Brady (2021)'s study proposed and developed predictive models based on historical AVL/GPS data, bus routes, and bus stop information to estimate bus journey and arrival times. The evaluation demonstrated that long short-term memory network (LSTM) outperformed Linear Regression and showed comparable performance to artificial neural network (ANN) in predicting overall journey times. At the same time, gradient boosting exhibited superior performance and robustness in predicting bus arrival times at bus stops compared to historical averaging and linear regression models. This study also supports the feasibility of accurately predicting bus journey time using historical GPS observations and bus route information alone. Kawatani, Yamaguchi, Sato, Maita, and Mine (2021) proposed prediction models for bus delay across various routes using one month of probe data and evaluated multiple machine learning models. The experimental results revealed the superior performance of the gradient boosting decision tree (GBDT) model and emphasized the significance of considering travel time over prior intervals. Esmaeilzadeh and Mokhtarimousavi (2020)'s study utilized a support vector machine (SVM) model to analyze flight delay outcomes and investigated factors such as pushback delay, taxi-out delay, ground delay program, and demand-capacity imbalance, revealing their significant associations with flight departure delays. Rebollo and Balakrishnan (2014) proposed a new class of models for predicting air traffic delays that consider both temporal and spatial delay states as explanatory variables and use RF models. The models were evaluated using operational data from 2007 and 2008, and the results showed good performance in classifying delays and predicting delay values. Balakrishna, Ganesan, Sherry, and Levy (2008)'s paper presented a method for estimating average taxi-out times at airports in advance using a probabilistic framework of stochastic dynamic programming and reinforcement learning. The algorithm was tested at John F. Kennedy International Airport, and the predicted average taxi-out times matched with an overall accuracy of about 60

Food delivery time prediction and e-commerce shopping delivery delay prediction are related because they both involve machine learning techniques to forecast the time it takes for a delivery to reach its destination. Both applications aim to provide accurate estimations of delivery time to improve customer satisfaction and optimize logistics operations. Machine learning models for food delivery can be trained on historical data to discover trends and variables like distance, traffic, order volume, and preparation time that affect delivery time. By analyzing these variables, the models can predict how long it will take for a food order to be delivered. Similarly, in e-commerce shopping delivery, machine learning models can leverage historical data and various features like distance, transportation mode, courier performance, and order characteristics to estimate delivery time. These models learn from past delivery patterns to provide accurate predictions and manage customer expectations. In both cases, the goal is to enhance the customer experience by providing reliable delivery time estimates, optimizing delivery routes, and mitigating delays. Machine learning prediction plays a crucial role in improving operational efficiency, managing logistics resources, and ultimately enhancing customer satisfaction in the food delivery and e-commerce shopping domains. Gao et al. (2021) investigated a deep learning model named FDNET for accurately estimating the driver's delivery route and time in the food delivery service, which is crucial for customer satisfaction and driver experience. FDNET predicts the probability of each feasible location the driver will visit next, significantly reducing the search space in delivery route generation and improving the utilization of various types of information. Yu et al. (2021) addressed the dispatching challenge in food delivery systems by proposing a prediction-based approach that combines machine learning and optimization. By considering the impact on driver efficiency and customer experience, the approach effectively decides to order delaying decisions, resulting in improved grouping rates without compromising customer satisfaction, as demonstrated through empirical experiments in various cities. Hoi, Leung, and Souza (2020) introduced an intelligent city system that leverages extensive data mining to predict food preparation time, addressing the challenge of accurately estimating the time for picking up take-out orders. The evaluation results demonstrated the effectiveness and practicality of the system in predicting food preparation time, contributing to the advancement of innovative city initiatives and providing benefits to both customers and delivery persons. By analyzing and extracting patterns from large volumes of data, machine learning models can learn from past observations to make accurate predictions about the time required for food preparation.

3. METHODOLOGY

3.1 Problem Definition

The logistics sector involves planning, managing, and carrying out transportation, storage, handling, and distribution services throughout the supply chain, from production to the final consumption of goods and services. The e-commerce logistics industry is a sub-sector of the logistics industry that specializes in the delivery of online shopping shipments. The supply chain processes covered by logistics operations can be broadly defined with three sub-stages. The first mile operations, which are the initial stage of transportation, cover all the steps in the process until the package leaves the vendor or manufacturer and reaches the relevant dock of the logistics provider company (Ranathunga, Wijayanayake, & Niwunhella, 2021). On the other hand, the last mile, the terminating stage of transportation, corresponds to the operations in the process from the final dock of the logistics company to the delivery to the customer (Gevaers, Van de Voorde, & Vanelslander, 2014). The middle mile is a general name for all transactions and transfers between these two processes. It varies from company to company, especially according to the middle-mile processes' number of middle docks (transfer units).

The rapidly growing logistics sector has accelerated this growth with a positive trend in e-commerce usage. With this growth, logistic activities have carried a critical role in the sustainability of online shopping and customer satisfaction and loyalty. This situation led to an increase in the number of companies operating in the e-commerce logistics field and a more competitive environment. In this competitive environment, factors such as increasing operational efficiency and reducing costs have gained vital importance for present companies to consolidate their position and for new entry companies to earn a place in the market. Customer satisfaction is one of the essential critical factors for success in the logistics industry. Providing a fast and on-time distribution experience is the most crucial criterion for customer satisfaction. In addition, delivery of the products to the customer without any damage is another critical issue. Deliveries may be delivered late to the recipient due to many reasons, such as late handing over of the package to the logistics provider by the seller, long loading and unloading times in transfer points, barcode recognition problems, inefficient routing of the travels between docks, weather conditions, heavy traffic, roadworks. These delays will negatively affect customer satisfaction and diminish customers' trust in logistics service provider companies. This project aims to predict the deliveries that are likely to be delayed in the logistics operations. The outputs of this study will create an opportunity for taking necessary precautions in advance and minimizing customer dissatisfaction caused by delays. In addition, the project's objectives also include a comprehensive analysis of delayed operations and identifying the antecedent factors causing related delays.

3.2 Classification

In classification algorithms, in cases where one class's observations are many times higher than the other class(s), standard decision-making systems fail to distinguish classes with fewer observations. This situation can be expressed as an underrepresentation of the minority class due to imbalanced data (Wojciechowski & Wilk, 2017). Although numerous studies have been conducted, imbalanced data is still an important challenge in classification and forecasting problems. It is difficult in machine learning algorithms to distinguish a small number of occurrences in problems (Chawla, Lazarevic, Hall, & Bowyer, 2003). The infrequency issue leads to misclassifications for these classes in classification algorithms (Haixiang et al., 2017). Misclassifying of the minority class can cause negative critical consequences and high costs in some applications, such as fraud detection (Gameng, Gerardo, & Medina, 2019).

In binary classification, the commonly used loss function is the binary cross-entropy loss, which is defined as:

$$\text{loss}(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \tag{3.1}$$

- $\Omega(f_j)$ is the regularization term used to penalize complex models. Two types of regularization are commonly used:

- L1 Regularization (Lasso): This regularization technique adds the absolute values of the weights as a penalty term in the objective function. It is defined as:

$$\Omega(f_j) = \gamma \sum_{k=1}^{K} |w_{jk}| \tag{3.2}$$

  where w_{jk} is the weight of the k^{th} feature in the j^{th} tree, and γ is the regularization parameter.
- L2 Regularization (Ridge): This regularization technique adds the square of the weights as a penalty term in the objective function. It is defined as:

$$\Omega(f_j) = \gamma \sum_{k=1}^{K} w_{jk}^2 \tag{3.3}$$

  where w_{jk} is the weight of the k^{th} feature in the j^{th} tree, and γ is the regularization parameter.

3.2.1 Logistic regression classifier

Regression methods offer a solution to explain the relationship between a response variable and more than one explanatory variable (Huppenkothen, Heil, Hogg, & Mueller, 2017). In regression models, simple and multiple linear regression methods are used when the dependent variable is generally in a continuous structure. The logistic regression (LR) model is more favorable than linear models when the dependent variable is categorical (Nigam & Govinda, 2017). This generalized model aims to provide high success compared to its easy-to-use in estimating the relationship between independent variables in continuous or categorical structure and dependent variables in categorical structure (Hosmer, Lemeshow, & Sturdivant, 2013).

The first applied studies using the LR model were conducted by Berkson (1944). LR models the probability of the occurrence of events over a linear function with various predictive variables. The model's parameters can be easily reported and can be used to generate mathematically interpretable functions. Also, the algorithm mainly employs the logistic function for classification. A feature set is used for classification, and weights are evaluated for each input variable. These weights are then utilized to predict the relevant class. The LR model has been widely used in the literature for problems where the output variable is categorical or binary. On the other hand, LR does not impose any restrictions on the input data set, supporting both categorical and quantitative values (Oktay, Üstün Özen, & Burmaoğlu, 2009).

The LR algorithm is more flexible than Linear Regression and can fit a wider range of data points. If the dependent variable is categorical, the LR model tends to provide more effective results. This is because LR has an elastic nature that enables it to cover most, if not all, of the data points that are being analyzed. In contrast, linear regression may not be able to fit all of the data points, especially when dealing with complex or non-linear relationships. The relationship between the dependent and independent variables can be linear, exponential, or polynomial. In such cases, the LR algorithm assumes a logit relationship between the variables and transforms the relationship to a linear form, thereby enabling the generation of non-linear models (Hosmer et al., 2013). The LR analysis in its fundamental methods categorizes the LR model based on the scale of the dependent variable. When the dependent variable has two categories, it is referred to as "Binary Logistic Regression" . If it has more than two unordered categories, it is called "Multinomial Logistic Regression," and if it has multiple ordered categories, it is termed "Ordinal Logistic Regression." In this study, the prediction of delivery delays was conducted using binary LR, and its performance was compared to other models. In cases where the classification problem is binary, there are two possibilities, 1 or 0, for the output Y value. According to the available data, the probability of both situations can be expressed as follows: $P(y_j = 0) = 1 - p_j$ and $P(y_j = 1) = p_j$. The model to fit these probabilities is the accommodated linear regression and is expressed as (Bishop, 2006):

$$\begin{aligned}\operatorname{logit}(P_1) = \ln\left(\frac{P_1}{1-P_1}\right) &= \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n \\ &= \beta_0 + \sum_{i=1}^{n} \beta_i x_i .\end{aligned} \tag{3.4}$$

In 3.4 , β_0 is the intercept value and $\beta_1, \beta_2 \ldots, \beta_n$ are the coefficients these are associated with the respective explanatory variable $x_1, x_2, \ldots, x_n$. As in 3.4, the variable output (dependent) is the natural logarithm of the probability ratio representing the ratio between two probabilities of a binary event Li and Jimenez (2018). In general, LR has fewer restrictions compared to ordinary linear regression (OLR). The LR algorithm does not assume linearity of the relationship between the explanatory variables and the response variable and does not require Gaussian distributed arguments. The probability of the possible outcome of an event as a function of the input (explanatory) variables is nonlinear for LR, as derived from Eq. 3.4, given in the Eq. 3.5 (Subasi & Erçelebi, 2005). The graphical representation of the sigmoid function, which is the complement of this equation, is provided in Figure 3.1.

$$P_1(x) = \frac{1}{1 + \mathrm{e}^{-\operatorname{logit}(P_1(x))}} = \frac{1}{1 + \mathrm{e}^{-\left(\beta_0 + \sum_{i=1}^{n} \beta_i x_i\right)}} \tag{3.5}$$

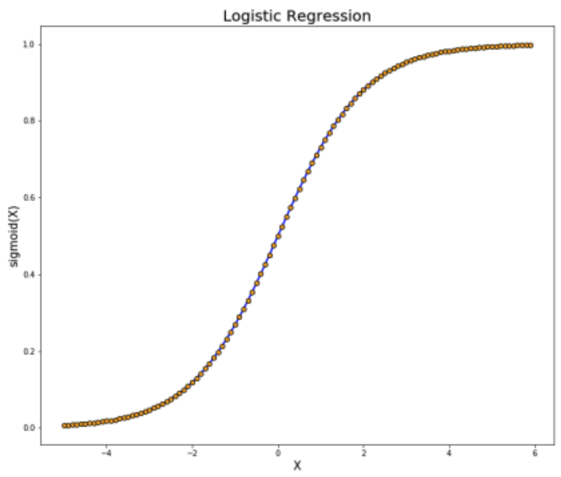


Figure 3.1: Sigmoid Function

The LR algorithm will limit the probability values to lie between 0 and 1 according to 3.5 ($P_1 \to 0$. as the right-hand side approaches $-\infty$, and $P_1 \to 1$ as it approaches

$+\infty$). The coefficients in the LR model are obtained by minimizing the log-likelihood function, which is the sum of the logarithms of the estimated probabilities Stoltzfus (2011). In order to build an accurate LR model, two different sets of data are used - a training set and a testing set. The training set is used to establish the LR model, while the testing set is used to evaluate the model's accuracy in predicting categorical values. By using these two different sets of data, it is possible to create a model that can accurately predict outcomes based on independent variables. This process is important for ensuring the model is accurate and can be used to predict outcomes in real-world scenarios.

3.2.2 CatBoost classifier

CatBoost Classifier stands for gradient boosting with categorical feature support is a gradient boosting framework that utilizes an innovative algorithm to improve the performance of machine-learning models. The CatBoost algorithm was introduced by Prokhorenkova, Gusev, Vorobev, Dorogush, and Gulin (2018). It is an algorithm in which reinforcement technique is used under ensemble learning methods. The algorithm incorporates a novel method for processing categorical features, significantly improving over the traditional one-hot encoding method. The CatBoost algorithm uses gradient boosting on decision trees, and its objective function is a combination of the log loss and L2 regularization. The log loss measures the difference between the predicted and actual values of the target variable. At the same time, the L2 regularization helps to prevent overfitting by adding a penalty term to the objective function. CatBoost's approach to handling categorical features is known as categorical encoding. It involves assigning a numerical value to each category in the feature, based on its impact on the target variable. The effect is calculated using a metric known as the target statistic, which is the mean value of the target variable for each category. This approach allows CatBoost to handle categorical features with high cardinality, which would otherwise be difficult to encode using one-hot encoding Prokhorenkova et al. (2018). The formula for the objective function of

CatBoost is as follows Prokhorenkova et al. (2018):

$$obj = \mathrm{sum}(\log(1 + \exp(-y_i * F(x_i))) + \lambda * G(F)/2) \tag{3.6}$$

where y_i is the target value for the i^{th} observation, $F(x_i)$ is the predicted value for the i^{th} observation, λ is the $L2$ regularization parameter, and $G(F)$ is the second derivative of the objective function with respect to F.

CatBoost is a powerful and versatile algorithm that can be effective for a wide range of classification problems, particularly those with imbalanced datasets. Its ability to handle categorical features, built-in feature scaling, robustness to noisy data, automatic handling of class imbalance, and regularization techniques make it a good choice for many real-world applications.

Handling categorical features: CatBoost can handle categorical features more effectively than many other algorithms. It uses a variant of gradient boosting that can naturally handle categorical features without the need for one-hot encoding, which can be particularly useful when dealing with imbalanced datasets.

Built-in feature scaling: CatBoost automatically scales features, which can be helpful when dealing with imbalanced datasets that have features with different scales.

Robust to noisy data: CatBoost is designed to handle noisy data and can tolerate missing or incorrect values.

Automatic handling of class imbalance: CatBoost includes an option to automatically handle class imbalance, using a combination of under-sampling, over-sampling, and gradient-based balancing.

Regularization techniques: CatBoost includes several regularization techniques, such as L1 and L2 regularization, that can help prevent overfitting and improve generalization performance on imbalanced datasets.

In conclusion, CatBoost is a robust gradient boosting framework that offers significant improvements over traditional machine learning algorithms Hancock and Khoshgoftaar (2020). Its innovative approach to handling categorical features makes it an ideal choice for datasets with high cardinality, and its objective function incorporating log loss and L2 regularization helps prevent overfitting. The CatBoost algorithm can run on the GPU outside the CPU when desired. It can also work with missing data on the dataset Punmiya and Choe (2019). The CatBoost algorithm can be applied to both classification problems and regression problems. CatBoost has free access as open source. All documentation and reference papers of the algorithm are available on https://CatBoost.ai/en/docs**CatBoost Official Website**.

3.2.3 XGBoost classifier

Extreme gradient boosting (XGBoost) is a powerful and efficient machine learning algorithm used for binary classification problems. It is an ensemble learning method that utilizes the gradient boosting framework Wang, Deng, and Wang (2020). The algorithm consists of two main components Gentek (2022) :

1. **Gradient Boosting:** This component involves the creation of a series of decision trees iteratively. The first decision tree is created with the entire dataset. The subsequent trees are created on the residual errors of the previous tree. This process continues until a pre-defined number of trees is reached.
2. **Regularization:** This component involves the use of regularization techniques to prevent over-fitting. Regularization techniques penalize complex models and help in selecting the most important features.

The objective function of XGBoost for binary classification is defined as follows Chen and Guestrin (2016):

$$\text{obj} = \sum_{i=1}^{n} \text{loss}(y_i, \hat{y} * i) + \sum *j = 1^m \Omega(f_j) \tag{3.7}$$

where:

- n is the total number of samples in the dataset.
- m is the total number of trees in the model.
- y_i is the true label of the i^{th} sample.
- $\hat{y}_i$ is the predicted label of the i^{th} sample.
- f_j is the j^{th} tree in the ensemble.
- $\text{loss}(y_i, \hat{y}_i)$ is the loss function used to measure the difference between the true label and the predicted label.

In XGBoost, L1 regularization (Lasso) and L2 regularization (Ridge) are used. The split finding process in XGBoost involves finding the optimal split points for each feature in the dataset. The algorithm uses a gradient-based approach to find the optimal split points. For each feature, the algorithm calculates the gradient and the Hessian of the loss function. The gradient measures the direction of steepest descent, while the Hessian measures the curvature of the function. The algorithm then uses these values to calculate the optimal split point for the feature Chen and Guestrin (2016). In conclusion, XGBoost is a powerful and efficient algorithm for binary classification problems. It utilizes gradient boosting and regularization techniques to prevent overfitting and select the most important features. The split finding and tree pruning processes further improve the performance of the model.

3.2.4 Random forest classifier

The random forest (RF) algorithm has gained popularity for its effectiveness in binary classification. It is an ensemble learning method that builds a collection of decision trees and returns the mode of the predictions of the individual trees as the final prediction. The RF algorithm is a combination of two techniques, ”bagging” and ”random subspace method”, which are used to reduce the variance of the predictions and improve the accuracy of the classification. The algorithm works by creating multiple decision trees with different random subsets of the training data and features. Each decision tree in the forest is created using a randomly selected subset of the training data and a random subset of the features. This randomness helps to prevent overfitting, which is when the model fits too closely to the training data and performs poorly on new, unseen data. To classify a new sample, the algorithm passes the sample through each decision tree in the forest and records the predicted class. The mode of the predicted classes is then returned as the final prediction.

The probability of a sample belonging to class 1 is given by:

$$p(y = 1|X) = \frac{1}{1 + e^{-f(X)}} \tag{3.8}$$

where $f(X)$ is the weighted sum of the outputs of the individual trees in the forest.

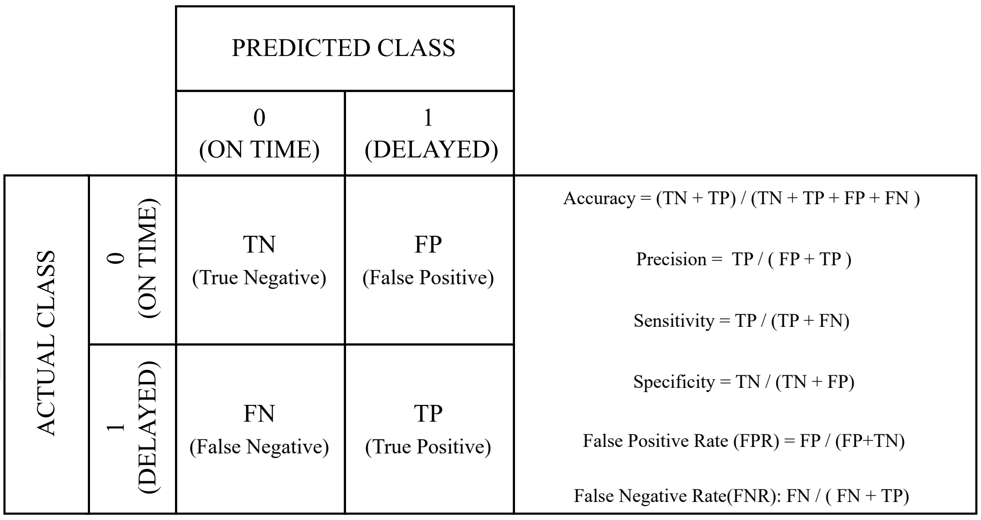
In this formula, X is the input data, y is the binary class label (either 0 or 1), and $f(X)$ is the weighted sum of the outputs of the individual trees in the forest. The sigmoid function is used to bound the output of the formula between 0 and 1, which gives the probability of the sample belonging to class 1.

The RF classifier is a powerful and versatile algorithm that can be used for a wide range of binary classification problems. It is particularly useful when dealing with high-dimensional data or when it is difficult to choose a single, optimal feature set. The RF algorithm is an extension of the decision tree algorithm. In a decision tree, the algorithm creates a tree structure where each node represents a decision based on a feature in the input data. The decision tree algorithm can be prone to overfitting, which is when the model fits too closely to the training data and performs poorly on new, unseen data. To prevent overfitting, the RF algorithm creates multiple decision trees with different random subsets of the training data and features. Each decision tree is trained on a different subset of the data, and the final prediction is made by combining the predictions of all the trees in the forest. The RF algorithm is widely used in various applications, including finance, healthcare, and marketing. It is a popular choice because it is easy to implement and can handle both numerical and categorical data. However, the RF algorithm can be computationally expensive and may not perform as well on small datasets. Additionally, the algorithm can be prone to bias towards features with more levels and can be difficult to interpret due to the complexity of the decision trees.

In the case of imbalanced binary classification problems, where one class has a significantly smaller number of instances compared to the other class, the algorithm can improve performance by addressing two common issues that arise in such problems: bias and overfitting. Firstly, RF can address bias by balancing the training data for each decision tree. During the training process, each decision tree is constructed using a randomly sampled subset of the majority class that is equal in size to the minority class. This means that each decision tree is trained on a balanced dataset, which helps to prevent the model from being biased towards the majority class. Secondly, RF can reduce the impact of noisy features and prevent overfitting by using only a subset of features for each decision tree. This ensures that each tree focuses on different aspects of the data, reducing the impact of any noisy or irrelevant features that may be present in the dataset. Also, the RF algorithm is a powerful algorithm for imbalanced binary classification problems because it can address bias, reduce overfitting, and improve overall model performance by leveraging an ensemble of decision trees trained on balanced subsets of the data. Also, the RF classifier is a valuable algorithm for binary classification problems. It is a versatile algorithm that can handle a wide range of data types, and it is widely used in various industries. While it has some limitations, it is an effective algorithm that can help improve the accuracy of classification tasks.

3.2.5 Classification metrics

Confusion Matrix: The number of cases where positive observations are predicted to be positive is shown in the true positive (TP) section. The number of cases where observations with a negative outcome are predicted as negative is shown in the true negative (TN) section. Both True Positive and True Negative results are values that indicate successful prediction. The number of cases where observations with a positive outcome are predicted as negative is shown in the false negative (FN, Type 2 Error) section. The number of cases where observations with a negative outcome are predicted as positive is shown in the false positive (FP, Type 1 Error) section. Both False Positive and False Negative results are values that indicate unsuccessful predictions Liang (2022).



		PREDICTED CLASS		
		0 (ON TIME)	1 (DELAYED)	
ACTUAL CLASS	0 (ON TIME)	TN (True Negative)	FP (False Positive)	Accuracy = (TN + TP) / (TN + TP + FP + FN) Precision = TP / (FP + TP) Sensitivity = TP / (TP + FN)
	1 (DELAYED)	FN (False Negative)	TP (True Positive)	Specificity = TN / (TN + FP) False Positive Rate (FPR) = FP / (FP+TN) False Negative Rate(FNR): FN / (FN + TP)

Figure 3.2: Confusion Matrix and Performance Metrices

Accuracy: Accuracy value is calculated by the ratio of the sum of the numbers in the fields (TP, TN) expressing successful predictions in the model to the size of the total data set.

Precision: Precision value is a metric that shows how many of the observations predicted as positive are actually positive.

True positive rate (TPR) and false positive rate (FPR) are metrics that are commonly used to evaluate the performance of binary classification models, especially when dealing with imbalanced datasets. TPR is also known as sensitivity or recall, and is defined as the number of true positives divided by the sum of true positives and false negatives. TPR measures the proportion of actual positive instances that are correctly identified by the classifier. FPR is defined as the number of false positives divided by the sum of false positives and true negatives. FPR measures the proportion of actual negative instances that are incorrectly identified as positive by the classifier. In imbalanced binary classification problems, where the number of instances in the minority class is much smaller than in the majority class, TPR and FPR are good metrics for evaluating the performance of classifiers because they provide insight into how well the classifier is able to identify positive instances while avoiding false positives. TPR is important because it measures how well the classifier is able to identify the minority class, which is often the class of interest in imbalanced binary classification problems. High TPR indicates that the classifier is correctly identifying a large proportion of the positive instances, which is crucial in applications such as fraud detection or disease diagnosis, where false negatives can have serious consequences. FPR is important because it measures how well the classifier is able to avoid false positives, which can be a major problem in imbalanced binary classification problems. High FPR can result in a large number of false positives, which can lead to unnecessary costs or actions, and can also reduce the overall performance of the classifier. In conclusion, TPR and FPR are important metrics for evaluating the performance of classifiers on imbalanced binary classification problems because they provide insight into how well the classifier is able to identify positive instances and avoid false positives. High TPR is important for correctly identifying the minority class, while low FPR is important for avoiding false positives.

F1 & F2 Scores : F1 and F2 scores are metrics that are commonly used to evaluate the performance of binary classification models, especially when dealing with imbalanced datasets. The F1 score is a harmonic mean of precision and recall, and is calculated as:

$$F1_{score} = 2 * (precision * recall)/(precision + recall) \tag{3.9}$$

where precision is the number of true positives divided by the sum of true positives and false positives, and recall is the number of true positives divided by the sum of true positives and false negatives. The F1 score ranges from 0 to 1, with 1 indicating perfect precision and recall, and 0 indicating poor performance.

The F2 score is a variant of the F1 score that places more emphasis on recall than precision. It is calculated as:

$$F2_{score} = (1 + beta^2) * (precision * recall)/(beta^2 * precision + recall) \tag{3.10}$$

where beta is a parameter that determines the relative weight given to recall. When beta is set to 2, the F2 score places more emphasis on recall than the F1 score, making it more suitable for imbalanced dataset where the focus is on correctly identifying the minority class.

The F1 and F2 scores are good metrics for imbalanced binary classification because they take into account both precision and recall, which are important for evaluating the performance of classifiers on imbalanced dataset. Precision measures the ability of a classifier to correctly predict the positive class, while recall measures the ability of the classifier to identify all positive instances. In imbalanced dataset, where the number of instances in the minority class is much smaller than in the majority class, it is important to correctly identify as many positive instances as possible. The F1 and F2 scores provide a balance between precision and recall, which is particularly useful in imbalanced binary classification problems.

ROC-AUC: The receiver operating characteristic (ROC) curve is a graphical representation of the binary classification system's performance. The ROC curve plots TPR against FPR for varying classification thresholds. The ROC-Area Under the Curve (AUC) is the numerical measure of the ROC curve's performance. AUC ranges from 0 to 1, with 1 indicating a perfect classifier and 0.5 indicating a random guess.

In imbalanced classification problems, there is a considerable class imbalance between the positive and negative classes. In such cases, the AUC is a more suitable metric than accuracy, as accuracy can be misleading. AUC is insensitive to class imbalance, making it a reliable metric for evaluating the performance of a binary classifier. A high AUC score indicates that the model can distinguish between the positive and negative classes effectively.

3.2.6 Parameter tuning

In machine learning, binary classification is the task of classifying instances into one of two classes. When building a binary classification model, there are several parameters that need to be tuned to achieve optimal performance. One important parameter is the learning rate, which determines how quickly the model adapts to new data. A high learning rate may cause the model to converge too quickly and miss important features, while a low learning rate may cause the model to converge too slowly and take longer to train. Another critical parameter is the regularization parameter, which controls the amount of regularization applied to the model. Regularization helps to prevent overfitting by adding a penalty term to the loss function, but too much regularization can lead to under-fitting. Finally, the choice of algorithm and hyper-parameters can also greatly impact the performance of the model. Popular algorithms for binary classification problems include LR, decision trees, and support vector machines. Overall, selecting the right machine learning parameters is crucial for achieving high accuracy and avoiding overfitting or under-fitting.

To optimize the parameters for the **XGBoost**, **CatBoost**, and **RF** algorithms, a rigorous process is implemented, whereby three parameters are optimized with three different values for each algorithm. The optimization is carried out in order to determine the best possible values for the parameters that will allow each algorithm to perform optimally. The possible values for the three parameters of CatBoost, XGBoost, and RF are shown in Table 3.1, which depicts the range of values that must be tested in order to find the optimal configuration for each algorithm. By optimizing the parameters of these algorithms, it can be ensured that the models built can perform well across a range of different scenarios, rendering them a valuable tool for data analysis and decision-making.

1. $max - depth$: This parameter in XGBoost sets the maximum depth of each decision tree in the model. This parameter controls the complexity of the

Parameter	**Values**
max-depth	[6,5,7]
learning-rate	[0.4, 0.3, 0.2]
n-estimators	[100,200,300]

Table 3.1: CatBoost, XGBoost and RF Parameter Grid

individual trees, and can be used to prevent overfitting by limiting the number of splits in each tree. A higher value for $max-depth$ allows the model to capture more complex interactions between features, but may also lead to overfitting.

2. $learning-rate$: This parameter controls the step size at each iteration while moving toward a minimum of the loss function. A lower learning rate allows the model to take smaller steps during training, which can help prevent overfitting and improve generalization performance. However, using a lower learning rate may also require more iterations to reach convergence.
3. $n-estimators$: This parameter sets the number of trees to include in the final model. Increasing the number of trees can improve model performance, but may also increase the risk of overfitting. It is important to balance the number of trees with other hyper-parameters, such as $max-depth$ and $learning-rate$, to ensure optimal performance of the model.

The performance of the classifier algorithms can be improved by adjusting their parameters. One approach to achieving this is using together grid search and k-fold validation techniques. Grid search is a hyperparameter tuning technique used in machine learning to systematically search for a given model's best combination of hyperparameters. The technique then systematically evaluates the model's performance using different combinations of these hyperparameter values. Cross-validation is a technique used to validate the model's performance on unseen data by splitting the data set into training and validation sets multiple times and averaging the results. GridSearchCV is a method of the popular Scikit-Learn library in Python. It uses together a grid search method with a k-fold validation technique. This technique permits the exploration of the optimal set of hyperparameters by testing all conceivable combinations within a predefined range of hyperparameters. However, it is essential to note that GridSearchCV can be computationally expensive, especially when many hyperparameters and a large dataset are present. Therefore, a reasonable range of hyperparameters should be chosen, and cross-validation should be employed to avoid overfitting.

3.2.7 Feature importance

Feature selection approaches identify the most critical features that enhance the model's performance and reduce the complexity of the models. One of the most promising approaches is shapley additive explanations (SHAP), a game-theoretic method that explains the contribution of each feature to the predicted outcome of a model. This thesis explores utilizing SHAP values to interpret feature importance in three prominent binary classification algorithms: XGBoost, RF, and CatBoost.

SHAP are based on the concept of Shapley values, which is a concept in cooperative game theory that measures the contribution of each player to a coalition. In machine learning, SHAP values can explain each feature's contribution to a model's predicted outcome. The SHAP values for XGBoost, RF, and CatBoost can be calculated using the SHAP package in Python.

According to a study by Lundberg and Lee (2017), SHAP values can explain the feature importance of XGBoost, RF, and CatBoost. They showed that SHAP values provide a more accurate and consistent way to interpret the feature importance of XGBoost compared to other methods. According to a study conducted by **?**, SHAP values can be used to explain the feature importance of CatBoost. They showed that SHAP values provide a more accurate way to interpret the feature importance of CatBoost compared to other methods. The SHAP values provide a unified and consistent way to interpret the output of any machine learning model.

SHAP values were used to interpret the feature importance of three popular binary classification algorithms, XGBoost, CatBoost, and RF. Utilizing SHAP values can facilitate a more comprehensive understanding of a machine learning model's decision-making. Step-wise selection approach was applied using SHAP values of 0.01 criteria. This process helped us identify and select the most important features for predicting delivery delays.

The step-wise approach was employed for the LR model with a p-value criterion of 0.01. This approach adds features to the LR model while monitoring their significance levels. In every iteration of the step-wise process, p-values of features are examined and compared against the predefined significance level of 0.01. A distinct set of statistically significant features for the LR model is determined. These selected attributes can differ with each step and comprise duration variables of previous business operations leading up to the current operation and details about sender and receiver cross-docks.

4. APPLICATION

4.1 Modelling of Business Operations

In e-commerce, a customer's order undergoes complex operational processes before reaching its final destination. These processes involve multiple intricate steps, including order receiving, unloading, handling, and truck loading. The number of steps may vary depending on companies' policies and procedures. Companies equipped with advanced technology instantly process every delivery transaction into their databases, enabling efficient tracking of the delivery process. This real-time recording of transactions enhances the overall customer experience and provides insights into operational performance. Companies can optimize processes, streamline operations, and achieve increased efficiency and cost savings by analyzing recorded data.

Successful delivery of an e-commerce order requires a well-coordinated effort involving several operational processes. Companies must continually improve their technological capabilities and implement streamlined procedures to ensure the delivery process is efficient and reliable. The use of real-time transaction recording can significantly enhance the efficiency of the delivery process, resulting in increased customer satisfaction and a competitive edge in the market.

An application has been developed for this study using data from a logistics company in Turkey. The company uses barcodes to identify deliveries, which allows them to track shipment information in the system and access details as needed. Each barcode goes through several processes and is recorded in the system. The system provides access to various information from the database, such as transaction details, delivery type, delivery date, and sender address of the shipment. It also enables operational performance analysis, determination of shipment frequency areas, and analysis of temporal information such as the day, month, and day of the week when the shipments are processed. Retrospective access can also be provided to review delivery problems experienced in the past.

The total time spent in operational processes is calculated by summing the durations of each transaction step. This research aims to utilize machine learning models, trained on data from both on-time and delayed deliveries, to predict the probability of delays at various operational milestones for recently accepted orders. By providing the model with the time spent in each step since the placement of the shipment order, the delay probability is calculated based on the current step. As each milestone is reached, the probability of delay is continuously updated to enhance the accuracy and consistency of the prediction. Throughout the study, the terms ”milestone,” ”step,” ”process,” and ”stage” are used interchangeably to refer to specific actions or operational delivery processes, such as order receiving, unloading, and handling. Detailed information about the shipments' delivery processes from when the order is received until the shipment is delivered is given below.

1. The delivery is picked up from the sender's warehouse and brought to the initial unit. This process, the first step, is called ”Delivery Collection”.
2. ”Loading-Initial Unit” refers to the efficient and safe loading of deliveries onto a vehicle in the initial unit.
3. “Transferring to First Transfer” refers to the movement of deliveries from the initial unit to the first transfer unit.
4. “Unloading-First Transfer” refers to the process of unloading deliveries from the vehicle in the first transfer unit.
5. “Handling-First Transfer” is the process that shipments undergo until they are unloaded from the vehicle and loaded back into the vehicle in the first transfer unit. This step includes labeling, measuring, weighing, scanning, and sorting.
6. ”Loading-First Transfer” refers to loading deliveries into the vehicle in the first transfer unit.
7. ”Transferring to Second Transfer” describes the movement of deliveries from

the first transfer unit to the second transfer unit.

8. The process of unloading deliveries from the vehicle in the second transfer unit is called ”Unloading-Second Transfer”.
9. “Handling-Second Transfer” is the process that shipments undergo until they are unloaded from the vehicle and loaded back into the vehicle in the second transfer unit. This includes labeling, measuring, weighing, scanning, and sorting.
10. Loading deliveries into the vehicle in the second transfer unit is identified as ”Loading-Second Transfer”.
11. “Transferring to Terminal Unit” refers to the movement of deliveries from the second transfer unit to the terminal unit.
12. ”Unloading - Terminal unit” designates the process of unloading deliveries from a vehicle at the terminal unit.
13. ”Handling-Terminal Unit” is the process that shipments go through until they are unloaded from the vehicle and received by the courier.
14. The step ”Handling-Courier” refers to the process from when the courier picks up shipments until the delivery process begins.
15. ”Delivery” refers to the act of delivering shipments to the recipient.

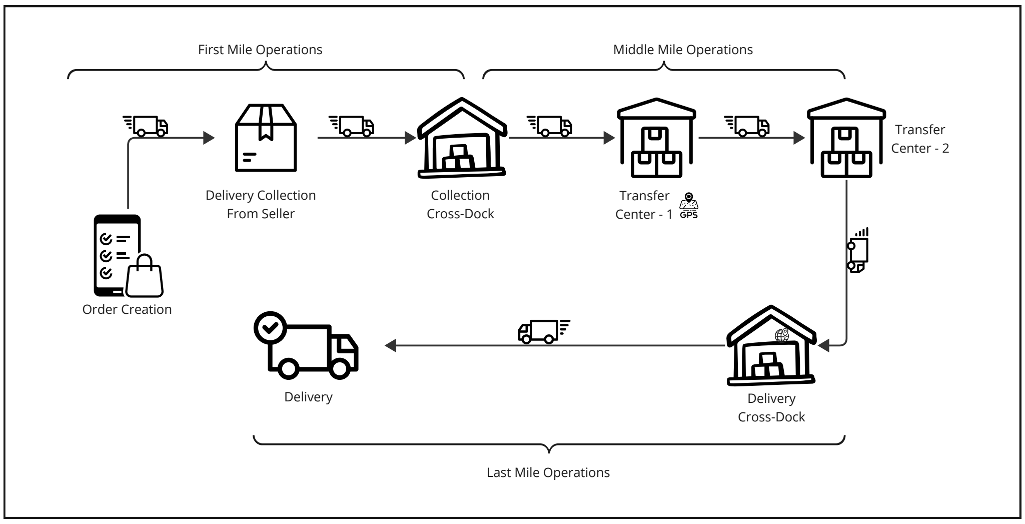


Figure 4.1: Business Operations Schema

1. **First Mile Operations**

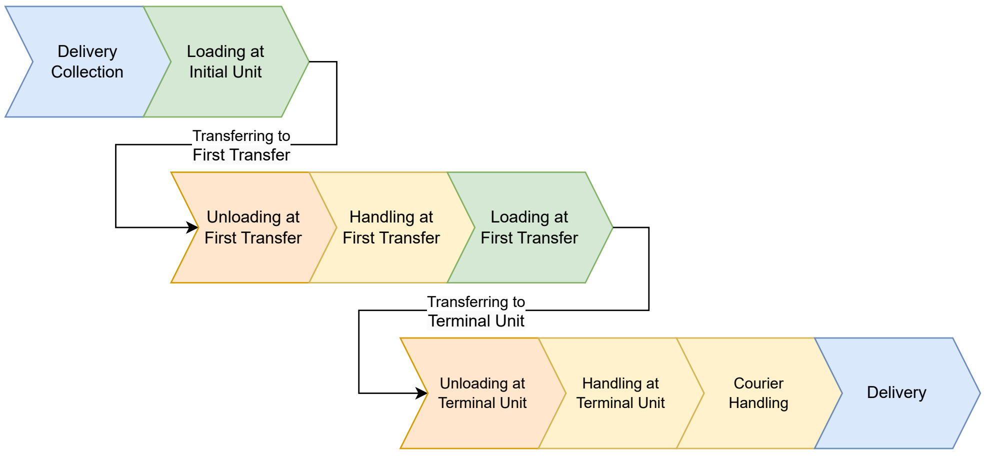
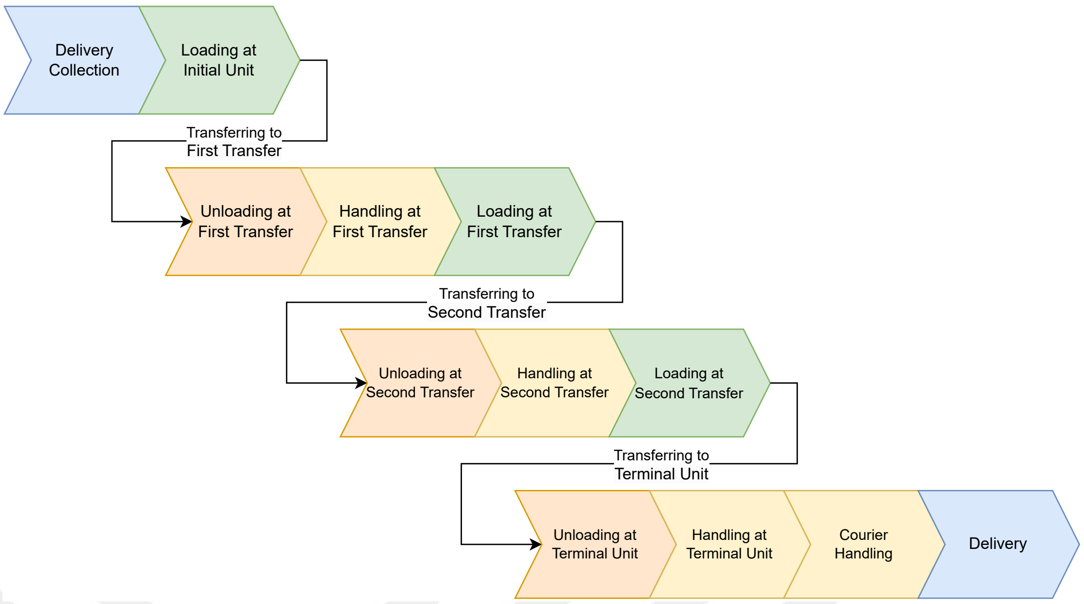

Delivery Collection
Loading at Initial Unit
Transferring to First Transfer
Unloading at First Transfer
Handling at First Transfer
Loading at First Transfer
Transferring to Terminal Unit
Unloading at Terminal Unit
Handling at Terminal Unit
Courier Handling
Delivery

Figure 4.2: Step-Flow of 11-Step

   (a) Acceptance of the delivery by the logistics company and arrival of the delivery at the initial logistics.
   (b) Loading the shipment into the first vehicle.
   (c) Approval of the delivery to leave from the initial unit

2. **Middle-Mile Operations**
   (a) Arrival of the delivery at the 1st transfer unit.
   (b) Unloading of the delivery from the vehicle at 1st transfer unit.
   (c) Loading the delivery into the vehicle for the next unit.
   (d) Approval of the delivery to leave from the current transfer unit.
   (e) Arrival of the delivery at the 2nd transfer unit.
   (f) Unloading of the delivery from the vehicle at 2nd transfer unit.
   (g) Loading the delivery into the vehicle for the next unit.
   (h) Approval of the delivery to leave from the 2nd transfer unit.

3. **Last-Mile Operations**
   (a) Arrival of the delivery at the terminal logistics unit.
   (b) Unloading of the delivery from the vehicle at terminal unit.
   (c) Taking custody of the delivery by the courier at terminal unit.
   (d) Approval of the courier to start delivery operation.
   (e) Delivery process.



Delivery Collection
Loading at Initial Unit
Transferring to First Transfer
Unloading at First Transfer
Handling at First Transfer
Loading at First Transfer
Transferring to Second Transfer
Unloading at Second Transfer
Handling at Second Transfer
Loading at Second Transfer
Transferring to Terminal Unit
Unloading at Terminal Unit
Handling at Terminal Unit
Courier Handling
Delivery

Figure 4.3: Step-Flow of 15-Step

4.2 Data Analysis

Analyzing the characteristics and structure of data is crucial in selecting an appropriate machine learning model to ensure improved results. Therefore, data analysis plays a significant role in this study. The dataset utilized was obtained from a logistics company and consisted of transactional information related to delivery. The data focuses explicitly on shipments collected and distributed within the same city, with promised delivery within one-day intervals. The dataset includes delivered shipments and spans eight months. The logistics company has provided transactional data for analysis. Successful deliveries are when items are delivered to the customer on or before the promised date. On the other hand, unsuccessful deliveries occur when items are delivered after the promised date, resulting in delays. Delayed deliveries can negatively affect customer satisfaction and erode trust in the company. Customers expect their packages to arrive on time, making timely delivery a top priority for both customers and cargo companies. Ensuring timely delivery helps keep customers satisfied and builds trust in the company's ability to fulfill its promises.

The target variable in this dataset is "on-time delivery," which has two tags, successful and unsuccessful. Successful deliveries are those that are delivered on time, while unsuccessful deliveries are those that are delayed. Upon evaluating the success rates of deliveries, it was determined that approximately 97% of shipments were delivered on time, while the remaining 3% were delayed. Although the rate of delayed shipments seems low, it can be costly for logistics companies. Therefore, companies desire to minimize the number of delayed deliveries. Delayed deliveries can lead to customer dissatisfaction, resulting in negative reviews, decreased customer loyalty, and potential loss of business. Furthermore, they can harm a company's reputation, making attracting new customers and maintaining existing relationships more challenging. As a result, on-time delivery is critical for logistics companies to maintain customer satisfaction and brand image.

Shipments go through several stages before they are delivered to the recipient. Data analysis was conducted to examine the frequency of different delivery types based on the number of steps involved. Figure 4.4 presents a histogram that illustrates the delivery distribution based on the number of steps. The x-axis represents the number of steps in the delivery process being analyzed, while the y-axis represents the percentage of observations of deliveries at each operation step. The findings revealed that deliveries with 11 and 15 steps dominated the dataset, accounting for more than 80% of the total deliveries combined. Given that the majority of the dataset consisted of deliveries with either 11 or 15 steps, this project specifically focused on studying these two types of deliveries.

When examining the success rates of shipments with 11 and 15 steps, it was found that the success rate is 97.38% for the delivery type with 11 steps and 97.78% for the delivery type with 15 steps (see Figure 4.5). The data reveals that both delivery types had impressive on-time delivery rates, with the 15-step process showing a slightly higher success rate. These findings highlight that regardless of the number of steps involved, most shipments are delivered on time and have a high success rate.

A time-based analysis was conducted to observe changes in the number of delivered

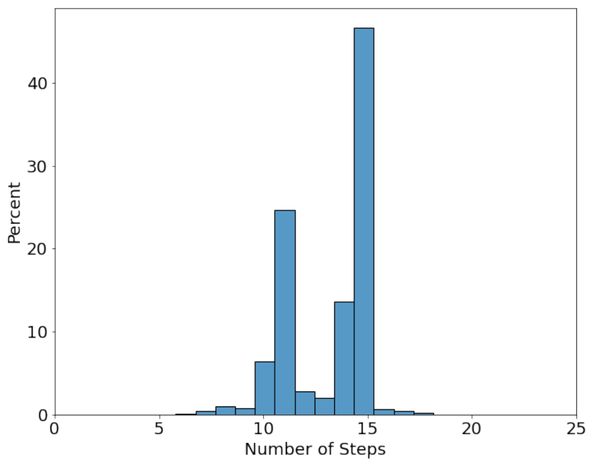


Figure 4.4: Delivery Frequency Histogram Respect to Number of Steps

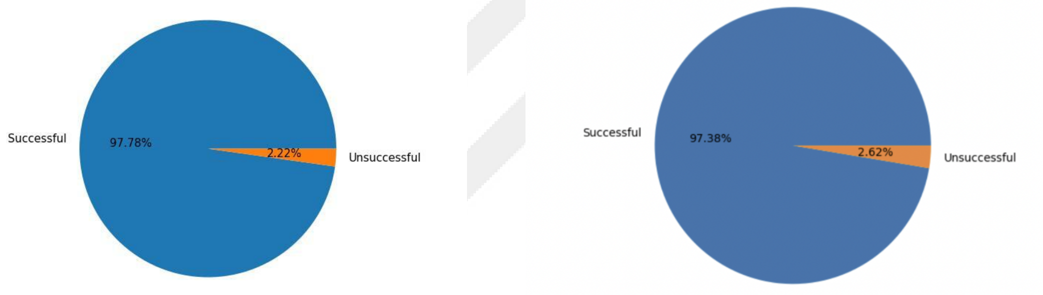


Figure 4.5: 15-Step and 11-Step Delivery Success Rate

shipments over time (see Figure 4.6). The first graph in Figure 3.4 displays the rate of delivered shipments by day of the week. The x-axis represents the days of the week, while the y-axis represents the ratio of delivered shipments for each specific day. The findings reveal that Tuesday had the highest percentage of deliveries, accounting for 25% of the total count. This result indicates that Tuesday is a hectic delivery day, potentially due to specific operational or logistical factors. The second graph in Figure 3.4 displays the monthly delivered shipment rate. The x-axis represents the month, while the y-axis represents the ratio of delivered shipments for each specific month. Additionally, when considering the monthly distribution, it was observed that month-5 stood out with a significant percentage of over 17.5%. The findings show that month-5 experiences a high volume of deliveries, possibly influenced by seasonal trends or specific events occurring during that month. Overall, these results highlight the importance of considering the day of the week and the month when analyzing the distribution of deliveries, as they can provide valuable insights for operational planning and resource allocation.

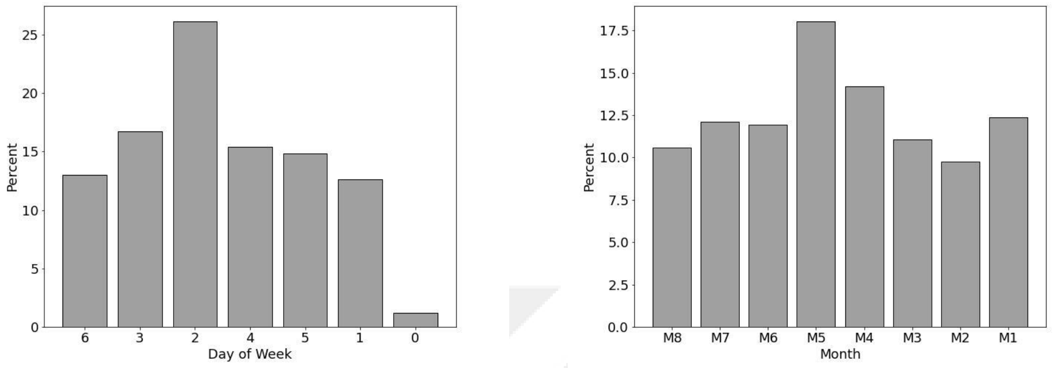


Figure 4.6: Day of Week and Monthly Percentages

In order to predict delivery delays, raw transactional data was extracted from databases. Rows containing invalid or null values were removed from the dataset. Table 4.1 provides an overview of the feature names and descriptions of the dataset used for modeling. It includes a comprehensive list of variables, such as delivery date, sender and receiver address details, and process-based information. This extensive dataset enables a detailed analysis of various features that may contribute to delivery delay prediction. Referring to Table 3.1 throughout the analysis ensures accurate identification and understanding of the specific variables involved in the prediction process, facilitating insightful and precise results.

Statistical tests provide a systematic and objective way to assess the significance of relationships, differences, or patterns in data. Before training machine learning models, the t-test, Shapiro-Wilk, and Levene's tests were used during data analysis. These tests provide insights into differences between groups, the normality of data distributions, and the homogeneity of variances. Such insights can increase the reliability of the machine learning model. A t-test is necessary to evaluate the difference between the durations of successful and unsuccessful shipments. However, since the t-test is a parametric test, it is crucial to meet the prerequisites before

Feature Name	Description
delivery_id	A unique identifier assigned to each delivery within dataset, enabling tracking and traceability of the delivery throughout the entire process.
delivered_date	This feature indicates the specific date on which a shipment was delivered to final recipient
ontime_delivery	This feature indicates whether a delivery was delivered within the promised timeframe or not.
transaction_type	This feature indicates the delivery-related transaction type.
transaction_date	This feature denotes the specific date on which a delivery-related transaction took place.
next_transaction_date	This feature indicates the date when the next shipment-related transaction occur.
current_unit_id	This feature represents the unique identifier assigned to the current operational unit.
initial_unit_id	This feature represents the unique identifier assigned to the initial operational unit of the delivery.
terminal_unit_id	This feature represents the unique identifier assigned to the terminal operational unit of the delivery.
receiver_town_id	This feature represents the unique identifier assigned to the specific town where the recipient's address is located for the delivery.
receiver_district_id	This feature represents the unique identifier assigned to the specific district where the recipient's address is located for the delivery.
sender_town_id	This feature represents the unique identifier assigned to the specific town where the sender's address is located for the delivery.
sender_district_id	This feature represents the unique identifier assigned to the specific district where the sender's address is located for the delivery.

Table 4.1: Raw Features with Descriptions

conducting it. Assumptions of parametric tests:

- Independence: The observations within each group or sample are assumed to be independent of each other.
- Normality: The data within each group or sample are assumed to follow a normal distribution.
- Equal Variances (Homoscedasticity): The variances of the data in each group or sample are assumed to be equal.

The Shapiro-Wilk Test: It is used to verify the normality assumption of a dataset and determine whether the data adhere to a normal distribution (King & Eckersley, 2019). Performing the Shapiro-Wilk test, one can assess whether the normality assumption is valid. Table 4.2 shows the results of the Shapiro-Wilk test for two sets of data: unsuccessful deliveries (U) and successful deliveries (S), which include 15 steps. This statistical test is used to assess the normality assumption of a dataset. Each row in the table represents a different variable or aspect of the delivery process. The columns are as follows:

- **Step**: This column represents the specific aspect of the delivery process being analyzed.
- **Statistic(U)**: This column displays the test statistic obtained from the Shapiro-Wilk test for the unsuccessful deliveries data. The test statistic measures how well the data follows a normal distribution.
- **P-Val(U)**: This column shows the p-value associated with the Shapiro-Wilk test for the unsuccessful deliveries data. The p-value indicates the probability of obtaining the observed test statistic under the null hypothesis that the data is normally distributed. A low p-value suggests that the data significantly deviates from a normal distribution.
- **Statistic(S)**: This column presents the test statistic obtained from the Shapiro-Wilk test for the successful deliveries data.
- **P-Val(S)**: This column displays the p-value associated with the Shapiro-Wilk test for the successful deliveries data.

The resulting p-values for unsuccessful and successful deliveries were reported as 0.0, indicating a significant deviation from normal distribution. It means that the variables or aspects of the delivery process mentioned in the table violate the assumption of normality. These results indicate that the data for these variables in the 15-Step Deliveries (see Table 4.2) does not follow a normal distribution. This finding should be considered when performing further statistical analyses or making inferences about the delivery process. The Shapiro-Wilk Test results for the 11-Step Deliveries (see Table 4.3) case show that the reported p-values are 0.0. Results indicate that the data for unsuccessful and successful deliveries for all features significantly deviates from a normal distribution. The p-value associated with the

Step	**Statistic**$_U$	**P-Val**$_U$	**Statistic**$_S$	**P-Val**$_U$
delivery_collection	0.596593	0.0	0.761377	0.0
loading_initial_unit	0.388165	0.0	0.247901	0.0
transferring_to_first_transfer	0.392649	0.0	0.560539	0.0
unloading_first_transfer	0.301078	0.0	0.303354	0.0
handling_first_transfer	0.295184	0.0	0.431631	0.0
loading_first_transfer	0.888582	0.0	0.860591	0.0
transferring_to_second_transfer	0.515689	0.0	0.446126	0.0
unloading_second_transfer	0.218558	0.0	0.572347	0.0
handling_second_transfer	0.091720	0.0	0.257130	0.0
loading_second_transfer	0.454178	0.0	0.432277	0.0
transferring_to_terminal_unit	0.754058	0.0	0.913855	0.0
unloading_terminal_unit	0.069320	0.0	0.310602	0.0
handling_terminal_unit	0.668978	0.0	0.317719	0.0
handling_courier	0.080579	0.0	0.477249	0.0
delivery	0.374547	0.0	0.744414	0.0

Table 4.2: Shapiro-Wilk Test Result For 15-Step Deliveries

Shapiro-Wilk test reflects the probability of obtaining the observed test statistic under the null hypothesis that the data is normally distributed. A low p-value suggests that the data significantly deviates from a normal distribution.

Step	$Statistic_U$	$P - Val_U$	$Statistic_S$	$P - Val_U$
delivery_collection	0.616162	0.0	0.833881	0.0
loading_initial_unit	0.376409	0.0	0.253508	0.0
transferring_to_first_transfer	0.499927	0.0	0.551831	0.0
unloading_first_transfer	0.310946	0.0	0.322677	0.0
handling_first_transfer	0.343701	0.0	0.516066	0.0
loading_first_transfer	0.614499	0.0	0.621260	0.0
transferring_to_terminal_unit	0.838990	0.0	0.917739	0.0
unloading_terminal_unit	0.063672	0.0	0.398098	0.0
handling_terminal_unit	0.659194	0.0	0.317031	0.0
handling_courier	0.240005	0.0	0.465912	0.0
delivery	0.399377	0.0	0.729780	0.0

Table 4.3: Shapiro–Wilk Test Result For 11 Step Deliveries

Levene's Test: It assesses the homogeneity of variances among multiple groups or samples (Schultz, 1985). It determines whether the variability of a variable is consistent across different groups. Levene's test helps identify whether this assumption of homogeneity of variances holds. The table below displays the results of Levene's test for the variations in the 15-step deliveries. Each row represents a different delivery process. Levene's test is a statistical test that evaluates whether the variances of different groups are significantly different. The following is a summary of the results:

- **Step:** This column specifies the delivery process being analyzed.
- **Statistic:** This column displays the test statistic for Levene's test, measuring the difference in variances between the groups for each delivery process.
- **p-value:** This column shows the p-value associated with each Levene's test. The p-value indicates the probability of observing the given difference in variances (or a more extreme difference) if there were no true difference between the groups. A smaller p-value suggests stronger evidence against the null hypothesis of equal variances.

The p-value for "loading_second_transfer" (p = 0.607) surpasses the significance level of 0.05. The result indicates no significant difference in variances between the groups for those specific processes. For the remaining delivery processes, the p-values are extremely small (close to zero), indicating strong evidence to reject the null hypothesis of equal variances between the groups. The findings suggest that the variations in the delivery processes differ significantly between successful and unsuccessful deliveries. Notable variations exist in the delivery processes between successful on-time deliveries and late deliveries based on the results of Levene's test, The table provided displays the results of Levene's test for the 11-step deliveries. The results indicate that p-values are close to zero for all analyzed delivery processes, suggesting significant differences in variations between the two groups. These consequences underscore the critical role of variation between the different steps of the delivery process in determining the success or failure of deliveries.

Step	**Statistic**	***p*-value**
delivery_collection	58479.844594	$0.000000e+00$
loading_initial_unit	43554.436007	$0.000000e+00$
transferring_to_first_transfer	487.984892	$4.215860e-108$
unloading_first_transfer	48074.164540	$0.000000e+00$
handling_first_transfer	13213.668583	$0.000000e+00$
loading_first_transfer	530.649085	$2.229881e-117$
transferring_to_second_transfer	158.136468	$2.912571e-36$
unloading_second_transfer	1135.136981	$1.143163e-248$
handling_second_transfer	1380.149999	$7.867709e-302$
loading_second_transfer	0.263854	$6.074846e-01$
transferring_to_terminal_unit	17.716241	$2.564559e-05$
unloading_terminal_unit	1786.970802	$0.000000e+00$
handling_terminal_unit	78208.675201	$0.000000e+00$
handling_courier	2829.116673	$0.000000e+00$
delivery	42286.134329	$0.000000e+00$

Table 4.4: Levene Test Result 15-Step Deliveries

Step	**test_stat_var**	***p*-value_var**
delivery_collection	37847.196179	$0.000000e+00$
loading_initial_unit	19803.864786	$0.000000e+00$
transferring_to_first_transfer	61.867915	$3.681925e-15$
unloading_first_transfer	26715.107911	$0.000000e+00$
handling_first_transfer	506.171059	$5.069570e-112$
loading_first_transfer	6.947822	$8.392457e-03$
transferring_to_terminal_unit	5.761094	$1.638532e-02$
unloading_terminal_unit	971.109646	$6.164327e-213$
handling_terminal_unit	36389.901762	$0.000000e+00$
handling_courier	1892.109939	$0.000000e+00$
delivery	24138.400863	$0.000000e+00$

Table 4.5: Levene Test Result 11 Step Deliveries

The results of the Shapiro-Wilk and Levene's tests indicated that the duration data did not follow a normal distribution, and the variances significantly differed between successful and unsuccessful deliveries. In this scenario, the Mann-Whitney U test, a non-parametric test, was used as an alternative.

Mann-Whitney U Test: The test examines whether the distribution of ranked responses between the two samples being compared is significantly different (Gaddis & Gaddis, 1990). Table 4.6 and Table 4.7 summarize the results of the Mann-

Whitney U test.

The table has the following columns and descriptions:

- **Step:** This column specifies the delivery process being analyzed.
- ***t*-test:** This column displays the test statistic calculated for the Mann-Whitney U test. The test statistic measures the magnitude of the difference between the two groups for each delivery process.
- ***p*-value:** This column shows the p-value associated with each Mann-Whitney U test. The p-value indicates the probability of observing the given test statistic (or a more extreme statistic) if there were no true difference between the groups.

The results of the Mann-Whitney U test for the 15- and 11-step analyses indicate significant differences between on-time and late deliveries in all examined delivery processes. These findings are critical for improving the delivery process. Addressing the specific steps that require improvement can increase the overall efficiency of the delivery process, resulting in more successful and on-time deliveries.

Step	***U* test**	***p*-value**
delivery_collection	$9.667615e+09$	$0.000000e+00$
loading_initial_unit	$9.666146e+09$	$0.000000e+00$
transferring_to_first_transfer	$8.846912e+09$	$5.962391e-120$
unloading_first_transfer	$9.740254e+09$	$0.000000e+00$
handling_first_transfer	$8.532704e+09$	$1.134028e-42$
loading_first_transfer	$8.799840e+09$	$7.803675e-106$
transferring_to_second_transfer	$8.475443e+09$	$7.039166e-33$
unloading_second_transfer	$8.783325e+09$	$4.295874e-101$
handling_second_transfer	$8.181800e+09$	$2.921201e-03$
loading_second_transfer	$8.270567e+09$	$1.295036e-08$
transferring_to_terminal_unit	$8.133611e+09$	$1.325003e-01$
unloading_terminal_unit	$8.472120e+09$	$2.373473e-32$
handling_terminal_unit	$1.150577e+10$	$0.000000e+00$
handling_courier	$8.568324e+09$	$1.980293e-49$
delivery	$9.632425e+09$	$0.000000e+00$

Table 4.6: Mann-Whitney U Test 15-Step Deliveries

Step	U **test**	p**-value**
delivery_collection	2.300637e + 09	0.000000e + 00
loading_initial_unit	2.188142e + 09	3.157620e − 204
transferring_to_first_transfer	1.970059e + 09	1.714562e − 27
unloading_first_transfer	2.255427e + 09	1.791180e − 292
handling_first_transfer	2.220914e + 09	3.214739e − 245
loading_first_transfer	1.730556e + 09	1.087146e − 26
transferring_to_terminal_unit	1.808924e + 09	2.726567e − 04
unloading_terminal_unit	1.983532e + 09	1.406245e − 33
handling_terminal_unit	2.565889e + 09	0.000000e + 00
handling_courier	1.925384e + 09	7.778930e − 12
delivery	2.224396e + 09	8.506877e − 250

Table 4.7: Mann-Whitney U Test 11 Step Deliveries

T-Test: A t-test is a statistical tool employed to determine whether there is a significant difference between the means of two groups or samples (Kim, 2015). It is beneficial for identifying whether a particular variable varies significantly between different groups or classes.

Table 4.8 and Table 4.9 display the results of t-tests that compare two groups: "S" representing successfully on-time deliveries and "U" representing late deliveries. The first table represents the statistical results for 15-Step Deliveries, and the second table represents 11-Step Deliveries. Each row represents a different delivery process duration, and the columns provide information on the difference in duration means between the two groups, Pearson's correlation coefficient, p-value, and t-statistic.

The description of the columns is as follows:

- **Step:** This column specifies the delivery process being analyzed.
- **Difference** $(s - u)$**:** This column shows the difference in means between successfully on-time deliveries (S) and late deliveries (U) for each process. Negative values indicate that the mean for successfully on-time deliveries is lower than the mean for late deliveries.
- **Pearson's Correlation Coefficient:** This column presents the correlation

coefficient, which measures the strength and direction of the linear relationship between the two groups. In this case, it appears to be the correlation coefficient between the delivery process and the outcome variable (on-time vs. late delivery).

- ***p*-value:** This column shows the p-value associated with each t-test. The p-value indicates the probability of observing the given difference in means (or a more extreme difference) if there were no true difference between the two groups. A lower p-value suggests stronger evidence against the null hypothesis of no difference.
- ***t*-statistic:** This column displays the t-statistic, which measures the difference between the means of the two groups relative to the variation within each group. A larger absolute t-statistic indicates a more significant difference between the means of the two groups.

According to the data, all p-values are very small, indicating strong evidence to reject the null hypothesis of no difference between the two groups for each delivery process. Generally, successful deliveries have lower values than unsuccessful deliveries for all the analyzed processes, as suggested by the negative differences in means. The results are consistent for both 15-Step Deliveries and 11-Step Deliveries, implying a significant difference in the durations of means for successful and unsuccessful shipments per operational process.

The variables used are given in Table 4.10. The given data in the table 4.11 provides descriptive statistics for various steps related to delivery and handling processes. Each step is categorized as successful (S) or unsuccessful (U). The statistics include the mean (mu) and standard deviation (sigma) for both successful and unsuccessful steps, as well as the minimum (min) and maximum (max) values. The data is scaled such that the maximum value for unsuccessful steps equals 100 for each step. For the "delivery-collection" step, the mean for successful steps is 2.0903, while for unsuccessful steps, it is 5.8681. The standard deviation for successful steps is 0.0233, and for unsuccessful steps, it is 0.1275. The minimum value for successful steps is

Step	**difference(s-u)**	**Pearson's**	***p* value**	**t stat**
delivery	-378.2568	0.2108	0.0000	-192.3364
delivery_collection	-227.4914	0.2374	0.0000	-217.9893
handling_courier	-31.4386	0.0577	0.0000	-51.5816
handling_first_transfer	-55.3917	0.1274	0.0000	-114.5571
handling_second_transfer	-10.3134	0.0413	0.0000	-36.8989
handling_terminal_unit	-1061.2981	0.3003	0.0000	-280.8085
loading_first_transfer	-21.4339	0.0285	0.0000	-25.4717
loading_initial_unit	-93.8742	0.2277	0.0000	-208.6519
loading_second_transfer	-6.4447	0.0039	0.0006	-3.4373
transferring_to_first_transfer	-10.3515	0.0316	0.0000	-28.1890
transferring_to_second_transfer	-40.9689	0.0147	0.0000	-13.1451
transferring_to_terminal_unit	-6.3968	0.0076	0.0000	-6.7967
unloading_first_transfer	-372.3869	0.2389	0.0000	-219.4409
unloading_second_transfer	-30.1467	0.0400	0.0000	-35.7513
unloading_terminal_unit	-24.1805	0.0481	0.0000	-42.9397

Table 4.8: T-Test Results For 15-Step Deliveries

Step	**difference(s-u)**	**Pearson's**	***p*-value**	**t test**
delivery	-424.0542	0.2223	0.0000	-144.2898
delivery_collection	-259.5062	0.2661	0.0000	-174.6569
handling_courier	-28.3243	0.0627	0.0000	-39.7349
handling_first_transfer	-105.3588	0.0560	0.0000	-35.4804
handling_terminal_unit	-1045.8869	0.2887	0.0000	-190.8360
loading_first_transfer	23.3652	0.0073	0.0000	4.6504
loading_initial_unit	-85.2365	0.2170	0.0000	-140.6666
transferring_to_first_transfer	-5.9697	0.0173	0.0000	-10.9711
transferring_to_terminal_unit	0.0578	0.0001	0.9718	0.0353
unloading_first_transfer	-444.1967	0.2501	0.0000	-163.4303
unloading_terminal_unit	-25.7932	0.0507	0.0000	-32.1233

Table 4.9: T-Test Results For 11-Step Deliveries

1.7638, and for unsuccessful steps, it is 2.4890. The maximum value for successful steps is 42.4451; for unsuccessful steps, it is 100.0000. Similarly, for each step such as "loading_initial_unit," "transferring_first_transfer," "unloading_first_ transfer," "handling_first_transfer," "loading_first_transfer," "transferring_terminal_unit," "unloading_terminal_unit," "handling_terminal_unit," "handling_courier," and "delivery," descriptive statistics including mean, standard deviation, minimum, and maximum values are provided for both successful and unsuccessful steps, scaled

Features		
barcode	delivered_date	delivery_success
delivery_date_promised	teslimat_basarisi_time	receiver_xdock
sender_xdock	receiver_town	sender_town
receiver_district	sender_district	
delivery_collection	loading_initial_unit	transferring_to_first_transfer
unloading_first_transfer	handling_first_transfer	loading_first_transfer
transferring_to_terminal_unit	unloading_terminal_unit	handling_terminal_unit
handling_courier	delivery	
delivery_collection_hour	loading_initial_unit_hour	transferring_to_first_transfer_hour
unloading_first_transfer_hour	handling_first_transfer_hour	loading_first_transfer_hour
transferring_to_terminal_unit_hour	unloading_terminal_unit_hour	handling_terminal_unit_hour
handling_courier_hour	delivery_hour	
delivery_collection_week	loading_initial_unit_week	transferring_to_first_transfer_week
unloading_first_transfer_week	handling-first_transfer_week	loading_first_transfer_week
transferring_to_terminal_unit_week	unloading_terminal_unit_week	handling_terminal_unit_week
handling_courier_week	delivery_week	
year	month	dom
doy	woy	week
hour	total_time	

Table 4.10: All Features

	μ_S	μ_U	σ_S	σ_U	M_S	M_U	min_S	min_U	max_S	max_U
delivery_collection	2.0903	5.8681	0.0233	0.1275	1.7638	2.4890	0.0048	0.0048	42.4451	100.0000
loading_initial_unit	0.3405	2.9644	0.0373	0.2671	0.0419	0.0591	0.0040	0.0052	91.1508	100.0000
transferring_first_transfer	2.6845	3.0471	0.1917	0.2369	2.1215	2.2946	0.0085	0.0109	100.0012	100.0000
unloading_first_transfer	0.1044	0.9861	0.0004	0.0061	0.0540	0.0821	0.0000	0.0002	7.0251	100.0000
handling_first_transfer	1.0466	1.5665	0.0067	0.0135	0.8858	1.1609	0.0000	0.0001	33.1537	100.0000
loading_first_transfer	10.5209	9.9363	0.3025	0.2938	6.8272	5.9539	0.0005	0.0108	159.5795	100.0000
transferring_terminal_unit	12.6499	12.6469	0.3950	0.4258	13.9030	13.6646	0.0542	0.0637	100.0005	100.0000
unloading_terminal_unit	0.2102	0.3441	0.0013	0.0107	0.1609	0.1721	0.0001	0.0006	15.6616	100.0000
handling_terminal_unit	0.4188	3.0853	0.0029	0.0118	0.1558	0.3634	0.0000	0.0001	22.2284	100.0000
handling_courier	1.1368	1.7698	0.0303	0.1078	0.8724	0.9140	0.0002	0.0013	70.0210	100.0000
delivery	0.6578	1.7785	0.0014	0.0093	0.5752	0.7365	0.0004	0.0004	34.7508	100.0000

Table 4.11: Descriptive Statistics For 11-Step Deliveries

according to the maximum value of unsuccessful steps being 100.

These statistics help understand the distribution and variability of the data for each step, indicating the performance and efficiency of the corresponding processes.

4.3 Proposed Approach

The initial phase entails acquiring historical transaction data to predict delivery delays. The dataset encompasses delivery dates, order dates, customer details, and other pertinent factors impacting delivery time. Subsequently, the data preparation phase is initiated. This phase involves preprocessing the data by handling missing values, outliers, and inconsistencies. Categorical variables are transformed into numerical representations using one-hot or label encoding techniques.

Upon completing the data preparation phase, the subsequent stage entails feature selection. This crucial step involves meticulously analyzing the available features to identify the most pertinent ones for accurately predicting delivery delays. Factors such as order date, customer location, and product details are deemed potential predictors deserving consideration. These carefully selected features constitute the training set with the target variable (ontime_delivery).

Once the training set has been prepared, the next stage involves training the machine learning models. The study utilized four algorithms: LR, XGBoost, CatBoost, and RF. Models were created and trained to predict delivery delays using the selected features and prepared the training dataset. Their performance is assessed using a suitable metric such as ROC-AUC. This thorough evaluation process enables the identification of the most appropriate model for each algorithm, providing valuable information about their effectiveness and predictive capabilities.

During the prediction phase, real-time data encompassing shipments still in transit is utilized. Each subsequent processing time of the deliveries collected by the cargo company is given as input to the model, and as a result, the delay of each shipment is predicted. The initial delivery delay is predicted using the time spent in this step while the shipment is still in step one. Then, when it comes to the second stage, the previous time is given to the model as a feature along with the time at this stage, and the delivery delay is re-predicted. The delay probability is updated iteratively at each step until the shipment reaches its final destination. This research seeks to determine the practicality of accurately predicting delays before shipments have reached the halfway point of their operational journey.

Finally, the prediction results are presented in the user interface or integrated with business intelligence tools for further analysis and visualization. Visualizations allow users to access and interpret the predicted delivery delay information easily. It is essential to regularly monitor and update the models as new data becomes available, ensuring that results remain accurate.

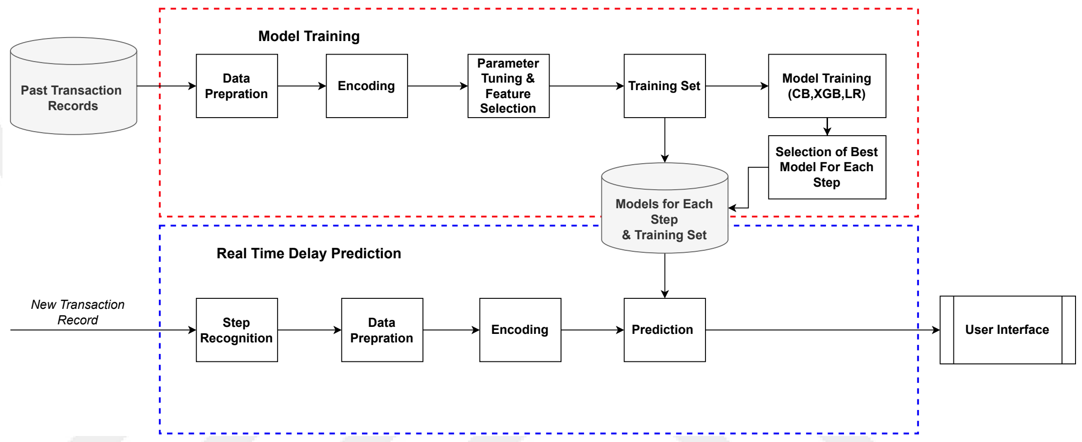

Model Training
Past Transaction Records
Data Prepration
Encoding
Parameter Tuning & Feature Selection
Training Set
Model Training (CB,XGB,LR)
Selection of Best Model For Each Step
Models for Each Step & Training Set
Real Time Delay Prediction
New Transaction Record
Step Recognition
Data Prepration
Encoding
Prediction
User Interface

Figure 4.7: Proposed Approach Schema

4.3.1 Data preparation

The unit and operation variables were first converted to a single variable called unit_operation. A pivot operation was subsequently performed between the unit_operation and duration variables to facilitate their utilization in the model. The values of 15 steps and 11 steps in the unit_operation variable have become column names. Furthermore, the pivot operation transformed the transaction time for each unit and transaction into values. The Table 4.12 below illustrates the attributes acquired for the cleaned and transformed 15-steps and 11-steps deliveries.

The features used in addition to these steps are given below.

- Unique Delivery Identifier
- Delivered Date
- Delivery Date Promised
- Terminal Unit ID
- Initial Unit ID
- Terminal Unit Town ID
- Initial Unit Town ID
- Terminal Unit District ID
- Initial Unit District ID
- On-Time (Target)

The encoding of sender and receiver cross-dock, town, and district information was carried out utilizing the LabelEncoder functionality provided by the scikit-learn preprocessing module. The LabelEncoder is a valuable tool for converting categorical variables into numerical representations, often required for effective utilization in machine learning algorithms. By applying the LabelEncoder to the sender and receiver cross-dock, town and, district information, distinct category values are assigned unique integer values. This encoding process ensures that the data is presented in a format that machine learning models can comprehend and effectively utilize. Following the encoding step, machine learning algorithms can employ the resulting numerical representations as input for training or prediction purposes.

In the data preparation phase, feature scaling was employed to normalize numerical data and enhance the performance of the machine learning algorithms utilized in this study. Specifically, the StandardScaler method from the scikit-learn preprocessing

Step	Feature Description
delivery_collection	The duration required to collect the package from the sender's location and prepare it for transportation.
loading_initial_unit	The duration required to load the package onto the initial transportation vehicle.
transferring_to_first_transfer	The time taken to transfer the package from the initial unit to the first transfer point.
unloading_first_transfer	The duration required to unload the package at the first transfer point.
handling_first_transfer	The time spent processing and organizing the package at the first transfer point.
loading_first_transfer	The duration required to load the package onto the transportation vehicle at the first transfer point.
transferring_to_second_transfer*	The duration required to transfer the package from the first transfer point to the second transfer point.
unloading_second_transfer*	The time taken to unload the package from the second transfer point.
handling_second_transfer*	The duration spent processing and organizing the package at the second transfer point.
loading_second_transfer*	The duration required to load the package onto the transportation vehicle at the second transfer point.
transferring_to_terminal_unit	The duration taken to transfer the package from the second transfer point to the destination terminal, such as a local distribution center or a regional warehouse.
unloading_terminal_unit	The time required to unload the package from the destination terminal.
handling_terminal_unit	The duration spent processing and organizing the package at the destination terminal.
handling_courier	The time taken for the courier to handle the package, including tasks like verification, signature collection, and any necessary paperwork.
delivery	The duration of the actual delivery process, starting from the departure of the courier from the destination terminal to the arrival of the package at the recipient's location.
Note: 11-Step deliveries do not include operation steps marked with *.	

Table 4.12: Descriptions of Step Features

module was applied to the duration values of each step column to ensure that the data were at the same scale. The StandardScaler method standardizes the data by subtracting the mean from values and dividing the value by the standard deviation of each feature, resulting in a transformed dataset with a mean of 0 and a standard deviation of 1. This scaling ensures that each feature contributes equally to the analysis and prevents features with larger magnitudes from dominating the analysis.

4.3.2 Feature selection

Feature selection techniques were used to identify the most relevant features for predicting delivery delay based on features in training sets. The primary aim was to identify a subset of attributes that significantly enhance the predictive performance of the models. A comprehensive investigation encompassing traditional statistical approaches and machine learning algorithms was undertaken to accomplish this. Step-wise selection based on a creation metric was applied to LR, XGBoost, CatBoost, and RF to select the best features. Using the LR model, p-values of dimensions are used as feature selection criteria. On the other hand, SHAP values are used for the decision-tree-based algorithms. A detailed explanation of SHAP values and LR p-values can be found in Section 3.2.7.

Step-wise selection is a popular technique for selecting the most efficient attribute subset. It combines forward and backward selection advantages, allowing for adding and removing variables based on criteria at different steps. The process can begin with either forward selection, where variables are incrementally added based on significance, or backward elimination, where variables are removed and potentially re-added later if they meet the significance criterion. In forward selection, the most significant variable is added first, and the process continues by re-evaluating and adding variables until no remaining variable is significant at the specified cut-off level. Conversely, backward elimination starts with a full model and iteratively removes the least significant variables until all remaining variables are significant. Step-wise selection requires distinct significance levels for adding and deleting variables to avoid infinite loops (Gramegna & Giudici, 2022).

By employing step-wise selection in LR, XGBoost, CatBoost, and RF, the most influential features were included, and irrelevant or redundant ones were eliminated. This iterative process ensured accurate and robust predictive models for delivery delay prediction.

4.3.3 Hyper-parameter optimization

Hyperparameter optimization is a crucial step in building and fine-tuning machine learning models. A study of hyperparameter optimization has been conducted for LR, XGBoost, CatBoost, and RF.

XGBoost is a popular gradient-boosting algorithm known for its high accuracy and speed. Some hyper-parameters that can be tuned for XGBoost include the learning rate, maximum depth, and the number of estimators. CatBoost is a gradient-boosting algorithm that is designed to handle categorical features without the need for one-hot encoding. Some hyper-parameters that can be tuned for CatBoost include the learning rate, depth, and regularization strength. Some of the hyper-parameters that can be tuned for random forest include the number of trees, the maximum depth of the trees, and the minimum number of samples required to split a node. RF has the advantage of handling high-dimensional data with many features, making it a popular choice for many machine learning problems. Grid search with cross-validation optimization was used to optimize the hyperparameters of these algorithms. It aimed to increase the algorithms' performance and obtain better accuracy in the data set by fine-tuning the hyperparameters.

4.4 Experiment Results

4.4.1 Logistic regression results

In this research phase, a LR classification model consisting of various delivery characteristics, including the duration of each business operation and geographic information of both sender and receiver locations, was used to estimate delivery delays accurately. A comprehensive exposition of the LR classifier methodology is provided in Section 3.2.1. The dataset consists of historical delivery records, incorporating essential attributes such as operation durations and time variables up to the current step. The **StatModels** package in Python was used to build the LR models. This package provides several tools and techniques for in-depth data analysis. All features are used in the first version of the Regression Classification model to understand how well the LR algorithm works for this particular problem. Early models have shown promising results regarding the area under the receiver operating characteristic curve ROC-AUC score, even in models still in the early stages of the delivery process. The models showed good proficiency in predicting delivery delay while shipments were still in step 4 (unloading the delivery at the first transfer unit), as demonstrated by an AUC score of 0.844 for 11-step deliveries and 0.847 for 15-step deliveries. These results showed that after step 4 (before the shipments reach half of the processing steps), the delay probability of the shipments could be predicted with high accuracy.

The predictive capability of the proposed approach improves as deliveries progress through subsequent business operations until they reach their final recipients. This observation is supported by the increasing trend in the AUC score, which further highlights the model's effectiveness in accurately predicting delivery delays. After successfully handling the delivery process at the terminal unit, the model achieved high AUC scores of 0.914 and 0.924 for 11-step and 15-step deliveries, respectively. These outcomes affirm its robust ability to predict potential delivery delays accurately.

After feature selection, LR models are trained using the selected features for each business operation of delivery with 11-step and 15-step. The feature selection process for a LR classifier is outlined in Section 4.3.2. Careful consideration of relevant variables is essential, as including irrelevant or redundant variables can reduce accuracy. For this reason, feature selection is a significant step to increase the performance of the LR model and identify the features related to the problem. The model performed well in predicting delivery delays, achieving high accuracy and AUC scores in the early stages. These performance metrics demonstrate that the feature selection process helped improve the model's predictive capability by selecting the most relevant features for predicting delivery delays.

Step	**AUC**		**Recall**		**F1-Score**		**Precision**	
	Initial	**Feature S.**	**Initial**	**Feature S.**	**Initial**	**Feature S.**	**Initial**	**Feature S.**
delivery_collection	0.542	0.561	0.215	0.224	0.326	0.314	0.727	0.830
loading_initial_unit	0.544	0.566	0.231	0.232	0.345	0.368	0.824	0.802
transferring_to_first_transfer	0.543	0.573	0.220	0.216	0.347	0.349	0.692	0.830
unloading_first_transfer	0.844	0.849	0.613	0.539	0.753	0.729	0.816	0.908
handling_first_transfer	0.849	0.851	0.601	0.638	0.715	0.761	0.793	0.843
loading_first_transfer	0.852	0.852	0.604	0.652	0.741	0.703	0.831	0.852
transferring_to_terminal_unit	0.853	0.857	0.644	0.633	0.718	0.712	0.774	0.908
unloading_terminal_unit	0.856	0.861	0.634	0.608	0.704	0.718	0.847	0.831
handling_terminal_unit	0.914	0.915	0.703	0.756	0.748	0.816	0.798	0.883
handling_courier	0.926	0.928	0.715	0.735	0.771	0.780	0.819	0.897
delivery	0.985	0.986	0.783	0.826	0.727	0.767	0.780	0.748

Table 4.13: Performance Metrics of LR Model for 11-Step

Step-wise elimination with a p-value criterion of 0.01 was employed to ascertain and incorporate the most significant features into the LR classification model. These selected features encompassed each business operation step's duration and time variables up to the current step, alongside the sender and receiver address information. The feature selection process considerably improved the model's performance, enhancing accuracy when predicting delivery delays. Four distinct models were developed, comprising two models for deliveries of 11 and 15 steps, utilizing the full range of features, and an additional two models representing 11 and 15-step deliveries, constructed through a feature selection method. AUC scores between the first and last created models showed a break at the same points. Following unloading at the first transfer unit, the model attained an AUC score of 0.849 for deliveries of 11 steps and a superior score of 0.858 for deliveries of 15 steps. Subsequently, upon completing the delivery process at the terminal unit, the AUC score increased from 0.914 to 0.915 for deliveries of 11 steps and from 0.924 to 0.927 for deliveries of 15 steps. These improved scores were obtained through a feature selection, enhancing the model's predictive capability. The AUC scores of the LR models, encompassing deliveries of both 11 and 15 steps, are showcased in Table **??**. Table 4.13 presents the scores for 11-step deliveries, illustrating a comparison between the initial model utilizing all features and the model after undergoing the feature selection step. Likewise, Table 4.14 displays the scores for deliveries with 15 steps. The purpose of these tables is to highlight the performance differences between the initial model and the model that underwent feature selection.

Step	ROC-AUC		PRECISION		F1 SCORE		RECALL	
	Initial	Feature S.	Initial	Feature S.	Initial	Feature S.	Initial	Feature S.
Delivery Collection	0.578	0.596	0.731	0.742	0.310	0.346	0.248	0.263
Loading - Initial Unit	0.585	0.602	0.740	0.751	0.331	0.340	0.252	0.266
Transferring to First Transfer	0.589	0.614	0.755	0.759	0.341	0.357	0.272	0.281
Unloading - First Transfer	0.847	0.858	0.864	0.881	0.694	0.711	0.573	0.585
Handling - First Transfer	0.859	0.861	0.875	0.880	0.711	0.723	0.621	0.644
Loading - First Transfer	0.860	0.861	0.882	0.893	0.740	0.754	0.643	0.652
Transferring to Second Transfer	0.865	0.867	0.884	0.902	0.751	0.755	0.644	0.654
Unloading - Second Transfer	0.862	0.864	0.885	0.901	0.750	0.754	0.644	0.655
Handling - Second Transfer	0.867	0.868	0.896	0.899	0.752	0.759	0.647	0.657
Loading - Second Transfer	0.867	0.868	0.896	0.904	0.761	0.765	0.658	0.660
Transferring to Terminal Unit	0.869	0.870	0.897	0.906	0.764	0.765	0.658	0.661
Unloading - Terminal Unit	0.869	0.871	0.869	0.886	0.752	0.767	0.659	0.664
Handling - Terminal Unit	0.924	0.927	0.855	0.872	0.806	0.813	0.749	0.754
Handling - Courier	0.930	0.932	0.864	0.875	0.808	0.817	0.745	0.753
Delivery	0.986	0.988	0.805	0.809	0.824	0.830	0.834	0.852

Table 4.14: Performance Metrics of LR Model for 15-Step

4.4.2 XGBoost results

XGBoost algorithm is a widely-used gradient boosting framework that effectively handles features with different characteristics. In this part, XGBoost was employed to predict delivery delays accurately. XGBoost Classifier proposed approach details are given in Section 3.2.3.

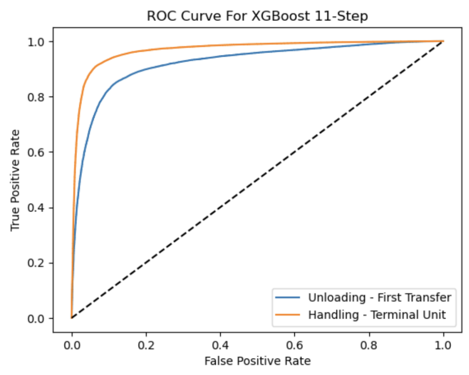
Initial models with XGBoost Classifier were used with all features and default parameters for delivery delay prediction. Additional techniques were employed to improve the models' performance, including feature selection based on SHAP values and parameter optimization using grid search with k-fold cross-validation. Hyper-parameters were tuned using the XGBoost package to optimize the performance of the XGBoost model. The hyper-parameters, encompassing learning rate, tree depth, and regularization parameters, play an influential role in governing the behavior and complexity of the XGBoost model. A practical grid search approach was employed to identify the optimal hyperparameter values. After hyper-parameter optimization, the SHAP values for each feature were calculated using the SHAP library within the XGBoost ecosystem in the feature selection process. Through SHAP values, valuable insights were obtained regarding the individual contributions of features toward the model's predictions, allowing for the inclusion of the most significant variables. The initial models for both 11-step and 15-step XGBoost deliveries were constructed without hyperparameter optimization or feature selection. AUC, Recall, Precision, and F-1 scores for each operation step are included in the Initial sub-column in Table 4.15 and Table 4.16.

For the 11-step model, the initial AUC scores in the various processing steps ranged from 72.2% to 99.9%, while the AUC scores for the 15-step model ranged from 69.6% to 98.2%. Although there were occasional small decreases in AUC scores, they generally showed an upward trend towards the delivery step in the operational steps for both models. Specifically, in the "unload_first_transfer" step, the AUC score increased to 91.3% for the 11-step model and 88.4% for the 15-step model. In addition, there was a more significant increase in AUC values in the "handling_terminal_unit" step compared to the other steps (although not like the "unload_first_transfer" step).

A hyper-parameter optimization study was conducted as a secondary step to enhance the outcomes. The GridSearchCV method was employed to identify the optimal values for the hyper-parameters, including the learning rate, max-depth, and n-estimator parameters. The corresponding tables presenting the determined optimum values are in the Appendix (see Table A.2 and A.3). The results of the optimized models are presented in the P.Tuned column of Table 4.15 and Table 4.16. The AUC scores significantly increased compared to the initial models with 11 and 15 steps. In the 11-step model, the most notable improvement was observed in the "loading_initial_unit" step, where the AUC score increased from 71% to 73%. Similarly, in the 15-step model, a substantial enhancement occurred at the "handling_second_transfer" step, with the AUC score increasing from 89.2% to 91.1%.

Upon analyzing the changes in AUC values between the steps of the optimized models, significant increases were observed at the "unloading_first_transfer" and "handling_terminal_unit" steps for 11-step and 15-step models. In the 11-step model, the AUC value escalated notably from 72.4% to 92% at the "unloading_first_transfer" process. Similarly, within the 15-step model, the AUC value experienced a substantial rise from 70.3% to 89.9% at the "unloading_first_transfer" step. Moreover, in the "handling_terminal_unit" step, the AUC value demonstrated remarkable improvements in both models. For the 11-step model, the AUC value increased from 93.2% to 96.7%, while for the 15-step model, it rose from 90.9% to 93.6%. These metrics illustrate the efficacy of the chosen hyperparameters in more accurately predicting delivery delays.

The final models, which outperformed the initial and P.Tuned models regarding AUC scores, were obtained after hyper-parameter optimization and feature selection with SHAP value criteria. The AUC scores of the final models for each step in the delivery process can be viewed in the column named "Final" in Table 4.15 and



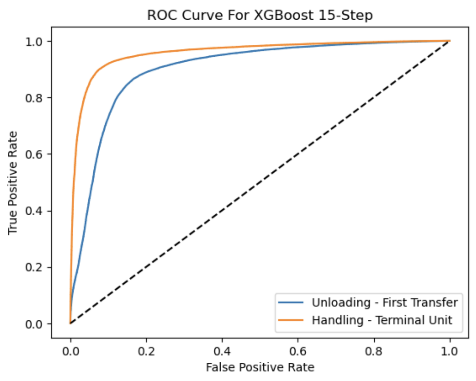


Figure 4.8: XGBoost 11-Step and 15-Step ROC Curves

Table 4.16. The increase in unloading_first_transfer and handling_terminal_units was sharply compared to previous models. The AUC value in the unloading_first_transfer increased from 73.8% to 92.3% for the 11-step model and 70.4% to 90% for the 15-step model. The AUC value in the handling_terminal_unit increased from 93.2% to 96.8% for the 11-step model and from 91.9% to 96% for the 15-step model. Figure 4.8 shows ROC curves of unloading_first_transfer and handling_terminal_unit steps for 11-step and 15-step XGBoost final models.

The final models displayed a consistent upward trend in AUC scores, indicating an ongoing enhancement in predictive accuracy as the steps progressed. No steps for the 11-step demonstrated a decrease in AUC scores compared to the P.Tuned model. These results further validated the effectiveness of the parameter tuning and feature selection process. Feature selection and hyperparameter tuning effectively increased AUC scores during critical stages of the delivery process. These improvements have helped to predict delivery delays in logistics and supply chain operations more accurately.

Detailed tables for the optimal hyperparameters and selected features for each step in the 11-step and 15-step XGBoost delivery processes can be found in the appendix. They are referred to as Table A.2, Table A.3, Table A.8, and Table A.14, respectively.

The experiments conducted demonstrate the high effectiveness of XGBoost in predicting delivery delays. The model's performance was improved by integrating fea-

Step	AUC			Recall			F1-Score			Precision		
	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final
Delivery Collection	0.722	0.726	0.727	0.233	0.233	0.235	0.362	0.363	0.371	0.853	0.869	0.879
Loading - Initial Unit	0.710	0.730	0.733	0.243	0.249	0.254	0.383	0.391	0.395	0.872	0.887	0.892
Transferring to First Transfer	0.708	0.724	0.738	0.244	0.248	0.253	0.376	0.384	0.393	0.839	0.856	0.871
Unloading - First Transfer	0.913	0.920	0.923	0.642	0.642	0.651	0.762	0.772	0.774	0.919	0.935	0.954
Handling - First Transfer	0.924	0.926	0.927	0.696	0.709	0.725	0.807	0.819	0.820	0.914	0.936	0.943
Loading - First Transfer	0.923	0.925	0.929	0.725	0.732	0.733	0.819	0.822	0.825	0.902	0.925	0.942
Transferring to Terminal Unit	0.926	0.929	0.932	0.704	0.719	0.733	0.805	0.809	0.822	0.917	0.921	0.935
Unloading - Terminal Unit	0.931	0.932	0.932	0.717	0.718	0.721	0.803	0.804	0.816	0.911	0.927	0.941
Handling - Terminal Unit	0.964	0.967	0.968	0.806	0.811	0.813	0.838	0.855	0.867	0.893	0.907	0.929
Handling - Courier	0.970	0.970	0.970	0.782	0.800	0.810	0.843	0.853	0.865	0.892	0.906	0.928
Delivery	0.999	0.999	0.999	0.881	0.888	0.906	0.854	0.856	0.872	0.827	0.829	0.841

Table 4.15: Performance Metrics of XGBoost Model for 11-Step

ture selection through SHAP values and parameter optimization via grid search and k-fold cross-validation. The feature selection process identified the most influential features, providing valuable insights into the underlying patterns. Likewise, the parameter optimization phase fine-tuned the model for optimal performance.

Step	AUC			Recall			F1-Score			Precision		
	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final
Delivery Collection	0.696	0.699	0.706	0.748	0.762	0.766	0.313	0.316	0.320	0.194	0.198	0.202
Loading - Initial Unit	0.703	0.710	0.717	0.611	0.625	0.629	0.328	0.332	0.333	0.223	0.223	0.226
Transferring to First Transfer	0.695	0.703	0.704	0.694	0.704	0.710	0.330	0.337	0.341	0.220	0.222	0.225
Unloading - First Transfer	0.884	0.899	0.900	0.859	0.861	0.869	0.729	0.734	0.738	0.627	0.633	0.641
Handling - First Transfer	0.902	0.910	0.915	0.837	0.839	0.844	0.722	0.732	0.745	0.643	0.655	0.668
Loading - First Transfer	0.887	0.892	0.915	0.816	0.836	0.856	0.716	0.731	0.747	0.647	0.659	0.663
Transferring to Second Transfer	0.905	0.908	0.913	0.855	0.877	0.880	0.743	0.760	0.771	0.678	0.681	0.686
Unloading - Second Transfer	0.899	0.906	0.915	0.799	0.807	0.823	0.730	0.737	0.747	0.669	0.677	0.684
Handling - Second Transfer	0.892	0.911	0.919	0.799	0.815	0.830	0.744	0.750	0.755	0.660	0.676	0.692
Loading - Second Transfer	0.904	0.915	0.920	0.841	0.852	0.868	0.760	0.768	0.769	0.662	0.678	0.691
Transferring to Terminal Unit	0.908	0.911	0.918	0.824	0.826	0.841	0.743	0.748	0.758	0.676	0.681	0.690
Unloading - Terminal Unit	0.901	0.909	0.919	0.814	0.835	0.847	0.741	0.757	0.760	0.666	0.682	0.689
Handling - Terminal Unit	0.924	0.936	0.960	0.835	0.856	0.858	0.777	0.793	0.806	0.757	0.758	0.759
Handling - Courier	0.944	0.952	0.965	0.836	0.843	0.853	0.776	0.789	0.805	0.744	0.744	0.763
Delivery	0.982	0.995	0.997	0.767	0.780	0.797	0.814	0.830	0.831	0.853	0.855	0.867

Table 4.16: Performance Metrics of XGBoost Model For 15-Step

4.4.3 CatBoost results

A CatBoost classification algorithm was employed for predicting delivery delays. The dataset utilized in this study comprised historical delivery records, encompassing the duration and time variables of each operation up to the current step. The duration of each business operation and information about the sender and receiver business units were incorporated into the training set. Further elaboration on the CatBoost algorithm is provided in Section 3.2.2. The initial models for 11 and 15-step deliveries were created using all features with the default parameters of the CatBoost algorithm without attribute selection. AUC, Recall, Precision, and F-1 scores for each operation step are included in the Initial sub-column in Table 4.17 and Table 4.18. From the first stages of the delivery process, with the increase in the number of steps, the AUC values of the 11-step model have increased continuously, except for the seventh point (AUC value decreased from 92.2% to 92.1% when loading_first_transfer step to the transferring_terminal_unit step). In the 15-step model, however, there was a decrease in two points (transferring_to_first_transfer, transferring_to_second_transfer), and the AUC value of the other steps increased. When the differences between the AUC values were examined, it was determined that there were jumps at the same points in both models. At the unload_first_transfer step, the AUC increased from 71.3% to 90.2% for the 11-step and from 71.9% to 79.2% for the 15-step.

In the first phase of the model, default parameters were used. A hyper-parameter optimization study was carried out as a second phase to improve the results. The scikit-learn library's GridSearchCV and StratifiedKFold methods were used to determine the best values for the learning rate, max-depth, and n-estimator parameters. Detailed information about hyper-parameter optimization methods is given in Section 4.3.3. Various combinations of hyperparameters were evaluated to determine optimal values that maximize the model's performance in estimating delivery delays. The CatBoost model was fine-tuned during this process to determine the optimal values for the learning rate, maximum depth, and n-estimators. Table A.1 and Table A.4 provide the determined optimal values. The AUC results of the optimized models are included in the P.Tuned column in Table 4.17 and Table 4.18. It was observed that the AUC scores with 11 and 15- steps models increased when compared to the first models. As in the first model, the AUC scores increased significantly at specific points. In the unloading_first_transfer process for the 11-step model, the AUC value increased from 72.6% to 91.6% and from 72.5% to 79.6% for the 15-step model.

The second improvement was in the handling_terminal_unit. AUC values increased from 93.1% to 96.8% in the 11-step model and from 83.3% to 93.7% in the 15-step model. These metrics demonstrate the effectiveness of the selected hyperparameters in accurately predicting delivery delays.

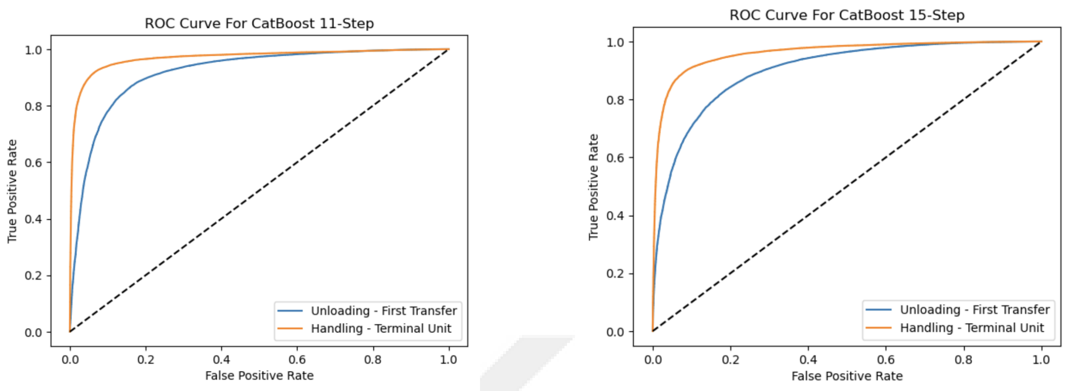


Figure 4.9: CatBoost 11-Step and 15-Step ROC Curves

During the final phase, feature selection was performed using a step-wise approach and the SHAP value criterion, as well as hyper-parameters specific to each step. The appendix (see Table A.7 and Table A.12) provides details on the selected attributes. It has been observed that the AUC values of models using both feature selection and hyper-parameter optimization are higher than those created using only hyper-parameter optimization. The performance of models with 11 and 15 steps was improved using both techniques. As with the initial and parameter-optimized models, the AUC values showed a sharp increase in the unloading_first_transfer and handling_terminal_unit features. The AUC value at unloading_first_transfer increased from 75.2% to 91.7% for the 11-step model and from 73.9% to 90.4% for the 15-step model. The AUC value at handling_terminal_unit increased from 93.4% to 97% for the 11-step model and 92.3% to 96% for the 15-step model. Figure 4.9 shows ROC curves of unloading_first_transfer and handling_terminal_unit steps for 11-step and 15-step CatBoost final models. These results indicate that the feature selection process based on SHAP values successfully improved the accuracy of models.

The performance of the delivery delay prediction model was enhanced by incorporating feature selection using SHAP values and parameter optimization with grid

Step	AUC			Recall			F1-Score			Precision		
	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final
Delivery Collection	0.701	0.730	0.734	0.235	0.234	0.238	0.371	0.369	0.375	0.875	0.873	0.879
Loading - Initial Unit	0.704	0.732	0.734	0.253	0.252	0.254	0.395	0.389	0.395	0.892	0.861	0.895
Transferring to First Transfer	0.713	0.726	0.752	0.253	0.254	0.253	0.399	0.394	0.393	0.871	0.868	0.903
Unloading - First Transfer	0.902	0.916	0.917	0.651	0.677	0.682	0.794	0.774	0.791	0.954	0.961	0.942
Handling - First Transfer	0.921	0.923	0.927	0.725	0.729	0.713	0.823	0.820	0.810	0.945	0.943	0.936
Loading - First Transfer	0.922	0.924	0.928	0.729	0.743	0.733	0.818	0.825	0.830	0.934	0.942	0.940
Transferring to Terminal Unit	0.921	0.926	0.929	0.722	0.733	0.731	0.815	0.822	0.813	0.936	0.935	0.916
Unloading - Terminal Unit	0.927	0.931	0.934	0.728	0.721	0.731	0.818	0.822	0.817	0.934	0.939	0.941
Handling - Terminal Unit	0.965	0.968	0.970	0.814	0.813	0.822	0.863	0.874	0.867	0.919	0.929	0.933
Handling - Courier	0.972	0.975	0.975	0.813	0.828	0.810	0.860	0.865	0.874	0.913	0.928	0.925
Delivery	0.997	0.999	0.999	0.884	0.906	0.922	0.860	0.889	0.872	0.911	0.913	0.915

Table 4.17: Performance Metrics of CatBoost Model for 11-Step

search and k-fold cross-validation. The feature selection process identifies the most influential features, providing valuable insights into late deliveries' underlying patterns. The parameter optimization phase ensured that the model was fine-tuned for increased performance.

Step	AUC			Recall			F1-Score			Precision		
	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final
delivery_collection	0.713	0.719	0.724	0.795	0.811	0.816	0.318	0.318	0.320	0.197	0.198	0.199
loading_initial_unit	0.721	0.726	0.728	0.813	0.834	0.835	0.336	0.338	0.347	0.216	0.219	0.219
transferring_to_first_transfer	0.719	0.725	0.739	0.812	0.832	0.841	0.338	0.344	0.346	0.213	0.218	0.218
unloading_first_transfer	0.792	0.796	0.904	0.885	0.902	0.924	0.694	0.711	0.719	0.569	0.574	0.589
handling_first_transfer	0.813	0.817	0.914	0.875	0.880	0.896	0.717	0.732	0.743	0.629	0.633	0.635
loading_first_transfer	0.816	0.817	0.915	0.873	0.877	0.894	0.735	0.740	0.754	0.628	0.643	0.652
transferring_to_second_transfer	0.806	0.814	0.917	0.879	0.901	0.903	0.747	0.749	0.754	0.639	0.645	0.647
unloading_second_transfer	0.813	0.819	0.916	0.866	0.888	0.904	0.738	0.745	0.762	0.647	0.650	0.659
handling_second_transfer	0.816	0.824	0.920	0.877	0.894	0.901	0.738	0.748	0.761	0.648	0.653	0.659
loading_second_transfer	0.821	0.826	0.921	0.895	0.903	0.904	0.756	0.758	0.764	0.649	0.661	0.661
transferring_to_terminal_unit	0.822	0.829	0.921	0.874	0.897	0.897	0.723	0.738	0.755	0.635	0.650	0.652
unloading_terminal_unit	0.825	0.833	0.923	0.868	0.886	0.894	0.751	0.765	0.767	0.649	0.659	0.672
handling_terminal_unit	0.927	0.937	0.960	0.855	0.870	0.891	0.804	0.810	0.821	0.748	0.752	0.760
handling_courier	0.944	0.945	0.968	0.866	0.874	0.885	0.807	0.815	0.817	0.744	0.754	0.759
delivery	0.983	0.991	0.997	0.801	0.805	0.809	0.819	0.824	0.830	0.834	0.834	0.853

Table 4.18: Performance Metrics of CatBoost Model for 15-Step

4.4.4 Random forest results

The initial models for both the 11-step and 15-step scenarios were constructed by employing all the features with the default parameters of the RF (Random Forest) algorithm. The column named "Initial" in Table 4.19 and Table 4.20 contains the AUC, Recall, Precision, and F-1 scores for each processing step. A progressive increase was ascertained upon examining the AUC values in the 11 and 15-step models. When the AUC values in each operation step were examined, it was determined that both models experienced jumps at the same points. In the unload_first_transfer step, the AUC increased from 72.2% to 90.1% in the 11-step model and from 66.2% to 80.2% in the 15-step model. In the handling_terminal_unit step, the AUC value in the 11-step model increased from 92.8% to 96.1%, while the 15-step model showed a significant increase from 85.6% to 86.9%.

A thorough hyper-parameter optimization study was carried out as a subsequent phase to augment the results. The GridSearchCV method was employed to ascertain the optimal parameters values: learning rate, max-depth, and n-estimator. The relevant tables containing the determined optimum values can be found in the Appendix (refer to Table A.6 and Table A.5). The outcomes of the optimized models are presented in the P.Tuned column of Table 4.19 and Table 4.20. Upon scrutinizing the changes in AUC values between the steps of the optimized models, noteworthy increases were observed, particularly at the "unloading_first_transfer" stage for both the 11-step and 15-step models. The AUC values demonstrated a remarkable escalation of 72.4% to 91.3% for 11 and 67.3% to 81.1% for 15 step model. The final models with higher AUC scores than the initial and optimized models were obtained through hyperparameter optimization and feature selection. As with the previous models, the AUC values suddenly increased at unloading_first_transfer step. The final models, which had higher AUC scores than the initial and optimization models, were obtained through hyper-parameter optimization and feature selection. The AUC value for unloading_first_transfer increased from 72.5% to 91.7% for the 11-step model and from 69.4% to 81.8% for the 15-step model. These results indicate that the feature selection process, combined with parameter optimization, successfully improved the accuracy of the models.

The optimized final models outperformed the initial and parameter-tuned models, showcasing the significance of feature selection and hyperparameter optimization techniques for accurate delivery delay predictions in logistics and supply chain operations.

Step	AUC			Recall			F1-Score			Precision		
	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final
delivery_collection	0.719	0.722	0.724	0.223	0.229	0.223	0.331	0.337	0.339	0.770	0.844	0.874
loading_initial_unit	0.720	0.721	0.721	0.242	0.235	0.236	0.358	0.381	0.384	0.830	0.873	0.846
transferring_to_first_transfer	0.722	0.724	0.725	0.243	0.239	0.230	0.357	0.356	0.355	0.768	0.835	0.793
unloading_first_transfer	0.91	0.913	0.917	0.640	0.591	0.600	0.757	0.750	0.735	0.851	0.934	0.887
handling_first_transfer	0.921	0.923	0.925	0.628	0.681	0.707	0.750	0.772	0.792	0.874	0.903	0.891
loading_first_transfer	0.92	0.921	0.923	0.668	0.674	0.697	0.814	0.762	0.821	0.899	0.894	0.883
transferring_to_terminal_unit	0.923	0.925	0.927	0.658	0.667	0.694	0.735	0.774	0.805	0.833	0.919	0.855
unloading_terminal_unit	0.928	0.928	0.929	0.669	0.654	0.676	0.777	0.730	0.771	0.850	0.861	0.895
handling_terminal_unit	0.961	0.962	0.963	0.760	0.763	0.745	0.799	0.852	0.800	0.826	0.905	0.926
handling_courier	0.966	0.967	0.969	0.730	0.797	0.765	0.795	0.842	0.836	0.872	0.904	0.896
delivery	0.997	0.997	0.997	0.847	0.877	0.819	0.804	0.796	0.869	0.818	0.822	0.838

Table 4.19: Performance Metrics of RF Model for 11-Step

Step	AUC			Recall			F1-Score			Precision		
	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final	Initial	P. Tuned	Final
delivery_collection	0.631	0.635	0.640	0.708	0.762	0.715	0.293	0.292	0.315	0.184	0.180	0.197
loading_initial	0.674	0.691	0.698	0.575	0.618	0.628	0.326	0.314	0.320	0.218	0.204	0.205
transferring_first_transfer	0.662	0.673	0.694	0.666	0.642	0.678	0.327	0.319	0.312	0.211	0.217	0.217
unloading_first_transfer	0.802	0.811	0.818	0.787	0.793	0.783	0.715	0.719	0.702	0.611	0.583	0.610
handling_first_transfer	0.818	0.821	0.848	0.836	0.799	0.807	0.664	0.703	0.691	0.600	0.611	0.621
loading_first_transfer	0.821	0.827	0.829	0.755	0.821	0.810	0.652	0.665	0.704	0.640	0.656	0.658
transferring_second_transfer	0.820	0.887	0.912	0.851	0.867	0.844	0.676	0.760	0.734	0.644	0.640	0.667
unloading_second_transfer	0.821	0.840	0.844	0.737	0.740	0.774	0.681	0.670	0.739	0.612	0.670	0.659
handling_second_transfer	0.824	0.824	0.831	0.778	0.791	0.795	0.673	0.701	0.727	0.655	0.674	0.689
loading_second_transfer	0.838	0.860	0.863	0.838	0.791	0.808	0.721	0.766	0.745	0.656	0.641	0.628
transferring_to_terminal_unit	0.840	0.847	0.881	0.771	0.757	0.828	0.680	0.682	0.728	0.624	0.623	0.689
unloading_terminal_unit	0.856	0.856	0.895	0.763	0.817	0.792	0.694	0.702	0.686	0.651	0.664	0.627
handling_terminal_unit	0.869	0.872	0.887	0.797	0.819	0.806	0.711	0.777	0.793	0.756	0.748	0.685
handling_courier	0.874	0.879	0.930	0.808	0.771	0.836	0.714	0.750	0.791	0.742	0.725	0.756
delivery	0.960	0.973	0.997	0.735	0.725	0.718	0.757	0.822	0.821	0.842	0.834	0.786

Table 4.20: Performance Metrics of RF Model for 15-Step

4.4.5 Comparison of methods

Four classification algorithms were used to predict delivery delay: Logistic Regression (LR), a conventional machine learning approach; Random Forest (RF) from bagging algorithms; and XGBoost and CatBoost from boosting algorithms. These well-established algorithms are widely employed in the logistics industry for addressing binary classification tasks. Diverse models were established by integrating feature selection and parameter optimization techniques, utilizing real data from a logistics company. Comprehensive performance comparisons were conducted for each operational step between logistic models involving 11 and 15-step shipments, which were constructed through attribute selection, and XGBoost, CatBoost, and

RF models incorporating 11 and 15-step deliveries, where parameter optimization was combined with the attribute selection method. The Area Under the Receiver Operating Characteristic Curve (AUC) was utilized as a comparative metric. Table 4.21 and Table 4.22 present the corresponding AUC scores for each algorithm at different operational steps in 11-step deliveries and 15-step deliveries, respectively.

When comparing the AUC scores of the 11-step models for each processing step, it was observed that the LR algorithm had the lowest score. The range of LR AUC scores was from 56.1% to 98.6%. Although the LR AUC score increased consistently at each step towards the delivery step, it remained lower than the AUC scores of the other three algorithms. Of the 11-step models, RF is another algorithm with a low AUC score, ranging from 72.1% to 99.7%. While it outperformed LR, it was less effective than Boosting algorithms. When evaluating the AUC scores of models created with XGBoost and CatBoost, it was found that the AUC scores for XGBoost ranged from 72.7% to 99.9%, whereas the AUC scores for CatBoost ranged from 73.4% to 99.9%. Although the results of the two algorithms were very similar in the 11-step models, CatBoost had higher AUC scores than XGBoost in the first three and last four steps.

While comparing the 15-step models at each operational step, the Logistic Regression (LR) and Random Forest (RF) algorithms demonstrated lower AUC scores compared to the Boosting algorithms. Further examination of the AUC scores for LR and RF revealed that LR exhibited higher performance in specific stages, while RF was successful in others. Specifically, LR achieved AUC scores ranging from 59.6% to 98.8%, while RF obtained scores between 64% and 99.7%. It was observed that LR slightly outperformed RF in terms of accuracy across multiple operational steps. After examining the AUC scores of models created using XGBoost and CatBoost, a different result was obtained from the analysis results of the 11-step models. Consequently, at each step, the AUC score of CatBoost was either equivalent to or greater than the AUC score of XGBoost. Regarding the shipment delay prediction, CatBoost exhibited slightly better performance in the 11-step models, with only

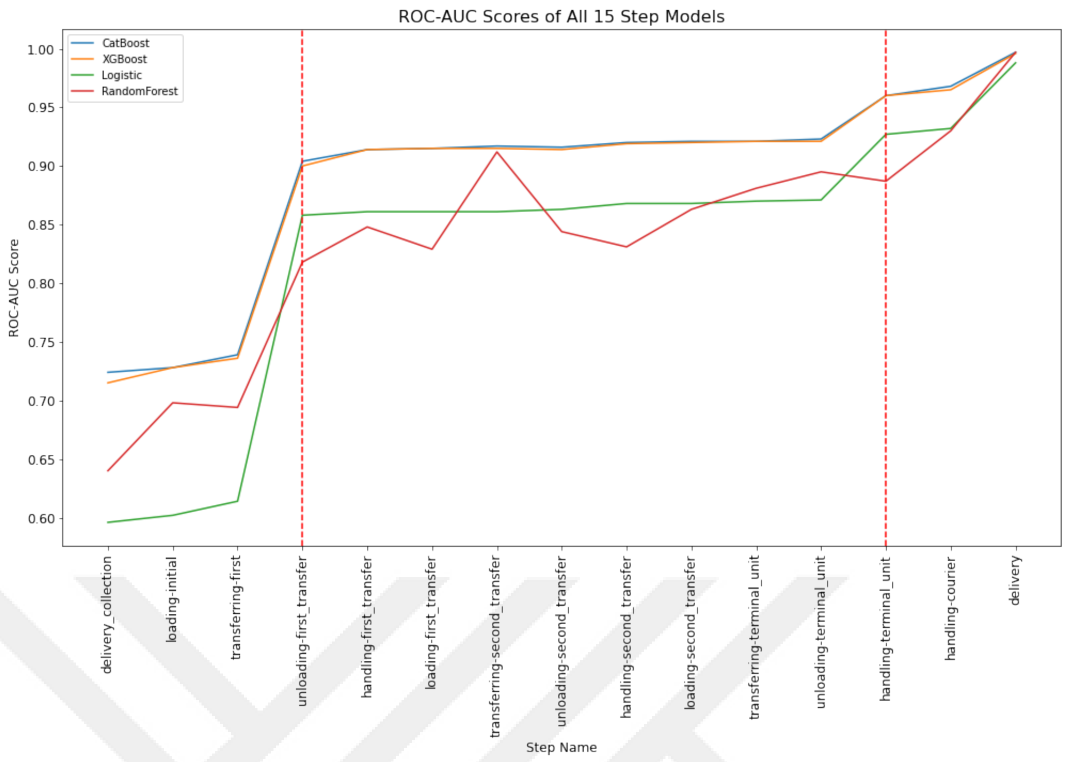


Figure 4.10: AUC Scores of All 15-Step Models

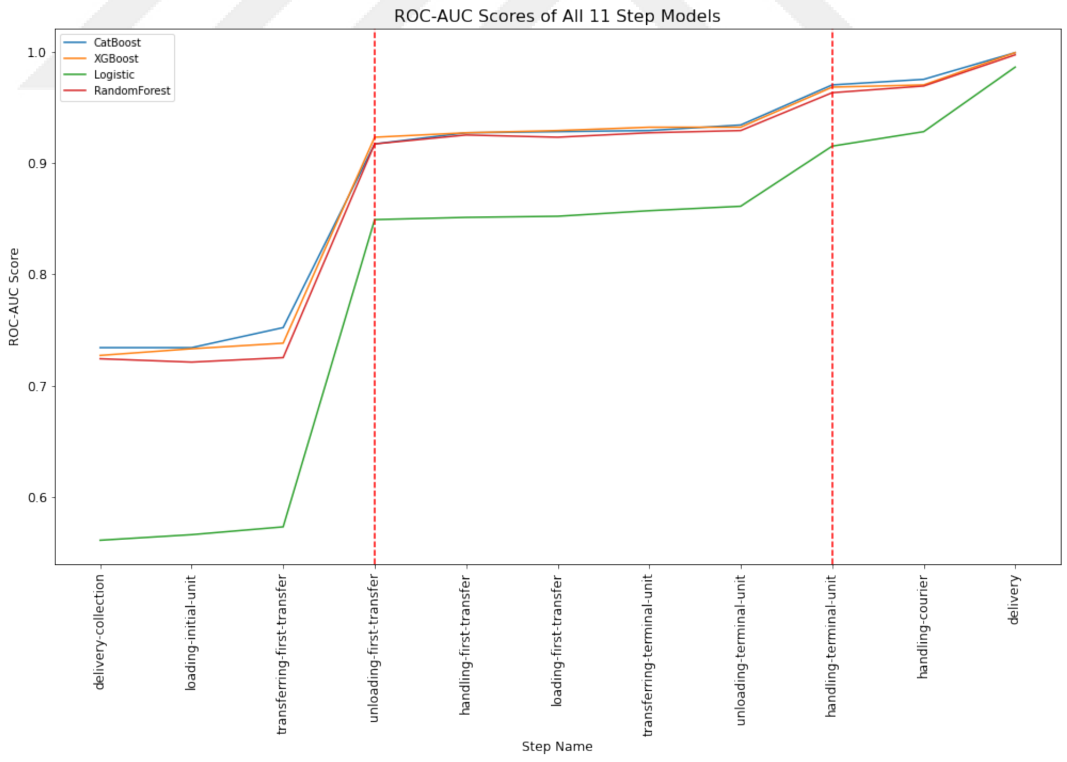


Figure 4.11: AUC Scores of All 11-Step Models

Step	**LR**	**XGBoost**	**CatBoost**	**RF**
delivery_collection	0.561	0.727	0.734	0.724
loading_initial_unit	0.566	0.733	0.734	0.721
transferring_to_first_transfer	0.573	0.738	0.752	0.725
unloading_first_transfer	0.849	0.923	0.917	0.917
handling_first_transfer	0.851	0.927	0.927	0.925
loading_first_transfer	0.852	0.929	0.928	0.923
transferring_to_terminal_unit	0.857	0.932	0.929	0.927
unloading_terminal_unit	0.861	0.932	0.934	0.929
handling_terminal_unit	0.915	0.968	0.970	0.963
handling_courier	0.928	0.970	0.975	0.969
delivery	0.986	0.999	0.999	0.997

Table 4.21: Table of AUC Scores for All 11-Step Models

marginal differences observed between the AUC scores of XGBoost and CatBoost. However, in the 15-step models, the CatBoost algorithm consistently outperformed XGBoost. Two notable increments in AUC scores were observed during the analysis. The first significant increase was observed as the transitioning point from the transferring_to_first_transfer stage to the unloading_first_transfer stage. The second prominent increase occurred from the unloading_terminal_unit step to the handling_terminal_unit step. Regarding these two breakpoints, it is evident that the utilized features and hyperparameters play a more substantial role in the prediction process.

Step	**LR**	**XGBoost**	**CatBoost**	**RF**
delivery_collection	0.596	0.715	0.724	0.640
loading_initial_unit	0.602	0.728	0.728	0.698
transferring_to_first_transfer	0.614	0.736	0.739	0.694
unloading_first_transfer	0.858	0.900	0.904	0.818
handling_first_transfer	0.861	0.914	0.914	0.848
loading_first_transfer	0.861	0.915	0.915	0.829
transferring_to_second_transfer	0.861	0.915	0.917	0.912
unloading_second_transfer	0.863	0.914	0.916	0.844
handling_second_transfer	0.868	0.919	0.920	0.831
loading_second_transfer	0.868	0.920	0.921	0.863
transferring_to_terminal_unit	0.870	0.921	0.921	0.881
unloading_terminal_unit	0.871	0.921	0.923	0.895
handling_terminal_unit	0.927	0.960	0.960	0.887
handling_courier	0.932	0.965	0.968	0.930
delivery	0.988	0.996	0.997	0.997

Table 4.22: Table of AUC Scores for All 15-Step Models

5. CONCLUSION AND FUTURE WORK

This study addresses the challenge of predicting delivery delays in the logistics industry through binary classification. The deliveries are either delayed (unsuccessful) or on time (successful). The performance of different algorithms, namely LR, XGBoost, CatBoost, and RF, is examined to determine the best algorithm for predicting delivery delays in this scenario. Real-world data are utilized for the experiment, and the accuracy of each algorithm is analyzed at different milestones in the delivery process. Remarkably, CatBoost models outperformed the other algorithms, particularly at the initial processing stage and subsequent milestones.

Feature selection and hyper-parameter optimization techniques were employed to optimize the models' performance. Relevant features associated with delivery delays, such as receiver and sender cross-dock, time features, and shipment type, were selected using grid search and k-fold validation. These selected features were utilized as input variables for the models. Additionally, the hyperparameters of each algorithm were fine-tuned to maximize their performance on the validation data. This process entailed adjusting the learning rate, number of trees, and regularization parameters for the best possible results. The combination of feature selection and parameter tuning played a pivotal role in enhancing the accuracy of the models and selecting the most suitable algorithm for predicting delivery delays within the logistics industry.

To further elaborate on the results, it is essential to understand the significance of each algorithm's performance. Moreover, the study's findings suggest that the algorithms' performance varied across different milestones of the delivery process, emphasizing the need for careful selection of the feature and hyperparameter set based on the specific context of the delivery scenario. Overall, the study provided valuable insights into the effectiveness of different algorithms for predicting delivery delays. These insights could benefit the logistics industry in optimizing its delivery processes and improving customer satisfaction.

In future work, additional features will be integrated using advanced feature engineering techniques to improve the performance of the models further. A comprehensive range of algorithms will also be explored, including deep learning models such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to achieve even greater accuracy and robustness. The proposed models can uncover more advanced insights and improved performance by utilizing larger datasets, even at a big-data scale. However, it should be noted that these approaches require high computational hardware and power. The challenges associated with model interpretability and computational requirements will be addressed in the next steps of the research. The accuracy, reliability, and scalability of predictive models can be improved by addressing these limitations, enabling better decision-making in logistics operations.

BIBLIOGRAPHY

Araujo, A. C., & Etemad, A. (2021). End-to-end prediction of parcel delivery time with deep learning for smart-city applications. *Electrical Engineering and Systems Science, Signal Processing*. doi: 10.48550/arXiv.2009.12197

Açılar, A. (2016). E-commerce in turkey. *PressAcademia Procedia*, *2*(1), 281 - 288. doi: 10.17261/Pressacademia.2016118648

Balakrishna, P., Ganesan, R., Sherry, L., & Levy, B. S. (2008). Estimating taxi-out times with a reinforcement learning algorithm. In *Proceedings of the 27th digital avionics systems conference, ieee/aiaa* (p. 3.D.3-1-3.D.3-12). St. Paul, Minnesota: IEEE. doi: 10.1109/DASC.2008.4702812

Berkson, J. (1944). Application of the logistic function to bio-assay. *Journal of the American Statistical Association*, *39*(227), 357-365. doi: 10.1080/01621459.1944.10500699

Bishop, C. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Berlin, Heidelberg: Springer-Verlag.

Chawla, N., Lazarevic, A., Hall, L., & Bowyer, K. (2003, January). Smoteboost: Improving prediction of the minority class in boosting. In *Proceedings of the 7th european conference on principles and practice of knowledge discovery in database* (Vol. 2838, p. 107-119). Cavtat-Dubrovnik, Croatia: Springer. doi: 10.1007/978-3-540-39804-2_12

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (p. 785-794). New York, NY, USA: ACM. doi: 10.1145/2939672.2939785

Esmaeilzadeh, E., & Mokhtarimousavi, S. (2020). Machine learning approach for flight departure delay prediction and analysis. *Transportation Research Record*, *2674*(8), 145-159. doi: 10.1177/0361198120930014

Gaddis, G., & Gaddis, M. (1990, September). Introduction to biostatistics: Part 5, statistical inference techniques for hypothesis testing with nonparametric data. *Annals of Emergency Medicine*, *19*(9), 1054—1059. doi: 10.1016/s0196

-0644(05)82571-5

Gameng, H. A., Gerardo, B. B., & Medina, R. P. (2019). Modified adaptive synthetic smote to improve classification performance in imbalanced datasets. In *Proceedings of the 2019 ieee 6th international conference on engineering technologies and applied sciences (icetas)* (p. 1-5). Kuala Lumpur, Malaysia: IEEE. doi: 10.1109/ICETAS48360.2019.9117287

Gao, C., Zhang, F., Wu, G., Hu, Q., Ru, Q., Hao, J., ... Sun, Z. (2021). A deep learning method for route and time prediction in food delivery service. In *Proceedings of the 27th acm sigkdd conference on knowledge discovery amp; data mining* (p. 2879–2889). NY, USA: Association for Computing Machinery. doi: 10.1145/3447548.3467068

Gentek, A. (2022). *Employee churn prediction in healthcare industry using supervised machine learning* (Doctoral dissertation, KTH Royal Institute of Technology). Retrieved from https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-321536

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. doi: 10.1016/j.sbspro.2014.01.1483

Gramegna, A., & Giudici, P. (2022). Shapley feature selection. *FinTech*, *1*(1), 72-80. doi: 10.3390/fintech1010006

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, 220-239. doi: 10.1016/j.eswa.2016.12.035

Hancock, J. T., & Khoshgoftaar, T. M. (2020, November). Catboost for big data: An interdisciplinary review. *Journal of Big Data , SpringerOpen*. doi: 10.1186/s40537-020-00369-8

Hoi, C. S., Leung, C. K., & Souza, J. (2020). Prediction of food preparation time for smart city. In *Proceedings of the ieee 22nd international conference on high performance computing and communications* (p. 1203-1210). Yanuca Island, Cuvu, Fiji: IEEE. doi: 10.1109/HPCC-SmartCity-DSS50907.2020.00156

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Introduction to the logistic regression model. In *Applied logistic regression* (p. 7-12). Hoboken, New Jersey: John Wiley Sons, Ltd. doi: 10.1002/9781118548387.ch1

Huppenkothen, D., Heil, L., Hogg, D., & Mueller, A. (2017, April). Using machine learning to explore the long-term evolution of grs 1915+105. *Monthly Notices of the Royal Astronomical Society*, *466*, 2364-2377. doi: 10.1093/mnras/stw3190

Jonquais, A. C. J., & Krempl, F. (2019). *Predicting shipping time with machine learning* (Master's thesis, Massachusetts Institute of Technology). Retrieved from https://dspace.mit.edu/bitstream/handle/1721.1/121280/Jonquais_Krempl_2019.pdf?sequence=1&isAl

Kawatani, T., Yamaguchi, T., Sato, Y., Maita, R., & Mine, T. (2021, June). Prediction of bus travel time over intervals between pairs of adjacent bus stops using city bus probe data. *International Journal of Intelligent Transportation Systems Research*, *19*(2), 456–467. doi: 10.1007/s13177-021-00251-8

Khiari, J., & Olaverri-Monreal, C. (2020). Boosting algorithms for delivery time prediction in transportation logistics. In *Proceedings of the 2020 international conference on data mining workshops (icdmw)* (p. 251-258). Sorrento, Italy: IEEE. doi: 10.1109/ICDMW51313.2020.00043

Kim, T. (2015, November). T test as a parametric statistic. *Korean Journal of Anesthesiology*, *68*, 540. doi: 10.4097/kjae.2015.68.6.540

King, A., & Eckersley, R. (2019, January). Inferential statistics iv: Choosing a hypothesis test. In *Statistics for biomedical engineers and scientists* (p. 147-171). London, United Kingdom: Academic Press. doi: 10.1016/B978-0-08-102939-8.00016-5

Li, N., & Jimenez, R. (2018, January). A logistic regression classifier for long-term probabilistic prediction of rock burst hazard. *Natural Hazards: Journal of the International Society for the Prevention and Mitigation of Natural Hazards*, *90*(1), 197-215. doi: 10.1007/s11069-017-3044-7

Liang, J. (2022, December). Confusion matrix: Machine learning. *POGIL Activity Clearinghouse*, *3*(4). Retrieved from https://pac.pogil.org/index.php/pac/

article/view/304

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems* (p. 4768–4777). Red Hook, NY, USA: Curran Associates Inc. doi: 10.48550/arXiv.1705.07874

Metzger, A., Leitner, P., Ivanović, D., Schmieders, E., Franklin, R., Carro, M., ... Pohl, K. (2015). Comparing and combining predictive business process monitoring techniques. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *45*(2), 276-290. doi: 10.1109/TSMC.2014.2347265

Nigam, R., & Govinda, K. (2017). Cloud based flight delay prediction using logistic regression. In *Proceedings of the 2017 international conference on intelligent sustainable systems (iciss)* (p. 662-667). Palladam, India: IEEE. doi: 10.1109/ISS1.2017.8389254

Oktay, E., Üstün Özen, & Burmaoğlu, S. (2009). Birleşmiş milletler kalkınma programı beşeri kalkınma endeksi verilerini kullanarak diskriminant analizi ve lojistik regresyon sınıflandırma performanslarının karşılaştırılması. *Savunma Bilimleri Dergisi*, *8*(2). Retrieved from https://dergipark.org.tr/tr/pub/khosbd/issue/19229/204331

Ozmen, E., Öner, A., Khosrowshahi, F., & Underwood, J. (2013, August). Sme buying behaviour: Literature review and an application agenda. *The Marketing Review*, *13*, 207-227. doi: 10.1362/146934713X13699019904768

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. In *Proceedings of the 32nd international conference on neural information processing systems* (p. 6639–6649). Red Hook, NY, USA: Curran Associates Inc. doi: 10.48550/arXiv.1706.09516

Punmiya, R., & Choe, S. (2019). Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. *IEEE Transactions on Smart Grid*, *10*(2), 2326-2329. doi: 10.1109/TSG.2019.2892595

Ranathunga, M. I. D., Wijayanayake, A. N., & Niwunhella, D. H. H. (2021). Solution approaches for combining first-mile pickup and last-mile delivery in an e-commerce logistic network: A systematic literature review. In *Proceedings of 2021 international research conference on smart computing and systems engineering (scse)* (Vol. 4, p. 267-275). Colombo, Sri Lanka. doi: 10.1109/SCSE53661.2021.9568349

Rebollo, J. J., & Balakrishnan, H. (2014). Characterization and prediction of air traffic delays. *Transportation Research Part C: Emerging Technologies*, *44*, 231-241. doi: 10.1016/j.trc.2014.04.007

Salari, N., Liu, S., & Shen, Z.-J. M. (2022). Real-time delivery time forecasting and promising in online retailing: When will your package arrive? *Manufacturing & Service Operations Management*, *24*(3), 1421-1436. doi: 10.1287/msom.2022.1081

Schultz, B. B. (1985). Levene's test for relative variation. *Systematic Zoology*, *34*(4), 449–456. doi: 10.2307/2413207

Stoltzfus, J. C. (2011). Logistic regression: A brief primer. *Academic Emergency Medicine*, *18*(10), 1099-1104. doi: 10.1111/j.1553-2712.2011.01185.x

Subasi, A., & Erçelebi, E. (2005). Classification of eeg signals using neural network and logistic regression. *Computer Methods and Programs in Biomedicine*, *78*(2), 87-99. doi: 10.1016/j.cmpb.2004.10.009

Taparia, A., & Brady, M. (2021). Bus journey and arrival time prediction based on archived avl/gps data using machine learning. In *Proceedings of the 7th international conference on models and technologies for intelligent transportation systems* (p. 1-6). Heraklion, Greece: IEEE. doi: 10.1109/MT-ITS49943.2021.9529328

Vernimmen, B., Dullaert, W., & Engelen, S. (2007). Schedule unreliability in liner shipping: Origins and consequences for the hinterland supply chain. *Maritime Economics and Logistics*, *9*, 193-213. doi: 10.1057/palgrave.mel.9100182

Wang, C., Deng, C., & Wang, S. (2020). Imbalance-xgboost: leveraging weighted and focal losses for binary label-imbalanced classification with xgboost. *Pattern Recognition Letters*, *136*, 190-197. doi: 10.1016/j.patrec.2020.05.035

Wojciechowski, S., & Wilk, S. (2017). Difficulty factors and preprocessing in imbalanced data sets: An experimental study on artificial data. *Founda-*

tions of Computing and Decision Sciences, *42*(2), 149–176. doi: 10.1515/fcds-2017-0007

Yu, Y., Zhou, Q., Yi, S., Zheng, H., Wang, S., Hao, J., . . . Sun, Z. (2021). Delay to group in food delivery system: A prediction approach. In *Proceedings of the 17th intelligent computing theories and application* (pp. 540–551). Cham, Germany: Springer International. doi: 10.1007/978-3-030-84529-2_46

APPENDIX A: APPENDIX

Step	**Hyperparameters**
delivery_collection	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 200}
loading_initial_unit	{'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 200}
transferring_to_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
unloading_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}
handling_first_transfer	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 200}
loading_first_transfer	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}
transferring_to_terminal_unit	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}
unloading_terminal_unit	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}
handling_terminal_unit	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}
handling_courier	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 300}
delivery	{'learning_rate': 0.3, 'max_depth': 7, 'n_estimators': 100}

Table A.1: Hyperparameters for CatBoost 11-Step Models

Step	**Hyperparameters**
delivery_collection	{'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 100}
loading_initial_unit	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
transferring_to_first_transfer	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 100}
unloading_first_transfer	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 100}
handling_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
loading_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
transferring_to_terminal_unit	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 100}
unloading_terminal_unit	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 100}
handling_terminal_unit	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
handling_courier	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 100}
delivery	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 300}

Table A.2: Hyperparameters for XGBoost 11 Steps Models

Step	**Hyperparameters**
delivery_collection	{'learning_rate': 0.3, 'max_depth': 6, 'n_estimators': 100}
loading_initial_unit	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
transferring_to_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
unloading_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
handling_first_transfer	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 100}
loading_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
transferring_to_second_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
unloading_second_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
handling_second_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
loading_second_transfer	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 100}
transferring_to_terminal_unit	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
unloading_terminal_unit	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
handling_terminal_unit	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 100}
handling_courier	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 100}
delivery	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 300}

Table A.3: Hyperparameters for XGBoost 15 Steps Models

Step	Hyperparameters
delivery_collection	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}
loading_initial_unit	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}
transferring_to_first_transfer	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 200}
unloading_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}
handling_first_transfer	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 200}
loading_first_transfer	{'learning_rate': 0.3, 'max_depth': 6, 'n_estimators': 200}
transferring_to_second_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
unloading_second_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
handling_second_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
loading_second_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
transferring_to_terminal_unit	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
unloading_terminal_unit	{'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 200}
handling_terminal_unit	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 300}
handling_courier	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 300}
delivery	{'learning_rate': 0.3, 'max_depth': 6, 'n_estimators': 200}

Table A.4: Hyperparameters for CatBoost 11 Steps Models

Step	**Hyperparameters**
delivery_collection	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 200}
loading_initial_unit	{'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 200}
transferring_to_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
unloading_first_transfer	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 100}
handling_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
loading_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
transferring_to_second_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
unloading_second_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
handling_second_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
loading_second_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
transferring_to_terminal_unit	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
unloading_terminal_unit	{'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 200}
handling_terminal_unit	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 300}
handling_courier	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 300}
delivery	{'learning_rate': 0.3, 'max_depth': 6, 'n_estimators': 200}

Table A.5: Hyperparameters for Random Forest 15 Steps Models

Step	Hyperparameters
delivery_collection	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 200}
loading_initial Unit	{'learning_rate': 0.3, 'max_depth': 5, 'n_estimators': 200}
transferring_to_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 200}
unloading_first_transfer	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 100}
handling_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
loading_first_transfer	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 100}
transferring_to_terminal_unit	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 100}
unloading_terminal_unit	{'learning_rate': 0.2, 'max_depth': 5, 'n_estimators': 300}
handling_terminal_unit	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 200}
handling_courier	{'learning_rate': 0.2, 'max_depth': 7, 'n_estimators': 100}
delivery	{'learning_rate': 0.2, 'max_depth': 6, 'n_estimators': 300}

Table A.6: Hyperparameters for Random Forest 11 Steps Models

Step	Selected Features
delivery_collection	initial_unit_id, terminal_unit_id, terminal_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection
loading_initial_unit	initial_unit_id, terminal_unit_id, terminal_town_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, week_loading_initial
transferring_to_first_transfer	initial_unit_id, terminal_unit_id, terminal_town_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, week_loading_initial, dur_transfer_1st_transfer, week_transfer_1st_transfer
unloading_first_transfer	initial_unit_id, terminal_unit_id, terminal_town_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer
handling_first_transfer	initial_unit_id, terminal_unit_id, terminal_town_id, dur_delivery_collection, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer
loading_first_transfer	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer
transferring_to_terminal_unit	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, week_unloading_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal
unloading_terminal_unit	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, week_unloading_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, week_unloading_terminal
handling_terminal_unit	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, week_transfer_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal
handling_courier	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, week_transfer_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, dur_handling_courier, hour_handling_courier, week_handling_courier
delivery	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, hour_loading_1st_transfer, week_loading_first_transfer, dur_transfer_terminal, week_transfer_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, week_handling_courier, dur_delivery, hour_delivery, week_delivery

Table A.7: Selected Features For Final CatBoost 11-Step Models

Step	Columns
delivery_collection	initial_unit_id, terminal_town_id, sender_town, receiver_district, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection
loading_initial_unit	initial_unit_id, terminal_town_id, sender_town, receiver_district, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_Loading-Initial Unit, week_loading_initial
transferring_to_first_transfer	initial_unit_id, terminal_town_id, sender_town, receiver_district, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_transfer_1st_transfer, hour_Transferring to 1st Transfer
unloading_first_transfer	initial_unit_id, terminal_town_id, sender_town, receiver_district, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, hour_Unloading-1st Transfer, week_unloading_1st_transfer
handling_first_transfer	initial_unit_id, terminal_town_id, receiver_district, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, hour_Unloading-1st Transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_Handling-1st Transfer, week_handling_1st_transfer
loading_first_transfer	initial_unit_id, terminal_town_id, receiver_district, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, hour_Unloading-1st Transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_Loading-1st Transfer, week_loading_1st_transfer
transferring_to_terminal_unit	initial_unit_id, terminal_town_id, receiver_district, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_unloading_1st_transfer, hour_Unloading-1st Transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, dur_loading_1st_transfer, hour_Loading-1st Transfer, week_loading_1st_transfer, dur_transfer_terminal, week_transfer_terminal
unloading_terminal_unit	initial_unit_id, terminal_town_id, receiver_district, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_unloading_1st_transfer, hour_Unloading-1st Transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, dur_loading_1st_transfer, hour_Loading-1st Transfer, dur_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_Unloading-Terminal Unit, week_unloading_terminal
handling_terminal_unit	initial_unit_id, terminal_town_id, receiver_district, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_Loading-1st Transfer, dur_transfer_terminal, week_transfer_terminal, hour_Unloading-Terminal Unit, week_unloading_terminal, dur_handling_terminal, hour_Handling-Terminal Unit, week_handling_terminal
handling_courier	initial_unit_id, terminal_town_id, receiver_district, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_Loading-1st Transfer, dur_transfer_terminal, hour_Unloading-Terminal Unit, dur_handling_terminal, hour_Handling-Terminal Unit, week_handling_terminal, dur_handling_courier, hour_Handling-Courier, week_handling_courier
delivery	initial_unit_id, terminal_town_id, receiver_district, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_Loading-1st Transfer, dur_transfer_terminal, hour_Unloading-Terminal Unit, dur_handling_terminal, hour_Handling-Terminal Unit, week_handling_terminal, dur_handling_courier, week_handling_courier, dur_delivery, hour_Delivery, week_Delivery

Table A.8: Selected Features For Final XGBoost 11-Step Models

Step	Selected Features
delivery_collection	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection
loading_initial_unit	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial
transferring_to_first_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_transfer_1st_transfer, hour_transfer_1st_transfer
unloading_first_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer
handling_first_transfer	initial_unit_id, terminal_town_id, initial_district_id, dur_delivery_collection, week_delivery_collection, week_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer
loading_first_transfer	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer
transferring_to_terminal_unit	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, week_unloading_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal
unloading_terminal_unit	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, week_unloading_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, week_unloading_terminal
handling_terminal_unit	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, week_transfer_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal
handling_courier	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, week_transfer_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, dur_handling_courier, hour_handling_courier, week_handling_courier
delivery	initial_unit_id, terminal_unit_id, terminal_town_id, week_delivery_collection, week_loading_initial, week_transfer_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, week_transfer_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, week_handling_courier, dur_delivery, hour_delivery, week_delivery

Table A.9: Selected Features For Final RandomForest 11-Step Models

Step	Selected Features
delivery_collection	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection
loading_initial_unit	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial
transferring_to_first_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial
unloading_first_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer
handling_first_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer
loading_first_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer
transferring_to_terminal_unit	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal
unloading_terminal_unit	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, dur_unloading_terminal
handling_terminal_unit	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, dur_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal
handling_courier	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, dur_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, dur_handling_courier, hour_handling_courier, week_handling_courier
delivery	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_terminal, dur_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, dur_handling_courier, hour_handling_courier, week_handling_courier, dur_delivery, hour_delivery, week_delivery

Table A.10: Selected Features For Final LogisticRegression 11-Step Models

Step	Columns
delivery_collection	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection
loading_initial_unit	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial
transferring_to_first_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer
unloading_first_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer
handling_first_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer
loading_first_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer
transferring_to_second_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, week_transfer_2st_transfer
unloading_second_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer
handling_second_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer, dur_handling_2st_transfer, week_handling_2st_transfer
loading_second_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer, dur_handling_2st_transfer, week_handling_2st_transfer, hour_loading_2st_transfer, week_loading_2st_transfer
transferring_to_terminal_unit	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer, dur_handling_2st_transfer, week_handling_2st_transfer, hour_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal
unloading_terminal_unit	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer, dur_handling_2st_transfer, week_handling_2st_transfer, hour_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, week_unloading_terminal
handling_terminal_unit	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer, dur_handling_2st_transfer, week_handling_2st_transfer, hour_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, week_handling_terminal
handling_courier	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer, dur_handling_2st_transfer, week_handling_2st_transfer, hour_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, week_handling_terminal, dur_handling_courier, hour_handling_courier, week_handling_courier
delivery	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, week_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer, dur_handling_2st_transfer, week_handling_2st_transfer, hour_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, week_handling_terminal, dur_handling_courier, hour_handling_courier, week_handling_courier, dur_delivery, hour_delivery, week_delivery

Table A.11: Selected Features For Final LogisticRegression 15-Step Models

Step	Selected Features
delivery_collection	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_district_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection
loading_initial_unit	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial
transferring_to_first_transfer	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, hour_loading_initial, week_loading_initial, dur_transfer_1st_transfer, hour_transfer_1st_transfer, week_transfer_1st_transfer
unloading_first_transfer	initial_unit_id, terminal_town_id, initial_district_id, dur_delivery_collection, week_delivery_collection, hour_loading_initial, week_loading_initial, hour_transfer_1st_transfer, week_transfer_1st_transfer, dur_unloading_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer
handling_first_transfer	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, week_loading_initial, hour_transfer_1st_transfer, week_transfer_1st_transfer, dur_unloading_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer
loading_first_transfer	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, week_loading_initial, hour_transfer_1st_transfer, week_transfer_1st_transfer, dur_unloading_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer
transferring_to_second_transfer	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, hour_transfer_1st_transfer, week_transfer_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_2nd_transfer, hour_transfer_2nd_transfer, week_transfer_2nd_transfer
unloading_second_transfer	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, hour_transfer_1st_transfer, week_transfer_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2nd_transfer, hour_transfer_2nd_transfer, week_transfer_2nd_transfer, dur_unloading_2nd_transfer, hour_unloading_2nd_transfer, week_unloading_2nd_transfer
handling_second_transfer	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, hour_transfer_1st_transfer, week_transfer_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2nd_transfer, hour_transfer_2nd_transfer, week_transfer_2nd_transfer, hour_unloading_2nd_transfer, week_unloading_2nd_transfer, dur_handling_2nd_transfer, hour_handling_2nd_transfer, week_handling_2nd_transfer
loading_second_transfer	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, hour_transfer_1st_transfer, week_transfer_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2nd_transfer, hour_transfer_2nd_transfer, week_transfer_2nd_transfer, hour_unloading_2nd_transfer, week_unloading_2nd_transfer, dur_handling_2nd_transfer, week_handling_2nd_transfer, dur_loading_2nd_transfer, hour_loading_2nd_transfer, week_loading_2nd_transfer
transferring_to_terminal_unit	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, week_transfer_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2nd_transfer, hour_transfer_2nd_transfer, week_transfer_2nd_transfer, hour_unloading_2nd_transfer, week_unloading_2nd_transfer, dur_handling_2nd_transfer, week_handling_2nd_transfer, dur_loading_2nd_transfer, hour_loading_2nd_transfer, week_loading_2nd_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal
unloading_terminal_unit	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, week_transfer_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2nd_transfer, hour_transfer_2nd_transfer, week_transfer_2nd_transfer, hour_unloading_2nd_transfer, week_unloading_2nd_transfer, dur_handling_2nd_transfer, week_handling_2nd_transfer, dur_loading_2nd_transfer, hour_loading_2nd_transfer, week_loading_2nd_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, week_unloading_terminal
handling_terminal_unit	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, week_transfer_1st_transfer, hour_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2nd_transfer, hour_transfer_2nd_transfer, week_transfer_2nd_transfer, hour_unloading_2nd_transfer, week_unloading_2nd_transfer, dur_handling_2nd_transfer, week_handling_2nd_transfer, dur_loading_2nd_transfer, hour_loading_2nd_transfer, week_loading_2nd_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal
handling_courier	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, week_transfer_1st_transfer, hour_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2nd_transfer, hour_transfer_2nd_transfer, week_transfer_2nd_transfer, hour_unloading_2nd_transfer, week_unloading_2nd_transfer, dur_handling_2nd_transfer, week_handling_2nd_transfer, dur_loading_2nd_transfer, hour_loading_2nd_transfer, week_loading_2nd_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, dur_handling_courier, week_handling_courier
delivery	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, week_transfer_1st_transfer, hour_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2nd_transfer, hour_transfer_2nd_transfer, week_transfer_2nd_transfer, hour_unloading_2nd_transfer, week_unloading_2nd_transfer, dur_handling_2nd_transfer, week_handling_2nd_transfer, dur_loading_2nd_transfer, hour_loading_2nd_transfer, week_loading_2nd_transfer, dur_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, week_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, dur_handling_courier, week_handling_courier, dur_delivery, hour_delivery, week_delivery

Table A.12: Selected Features For Final CatBoost 15-Step Models

Step	Selected Features
delivery_collection	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection
loading_initial_unit	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial
transferring_to_first_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, dur_transfer_1st_transfer, hour_transfer_1st_transfer, week_transfer_1st_transfer
unloading_first_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_transfer_1st_transfer, hour_transfer_1st_transfer, week_transfer_1st_transfer, dur_unloading_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer
handling_first_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer
loading_first_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer
transferring_to_second_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, week_transfer_2st_transfer
unloading_second_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, dur_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer
handling_second_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, dur_handling_2st_transfer, hour_handling_2st_transfer, week_handling_2st_transfer
loading_second_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, dur_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, dur_handling_2st_transfer, hour_handling_2st_transfer, week_handling_2st_transfer, dur_loading_2st_transfer, week_loading_2st_transfer
transferring_to_terminal_unit	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, dur_handling_2st_transfer, hour_handling_2st_transfer, week_handling_2st_transfer, dur_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal
unloading_terminal_unit	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, dur_handling_2st_transfer, hour_handling_2st_transfer, dur_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal
handling_terminal_unit	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, dur_handling_2st_transfer, dur_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal
handling_courier	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, dur_handling_2st_transfer, dur_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, dur_handling_courier, hour_handling_courier, week_handling_courier
delivery	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, dur_handling_2st_transfer, dur_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, dur_handling_courier, hour_handling_courier, week_handling_courier, dur_delivery, hour_delivery, week_delivery

Table A.13: Selected Features For Final XGBoost 15-Step Models

Step	Selected Features
delivery_collection	initial_unit_id, terminal_xdock_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection
loading_initial_unit	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial
transferring_to_first_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, week_loading_initial, dur_transfer_1st_transfer, hour_transfer_1st_transfer, week_transfer_1st_transfer
unloading_first_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_transfer_1st_transfer, hour_transfer_1st_transfer, week_transfer_1st_transfer, dur_unloading_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer
handling_first_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer
loading_first_transfer	initial_unit_id, terminal_town_id, initial_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_transfer_1st_transfer, dur_unloading_1st_transfer, week_unloading_1st_transfer, dur_handling_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer
transferring_to_second_transfer	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, hour_transfer_1st_transfer, week_transfer_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, dur_loading_1st_transfer, hour_loading_1st_transfer, week_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, week_transfer_2st_transfer
unloading_second_transfer	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, hour_transfer_1st_transfer, week_transfer_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, week_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer
handling_second_transfer	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, hour_transfer_1st_transfer, week_transfer_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, week_transfer_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer, dur_handling_2st_transfer, hour_handling_2st_transfer, week_handling_2st_transfer
loading_second_transfer	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, hour_transfer_1st_transfer, week_transfer_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, week_transfer_2st_transfer, hour_unloading_2st_transfer, week_unloading_2st_transfer, dur_handling_2st_transfer, week_handling_2st_transfer, dur_loading_2st_transfer, hour_loading_2st_transfer, week_loading_2st_transfer
transferring_to_terminal_unit	initial_unit_id, terminal_town_id, initial_district_id, week_delivery_collection, hour_loading_initial, week_transfer_1st_transfer, hour_unloading_1st_transfer, week_unloading_1st_transfer, hour_handling_1st_transfer, week_handling_1st_transfer, week_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, week_transfer_2st_transfer, hour_unloading_2st_transfer, hour_handling_2st_transfer, week_handling_2st_transfer, dur_loading_2st_transfer, hour_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal
unloading_terminal_unit	initial_unit_id, terminal_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, dur_loading_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, dur_handling_2st_transfer, hour_handling_2st_transfer, dur_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal
handling_terminal_unit	initial_unit_id, terminal_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, dur_handling_2st_transfer, dur_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, hour_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal
handling_courier	initial_unit_id, terminal_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, dur_handling_2st_transfer, dur_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, dur_handling_courier, hour_handling_courier, week_handling_courier
delivery	initial_unit_id, terminal_town_id, initial_district_id, dur_delivery_collection, hour_delivery_collection, week_delivery_collection, dur_loading_initial, hour_loading_initial, dur_unloading_1st_transfer, dur_handling_1st_transfer, dur_transfer_2st_transfer, hour_transfer_2st_transfer, dur_unloading_2st_transfer, hour_unloading_2st_transfer, dur_handling_2st_transfer, dur_loading_2st_transfer, week_loading_2st_transfer, dur_transfer_terminal, week_transfer_terminal, dur_unloading_terminal, hour_unloading_terminal, dur_handling_terminal, hour_handling_terminal, week_handling_terminal, dur_handling_courier, hour_handling_courier, week_handling_courier, dur_delivery, hour_delivery, week_delivery

Table A.14: Selected Features For Final Random Forest 15-Step Models

CURRICULUM VITAE

Personal Information

Name and Surname: Büşra Ülkü Küp

Academic Background

Bachelor's Degree Education:

Computer Engineering (2014-2019) at Kadir Has University

Industrial Engineering (2016-2020) at Kadir Has University

Post Graduate Education:

Master of Science (2020-2023) in Industrial Engineering at Kadir Has University

Foreign Languages:

English (Advanced)

German (Intermediate)

Work Experience

Data Scientist (2021-) in HepsiJET