Predicting and explaining turnaround delays through counterfactuals

A Schiphol Airport case study

Author:	Daan van der Veldt
Student number:	500680616
Company supervisor:	Koos Noordeloos
Thesis supervisor:	Debarati Bhaumik
Second Assessor:	Raymond Zwaal
Submission date:	19th of June 2024

Blank page

Cover page picture by Sisson (2017)

Executive Summary

This research addresses the deviations between Target Off-Block Time (TOBT) and Actual Off-Block Time (AOBT) in aircraft turnaround processes at Schiphol Airport. Accurate TOBT prediction is critical for effective airport operations, significantly impacting runway capacity and overall flight schedules. Despite the implementation of the Airport-Collaborative Decision Making (A-CDM) system, differences between TOBT and AOBT frequently occur, leading to delays and inefficiencies. The primary aim of this research is to investigate the main factors influencing these deviations, which led to the research question: "What are the main factors influencing Target Off-Block Time estimation deviations during airport turnaround processes at Schiphol Airport?"

The research utilizes historical data from Schiphol's DeepTurn IT department and CDM data from Luchtverkeerleiding Nederland (LVNL). Various statistical analyses and machine learning models, including Random Forest, LightGBM, and XGBoost, are employed to identify key variables affecting TOBT deviations. Counterfactual explanations are integrated to enhance the interpretability of these models, providing actionable insights into the minimal changes required to chnage the model's predictions. This approach aims to offer practical guidance for operational improvements and decision-making.

The study identifies several operational and contextual factors that significantly influence TOBT deviations. Key variables include the duration of baggage handling and pushback events, as well as specific time-related factors such as the time of day and month. These findings suggest that certain operational activities and scheduling contexts play a crucial role in predicting and managing TOBT deviations.

To facilitate post hoc analysis for airport stakeholders, a user-friendly dashboard was developed using Flask and Dash frameworks. This dashboard visualizes the predictions and counterfactual insights, enabling stakeholders to monitor and address TOBT deviations. By integrating these predictive models and counterfactual explanations into a dashboard, LVNL can do post hoc analyses based on the insights that are created.

While the study offers valuable insights, it is not without limitations. The quality of the data, particularly the noise and inaccuracies in event documentation by DeepTurn cameras created issues. Some events may be recorded inaccurately or not at all, affecting the reliability of the model predictions. Examples are found where events were started but not ended in the data. Moreover, to be able to accurately record duration for events as refueling and catering services, clear start and end points are missing. This leads to errors in interpreting the event durations.

To enhance TOBT predictions and turnaround efficiency at Schiphol Airport, it is essential to improve data collection methods and redefine event protocols. Using higher frame rate cameras and advanced AI algorithms can significantly improve the accuracy of detecting and classifying turnaround activities. Ensuring full camera coverage of all critical ramp areas and performing regular maintenance and calibration will further enhance data quality. Finding improvements in data gathering can potentially reduce the noise in the data.

Redefining event protocols involves developing standardized methods for identifying precise start and end points of each event. For example, fuel events should be recorded from the moment the first fuel truck stops in position to when the last fuel truck leaves. Similarly, pushback events should be defined more accurately by capturing both the start and end of the tug idle connected phase. These changes will improve the accuracy and reliability of predictive models for TOBT, leading to better decision-making and reduced turnaround delays.

List of abbreviations

A-CDM	Airport-Collaborative Decision Making
ACTREC	Actional Resource method
AI	Artificial Intelligence
ANFIS	Adaptive Network-based Fuzzy Inference System
ANN	Artificial Neural Networks
ANOVA	Analysis Of Variance
AOBT	Actual Off-Block Time
ATC	Air Traffic Control
ATOT	Actual Take-Off Time
CEML	Counterfactuals for Explaining Machine Learning
C-HVAE	Counterfactual Conditional Heterogeneous Autoencoder
DACE	Distribution-Aware Counterfactual Explanation
DICE	Diverse Counterfactual Explanations
DOBT	Delta Off-Block Time
EIBT	Estimated In-Block Time
EOBT	Estimated Off-Block Time
ETOT	Estimated Take-Off Time
LightGBM	Light Gradient Boosting Machine
LIME	Local Interpretable Model-agnostic Explanations
LOF	Local Outlier Factor
LVNL	Luchtverkeersleiding Nederland
MACE	Model-Agnostic Counterfactual Explanation
MAE	Mean Absolut Error
MTTT	Minimum Turnaround Time
OCEAN	Optimal Counterfactual Explainer
ORDCE	Ordered Counterfactual Explanation
OTP	On-Time Performance
PCA	Principle Components Analysis
PEGT	Predicted End of Ground handling Time
SHAP	Shapley Additive Explanations
SOBT	Scheduled Off-Block Time
SVM	Support Vector Machine
T-DPI-t	Target Departure Planning Information-target
TOBT	Target Off-Block Time
TSAT	Target Start-Up Approval Time
TTOT	Target Take-Off Time
VIF	Variance Inflation Factor
XAI	Explainable Artificial Intelligence
XGBoost	Extreme Gradient Boosting

Table of contents

Executive Summary 3
List of abbreviations 4
1. Introduction 7
1.1 Problem Statement 7
1.2 Research objectives and questions 7
1.3 Theoretical and Methodological Approach 8
1.4 Contribution to Practice and Academia 8
1.5 Report structure 9
2. Literature review 10
2.1 A-CDM and TOBT 10
2.2 Important features 11
2.3 TOBT prediction models 12
2.4 Implementing a model for Schiphol Airport 14
2.5 Machine learning explainability 14
2.6 Dashboard 16
3. Methodology 18
3.1 Variable selection 18
3.2 Data collection 18
3.3 Data cleaning 19
3.4 Data analysis 20
3.4.1 Statistical tests 20
3.4.2 Machine learning algorithms 21
3.4.3 Counterfactuals 24
3.5 Dashboard 25
3.6 Reliability and validity 26
3.7 Ethical considerations 27
4. Results 28
4.1 Statistical analysis 28
4.1.1 Numeric features 28
4.1.2 Categorical features 29
4.2 Prediction models 30
4.3 Counterfactual explanations 31
4.3.1 Counterfactual instances 31
4.3.2 Global feature importance 32

4.4 Dashboard 32
5. Discussion 34
5.1 Literature findings 34
5.2 Feature importance 35
5.2.1 Statistical results 35
5.2.2 Feature importance from prediction models 36
5.2.3 Feature importance from global counterfactuals 38
5.2.4 Most important features 39
5.3 External Factors Influencing Turnaround Delay 39
5.4 Post analysis dashboard for turnaround delay 40
5.5 Limitations 41
5.5.1 Noise in the Data 41
5.5.2 Hard-to-Define Events 41
5.5.3 Effect on research 42
6. Conclusion 44
7. Recommendations 45
References 47
List of Appendices 60

1. Introduction

In the highly dynamic world of aviation, minor delays can trigger a domino effect that leads to significant disruptions in flight schedules (Fan & Zhuang, 2020). Preventing this from happening underscores the importance of planned target times to closely mirror real-time conditions. An important contributor to these delays is the unexpected variation in aircraft turnaround times (Zhou et al., 2019). At Schiphol Airport, as well as at other European airports, the Airport-Collaborative Decision Making (A-CDM) system enhances operational planning by integrating inputs from stakeholders throughout the inbound, turnaround, and outbound phases of flights (EUROCONTROL, 2017). This research investigates factors that influence target times within the A-CDM system, specifically the off-block time. This chapter aims to provide a comprehensive overview of the research problem (1.1), objectives, and questions (1.2). It furthermore introduces the theoretical and methodological approaches used in the research (1.3). It explains how this research contributed to practice and academic purposes (1.4) and outlines the report's structure (1.5).

1.1 Problem Statement

By using this data, A-CDM calculates several target times, with the Target Off-Block Time (TOBT) being one The Airport Collaborative Decision Making (A-CDM) system plays a crucial role in optimizing airport operations by calculating several target times, with the Target Off-Block Time (TOBT) being one of the key milestones for planning (Schiphol Airport, 2021). TOBT represents the precise moment an aircraft is fully prepared to leave the gate post-turnaround. However, despite frequent updates to TOBT by the A-CDM system, discrepancies often arise between TOBT and the Actual Off-Block Time (AOBT). These discrepancies can lead to challenges in runway capacity management and cause take-off delays (Strohmeier et al., 2018).

This research aims to investigate the primary factors influencing the deviations between TOBT and AOBT, using data from Luchtverkeerleiding Nederland (LVNL) and the DeepTurn data at Schiphol Airport. From the perspective of LVNL, there has been no previous research into turnaround delays from a data-driven perspective. There is currently insufficient knowledge regarding the influence of various events during the turnaround process and the impact of different handler companies responsible for these events.

By analyzing this data, this research seeks to identify and understand the key factors contributing to these deviations, thereby providing insights that can enhance the efficiency of airport operations and reduce take-off delays.

1.2 Research objectives and questions

The main objective of this research is to identify the key factors influencing the turnaround delays at Schiphol Airport. This is done by doing statistical tests, create machine learning models that lead to insights in important model predictors and generating counterfactual explanations that indicate feature importance. By doing so, it aims to identify factors that can be improved on, enhancing overall operational efficiency, and minimizing delays. Furthermore, another objective of this research is to create a post turnaround analysis tool. This tool can be used to create counterfactual explanations for individual turnarounds. By using counterfactual explanations, the minimal required change in the turnaround can be identified to predict a turnaround as not delayed rather than delayed. This objective translates into the following main research question:

What are the main factors influencing Target Off-Block Time estimation deviations during airport turnaround processes at Schiphol Airport?

To systematically address this main question, several sub-questions have been created to guide the investigation:

- *What is Target Off-Block Time and how is it estimated?*

This sub-question provides foundational knowledge on TOBT and its estimation process, essential for understanding the context and significance of any deviations.

- *Which prediction machine learning models are currently used for Target Off-Block Time estimation?*

This sub-question explores existing machine learning models to determine their effectiveness and applicability, providing a basis for selecting models for TOBT estimation.

- *How can the machine learning models be used to predict Target Off-Block Time at Schiphol Airport?*

By investigating how machine learning models to predict TOBT at Schiphol can be used, the research can be constructed based on airport specific data and characteristics.

- *Which features are important when estimating Target Off-Block Time?*

Identifying key metrics helps to pinpoint the variables that influence TOBT estimation, allowing for a more targeted investigation into factors causing deviations.

- *How can machine learning explainability lead to counterfactual insights?*

Understanding ML model explainability allows for insights into why certain predictions are made, helping to identify and mitigate factors causing deviations in TOBT.

- *How can a dashboard be created to overview the model's predictions?*

Developing a dashboard facilitates the visualization and monitoring of model predictions, enabling stakeholders to quickly identify and address deviations in TOBT.

These sub-questions are designed to be clear, feasible, interconnected, and relevant, ensuring they collectively address the research question effectively.

1.3 Theoretical and Methodological Approach

This research will use a mixed-methods approach, integrating statistical analysis and machine learning techniques to evaluate the factors affecting TOBT deviations. First, a literature review will be conducted to understand the existing methodologies and frameworks used in TOBT estimation (Chapter 2). This will involve an overview of different models used for TOBT predictions and variables found to be important contributors to delay in previous research. Furthermore, it will involve a review of different counterfactual explanation methodologies and a review of methodologies that can be used for creating dashboards. Following this, the methodology section (Chapter 3) will outline the data collection, handling, cleaning, and analysis procedures. The analysis will include the development and evaluation of predictive machine learning models, generation of counterfactual explanations to understand model predictions, and creation of a user-friendly dashboard to visualize these insights. Furthermore, model feature importance and global counterfactual feature importance are outlined an explained.

1.4 Contribution to Practice and Academia

The findings of this research are expected to significantly benefit the aviation industry, particularly in enhancing the operational efficiency of airports by improving TOBT accuracy. For people working in practise, this research provides actionable insights into the factors affecting turnaround times and offers a predictive tool to mitigate delays (Lulli & Odoni, 2007). Furthermore, the findings are especially valuable for Schiphol Airport and LVNL, since the case study using Schiphol's data directly identify key factors that influence turnaround delay.

Academically, this study contributes to the existing body of knowledge on airport operations and machine learning applications in aviation, offering new perspectives and methodologies for future research. Furthermore, no earlier research into the use of counterfactual machine learning explainability has been found, thereby presenting a significant research gap.

1.5 Report structure

This report is constructed in the following manner. After the introduction, the report continues with a literature review (chapter 2). In this chapter, previous research outlined based on each of the sub questions. Thereafter, the methodology section (chapter 3) of this research explains the way research was done. It primarily functions as an explanation of how data was gathered, handled, and analysed. Then, the results of the analysis are shown (chapter 4). These consists of statistical results, prediction models, counterfactual explanations, and a dashboard. Afterwards, these results are discussed (chapter 5) to be able to reflect on them. This research is summarized in the conclusion (chapter 6) and ends with some recommendations (chapter 7).

2. Literature review

The literature review of this research is constructed in the following manner. To create a understanding of the A-CDM process and the TOBT variable, documentation as well as research into this subject is issued (2.1). Secondly, previous research is called up on to understand which variables have been found influencing TOBT delays (2.2). Then, similar research using different delay prediction models are overviewed (2.3). Since this research is specifically using data from Schiphol Airport, the next sections explain the A-CDM specific complications at Schiphol Airport (2.4). The next sections overviews research into counterfactual machine learning explainability techniques (2.5). This section is part of a greater literature review done into this subject, which can be found in Appendix 1. Finally, the last section of this chapter explains previous research into using a dashboard which includes machine learning and explainability techniques (2.6).

2.1 A-CDM and TOBT

Due to the rising demand of air traffic, capacity at airports is often challenged. For these airports, A-CDM is key for air traffic management (Okwir et al., 2017). A-CDM aims to improve predictability of air traffic, by creating a collaborative environment between airport operators, aircraft operators, ground handlers, air traffic controllers (ATC), and the Network Manager (EUROCONTROL, 2017a). By facilitating efficient turnaround processes, enhancing flight predictability through real-time data exchange, and optimizing gate management, A-CDM improves airport operations and decreases delays while maximizing available capacity and resources. However, successful A-CDM applications require careful stakeholder engagement (IATA Airline A-CDM Coordination Group, 2018). This way of handling flights is fully implemented at 32 European airports after being first introduced at München Airport in 2007 (EUROCONTROL, 2017a). The concept of A-CDM follows the Milestone Approach, in which every milestone is based on the flight process progression. The Milestone Approach's main goals are to improve situational awareness at all stages of the flight by identifying key events, defining information updates, defining data quality criteria, connecting arriving, and departing flights, enabling early decision-making during disruptions, and generally enhancing information quality (EUROCONTROL, 2017b). During the milestones, target times such as TOBT are being updated based on milestone achievements. For example, if an aircraft happened to arrive with a delay, TOBT might be extended due to shortened available time of the turnaround phase. An overview of this approach is found in figure 1.

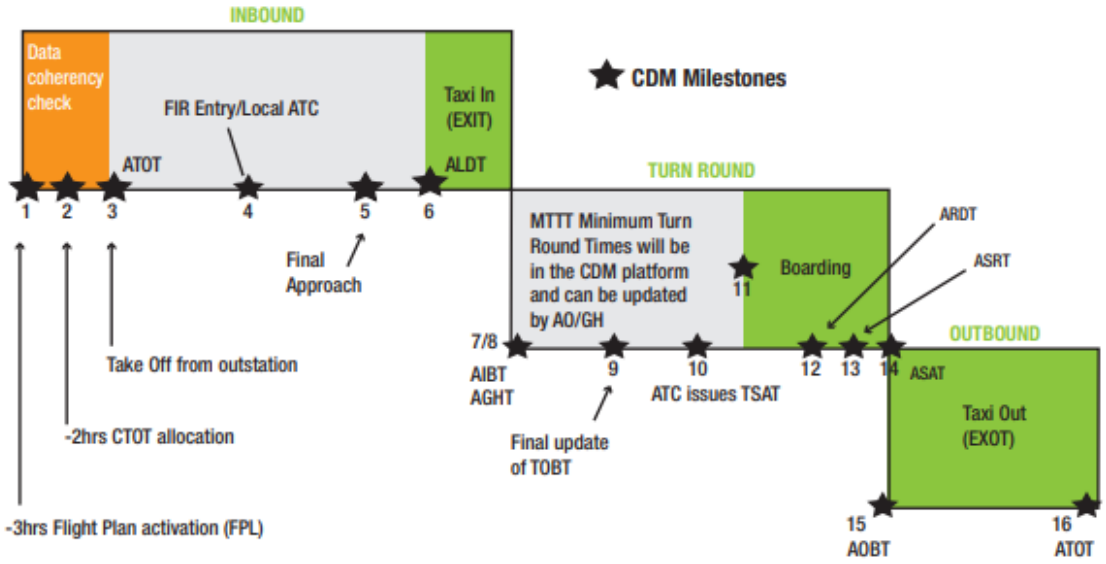


Figure 1: Milestone Approach (EUROCONTROL, 2017b)

TOBT is seen as one of the key variables. It is seen as a main input of runway capacity planning (Schiphol Airport, 2024). The initial TOBT is equal to the Schedule Off-Block Time (SOBT), which is based on the scheduled time of departure for the flight. The third milestone is the first time that the TOBT can be changed. This milestone is reached when the flight locates 2 hours before the Estimated Off-Block Time (EOBT). The network operator of the airport is informed on possible discrepancies in Actual Take-Off Time (ATOT) from the origin airport. This information is shared with the network operator by a Target Departure Planning Information-target (T-DPI-t) message. The system will then automatically calculate if the Estimated in Block Time (EIBT), the moment the aircraft arrives at its parking position after landing and taxing, and the Minimum Turnaround Time (MTTT) will take longer than the SOBT. The MTTT is determined based on several variables, such as aircraft type, type of stand and airline procedures. If the calculation exceeds the SOBT, the system automatically updates the TOBT. At milestone 4, 5 and 6, similar procedures take place considering automatic recalculations of the TOBT. The fourth milestone occurs by a local radar update, the fifth milestone occurs during the final approach and the sixth during the landing. The final automatic update to the TOBT occurs when milestone 7 is reached. At this point, the aircraft will be in-blocks and ground handling commence. Milestone 8 occurs when an aircraft's ground handling has started. This milestone only occurs to specific flights that did not follow a normal turnaround. This for example occurs when it's the aircraft's first flight of the day. Milestone 9 is the final confirmation of TOBT and is the last time TOBT changes can occur during the Milestone Approach. This time, the TOBT is not automatically being changed but manually by an aircraft operator or ground handler. This happens on a predetermined interval before the EOBT. This interval can change between airports. This update to the TOBT is manual, based on an estimation on the turnaround process (EUROCONTROL, 2017b). While being done by specialized personnel, a TOBT estimation done by human assessment can be inaccurate. This while TOBT is an important metric that influences A-CDM positively (Rott et al., 2023).

2.2 Important features

There have been multiple research projects done that investigated variables influencing TOBT delays. Postorino et al.'s researched (2020) the impact of disruptive events on airport airside operations, offering insights into mitigating operational disruptions and enhancing efficiency through detailed simulation modelling. They listed the most important activities during a turnaround process and divided these activities into sub activities. This gives insights in turnaround process activities and shows the effectiveness of dividing these in smaller sub activities. They found that a lack of available ground personnel has a correlation with average delay. Volt et al. (2023) researched the possibilities of quantification of factors influencing aircraft handling processes and TOBT predictions. They investigated features as number of passengers, number of baggage carts, (un)loading duration, (un)boarding duration and refuelling duration. These features were examined for over 12,000 flights that did a turnaround at Václav Have Airport Prague. They found significant correlations, based on its P-value, in 9 features. These were features related to the number of carts at unloading and loading, the duration of unloading, the loading times, the boarding times, and the fuelling times. This research is of use since it presents a method of measuring metrics' correlation to TOBT and presents results that may be compared to this research's results. Rebollo and Balakrishnan (2014) developed random forest ML model to predict root delay at multiple American airports. They found that delay in their model was most caused by the time-of-day, making it the most important explanatory variable. It indicates that time-of-day is an important feature to research since this metric can also affect TOBT delays in peak hours at Schiphol Airport. Using TOBT as a feature, Dalmau et al. (2019) employed machine learning approaches to investigate the discrepancy between ATOT and Estimated Take-Off Times (ETOTs). They found in their Shapley analysis that the available time for turnaround significantly impacted delays, with lower values indicating potential delay. Supporting this, De Falco (2023) found that available turnaround time was the most important value, when they created a model predicting TOBT delays and did a Shapley analysis.

Feature	Paper	Explanation
Available ground personnel	Postorino et al (2020)	Amount of personnel available to execute ground handling during turnaround.
Number of passengers	Volt et al. (2023)	Total count of passengers on board the aircraft, influencing boarding and disembarkation times as well as other turnaround activities.
Number of baggage carts	Volt et al. (2023)	Total count of baggage carts used during loading and unloading, affecting turnaround times and efficiency of baggage handling.
(Un)loading duration	Volt et al. (2023)	Time taken to load or unload baggage or cargo onto or off the aircraft, influencing overall turnaround time.
(Un)boarding duration	Volt et al. (2023)	Time taken for passengers to embark or disembark from the aircraft, affecting overall turnaround time and efficiency of passenger handling.
Refuelling duration	Volt et al. (2023)	Time taken to refuel the aircraft, impacting turnaround time and dependent on factors such as fuel capacity and refuelling infrastructure.
Time-of-day	Rebollo and Balakrishnan (2014)	The specific hour of the day when the turnaround process occurs, which can influence various aspects of operations and delays, especially during peak hours.
Available turnaround time	Dalmau et al. (2019) De Falco et al. (2023)	The duration of time available for completing the turnaround process, which significantly impacts the likelihood of delays and efficient turnaround operations.

Table 1: Important features in previous research

2.3 TOBT prediction models

Machine learning techniques have gained significant traction in predicting flight delays, including TOBT predictions, due to their ability to handle complex and large datasets. Various research projects have explored different machine learning models to enhance the accuracy and reliability of these predictions.

Rebollo et al. (2014) applied random forests to predict root delays, testing their model at several airports in the United States with forecast horizons of 2, 4, 6, and 24 hours. They found that prediction errors increased with the length of the forecast horizon. Khanmohammadi et al. (2014) used an Adaptive Network-based Fuzzy Inference System (ANFIS) to predict root delays, employing the predictions as inputs for a fuzzy decision-making method to sequence arrivals at JFK International Airport. Balakrishna et al. (2010) utilized a reinforcement learning algorithm to predict taxi-out delays. The problem was modeled as a Markov decision process, achieving good performance when run 15 minutes before scheduled departure times at JFK and Tampa Bay International Airports. Lu et al. (2016) developed a recommendation system using the k-Nearest Neighbor algorithm to forecast delays at airports due to propagation effects, highlighting its fast response time and ease of interpretation. Ganesan et al. (2010) employed approximate dynamic programming to predict airport taxi-out times, integrating different data sources to improve prediction accuracy. George and Khan (2015) developed an improved Q-learning approach to predict taxi-out times, enhancing accuracy in dynamic and complex airport environments.

Expanding on these findings, Gui et al. (2019) focused on predicting flight delays using neural networks and random forests, utilizing metrics such as time-of-day variables. They proposed a classification approach for predicting delays, as opposed to traditional regression methods. Dalmau et al. (2019) used Light Gradient Boosting Machine (LightGBM) and Artificial Neural Networks (ANN) to research the difference between ETOT and ATOT, showing a 30 percent improvement in Mean Absolute Error (MAE) compared to base models. Mamdouh et al. (2020) applied Support Vector Machine (SVM) techniques to predict required ground handling resources, achieving high accuracy, and recommending the exploration of other

supervised machine learning models for similar tasks. De Falco et al. (2023) created a probabilistic prediction model for TOBT using eXtreme Gradient Boosting (XGBoost) on data from four airports, significantly improving MAE compared to base models. Yildiz et al. (2022) introduced an innovative system using deep learning and computer vision to automate ground service monitoring during aircraft turnaround processes, improving real-time data processing and precise monitoring. Gao et al. (2015) employed an ANN to predict flight turnaround times at airports, achieving high predictive accuracy with a relative error of less than 25% in 85% of cases.

These diverse applications of machine learning techniques demonstrate their growing importance and effectiveness in predicting various aspects of flight delays, including TOBT. By using historical data and various features, these models significantly improve the accuracy of predictions, thereby improving in optimizing airport operations and resource allocation. To decided which models are most applicable for this research, a pro-cons analysis is done. The results are found in table 2, the models are sorted from simple to complex.

Paper	Model	Pros	Cons
Lu et al. (2016)	k-Nearest Neighbor	Simple to implement, fast response time.	Performance can degrade with high-dimensional data, requires a large amount of memory for storing data.
George and Khan (2015)	Improved Q-Learning	Enhanced prediction accuracy, well-suited for dynamic environments.	Can be computationally intensive, may require significant training data.
Rebollo et al. (2014)	Random Forest	Good performance for classification tasks, interpretable results.	Performance decreases with longer forecast horizons.
Mamdouh et al. (2020)	SVM	High accuracy, effective for classification problems.	Can be sensitive to parameter settings, may not scale well with large datasets.
Dalmau et al. (2019)	LightGBM	High improvement in MAE, efficient for large datasets.	Complex models, require careful tuning and validation.
Gui et al. (2019)	Neural Networks	High accuracy, handles complex relationships in data.	Requires large datasets, may be prone to overfitting.
Ganesan et al. (2010)	Approximate Dynamic Programming	Integrates multiple data sources, flexible model.	Computationally intensive, requires detailed data.
Khanmohammadi et al. (2014)	ANFIS	Combines fuzzy logic with neural networks, good for handling uncertainty.	Can be complex to implement, requires extensive parameter tuning.
Balakrishna et al. (2010)	Reinforcement Learning	Adapts to changing environments, effective for sequential decision-making.	Requires significant computational resources, can be complex to implement.
De Falco et al. (2023)	XGBoost	High predictive performance, robust to overfitting.	Requires careful parameter tuning, can be computationally demanding.
Yildiz et al. (2022)	Deep Learning, Computer Vision	Real-time data processing, precise monitoring, handles unstructured data like images.	Very complex, requires significant computational resources and specialized knowledge to implement.
Gao et al. (2015)	ANN	High predictive accuracy, good for capturing nonlinear relationships.	Requires large datasets, can be prone to overfitting, and requires significant computational power.

Table 2: Important models in previous research

Random Forest, LightGBM, and XGBoost were chosen for this research due to their advantages in handling complex datasets and providing high predictive accuracy. Random Forest is known for its robustness and

ability to handle many input features without significant overfitting. Its interpretability and effectiveness in classification tasks make it a reliable choice for predicting root delays. LightGBM excels in handling large datasets with high efficiency and speed. Its capability to perform well with less training time while providing significant improvements in performance makes it a valuable model for predicting TOBT. XGBoost is chosen for its high predictive performance and robustness to overfitting. It is particularly effective in handling complex relationships within the data, making it suitable for TOBT prediction. These models were selected for their balance of accuracy, efficiency, and robustness, addressing the complexities that are presented in TOBT prediction.

2.4 Implementing a model for Schiphol Airport

The biggest airport of The Netherlands, Schiphol Airport, introduced A-CDM in 2018 (EUROCONTROL, 2017a). As explained in 2.1, some variables of A-CDM such as the final confirmation of the TOBT can vary between airports (EUROCONTROL, 2017b). At Schiphol Airport, this variable is set to 10 minutes before the final TOBT (Schiphol Airport, 2024). Airport turnaround procedures at Schiphol Airport are being revolutionized by Deep Turnaround, an AI-driven solution that offers real-time insights and predictive capabilities (Schiphol Airport, n.d.). Deep Turnaround uses artificial intelligence (AI) image-based processing to identify and document more than 72 turnaround events, enabling actions to reduce disturbances. Its historical, real-time, and predictive data analysis enables stakeholders to optimize turnaround times and make well-informed decisions (Schiphol Airport, n.d.). Collaboration among stakeholders and resource utilization have all improved because of the Deep Turnaround implementation. Deep Turnaround has a positive effect on performance and predictability, as demonstrated by case studies from airports like Eindhoven Airport and Amsterdam Airport Schiphol (Schiphol Airport, n.d.). As a result, it is a useful tool for researching airport operations and improving passenger experiences. The goal of Deep Turnaround is to improve predictability, reduce the number and duration of delays and improve On-Time Performance (OTP). Two cameras are taking snapshots of the current situation every 5 seconds collecting data (WG CDM, 2023). This data is collected in the system of Schiphol and is used for predicting the Predicted End of Ground Handling Time (PEGT), which helps estimating the time a plane is ready for departure. By analysing the turnarounds of narrow body aircraft at Schiphol Airport in 2022, PEGT proved to be 70 percent right of the cases, while TOBT was only 45 percent right (WG CDM, 2023). This underlines the potential of this data source for this research into TOBT and AOBT misalignments. The camera setup and view can be seen in Appendix 2.

2.5 Machine learning explainability

Machine learning explainability is a crucial aspect of modern artificial intelligence, providing insights into how and why models make certain predictions. Explainability ensures that models are transparent, interpretable, and trustworthy, making them more useful in practical applications (Doshi-Velez & Kim, 2017). It involves various techniques and methods to discover the complex decision-making processes of machine learning models.

Explainability in machine learning can be categorized into two main types: intrinsic and post-hoc. Intrinsic explainability refers to the use of interpretable models such as linear regression, decision trees, and rule-based systems. These models are designed to be straightforward, making it easier to understand their predictions directly from their structure (Rudin, 2019). Post-hoc explainability, on the other hand, applies to complex models like neural networks and ensemble methods, which are not naturally interpretable. Post-hoc techniques aim to provide explanations after the model has been trained, offering insights into how the model makes its decisions without altering the model itself (Lipton, 2018). Common post-hoc methods include feature importance scores, partial dependence plots, Local Interpretable Model-agnostic Explanations (LIME), and SHapley Additive exPlanations (SHAP) (Ribeiro et al., 2016; Lundberg & Lee, 2017).

Counterfactual explanations

One particularly powerful post-hoc explainability method is counterfactual explanations. Counterfactual explanations involve generating alternative scenarios by making minimal changes to the input features to alter the model's prediction. These explanations help understand what needs to change for a different outcome to occur, thus providing actionable insights (Wachter et al., 2017). Dandle and Molnar (2023) describe counterfactual explanations as the smallest modification to feature values that transforms a prediction into a predetermined output, improving the interpretability and transparency of machine learning models. However, generating meaningful counterfactuals can be challenging due to the Rashomon effect, where multiple plausible counterfactuals exist for a single instance, potentially leading to ambiguity in interpretation (Molnar, 2022). Despite this, counterfactual explanations are invaluable for clearing up complex model behavior and improving decision-making processes (Karimi et al., 2020). Ferrario and Loi (2022) emphasize the importance of counterfactual explanations within the domain of eXplainable Artificial Intelligence (XAI), highlighting them as essential interfaces between humans and machine learning models. These explanations not only clarify model results but also provide practical guidance on how to achieve different outcomes, making them crucial for understanding feature importance in prediction models.

Various methodologies have been proposed to address the challenge of creating counterfactual explanations in machine learning models, offering insights into why certain predictions are made and what changes could lead to different outcomes. In total, 62 methodologies have been considered in this review and are found in Appendix 1. From this list, 8 methodologies are shortlisted in this literature review due to their applicability. This applicability is based on their feasibility for tabulate data, their actionability, validity, categorical applicability, their optimization strategy, and their GitHub repository availability.

Scoring

In table 3, each counterfactual machine learning explainability has received a score. This score is based on the git quality check evaluation tool (Gcattan, n.d.). This tool evaluates the quality of Git commits and generates indicators based on four criteria: the percentage of commits containing prohibited words, the percentage of commits related to testing, the percentage of branches where the last commit is older than two months, and the percentage of coupled branches (determined by whether a branch is included in another branch's history using git branch contains). The overall score, which ranges from 0% to 100%, reflects the quality of the repository, with higher scores indicating better quality.

XAI	Paper	Advantage	Score
CEML	Artelt (2019)	Offers optimization-based solutions for diverse models	23,82%
ACTREC	Ustun et al. (2019)	Handles actionability constraints with integer programming	49,61%
DACE	Kanamori et al. (2020)	Evaluates plausibility through novel loss function	54,92%
MACE	Karimi et al. (2020)	Facilitates interpretable insights using SMT solvers	51,42%
DICE	Mothilal et al. (2020)	Promotes diversity in generated counterfactuals	58,26%
C-HVAE	Pawelczyk et al. (2020)	Utilizes latent space for counterfactual search	45,48%
OCEAN	Parmentier and Vidal (2021)	Accounts for plausibility and actionability in tree ensembles	41,02%
ORDCE	Kanamori et al. (2021)	Returns counterfactuals with ordered feature change	39,87%

Table 3: Counterfactual XAI methodologies

Counterfactual methodologies overview

Artelt (2019) introduced Counterfactual Explanations via CEML, a toolbox for producing counterfactual explanations across various types of black-box models. Although not formally presented in a paper, CEML offers optimization-based solutions for different model types, contributing to the generation of insightful explanations in diverse scenarios by focusing on model-specific optimizations. Ustun et al. (2019) presented the Actionable Recourse (ACTREC) method, which addresses the problem of actionability in

counterfactual explanations by constraining generated counterfactuals to ensure that unchangeable features remain unchanged. The method formulates the problem through mixed-integer programming, incorporating constraints on actionable features to maintain valid and actionable solutions. Designed for tabular data and differentiable classifiers, ACTREC also handles categorical features through discretization.

Kanamori et al. (2020) proposed the Distribution-Aware Counterfactual Explanation (DACE) method, which uses mixed-integer linear optimization to generate counterfactual explanations. DACE incorporates a loss function that combines the Mahalanobis distance and the Local Outlier Factor (LOF) to evaluate the possibility of counterfactuals. By minimizing distance while maintaining plausibility, DACE provides explanations for linear classifiers and tree ensembles, using one-hot encoding to handle categorical features. Karimi et al. (2020) introduced the Model-Agnostic Counterfactual Explanation (MACE) approach, which operates on diverse tabular data with any given distance function. MACE maps the problem into a sequence of satisfiability problems, expressing black-box models, distance functions, and constraints as logic formulas. By employing satisfiability modulo theories solvers, MACE generates counterfactual explanations, facilitating interpretable insights into model predictions.

Mothilal et al. (2020) proposed Diverse Counterfactual Explanations (DICE), which solves an optimization problem with various constraints to ensure the feasibility and diversity of generated counterfactuals. The method promotes actionability and feasibility by penalizing solutions that are too similar, thereby encouraging diversity. DICE handles categorical features through one-hot encoding and utilizes the Adam optimizer for efficient computation. Pawelczyk et al. (2020) introduced the Counterfactual Conditional Heterogeneous Autoencoder (C-HVAE), a model-agnostic explainer for tabular data that uses an autoencoder to model heterogeneous data and approximate conditional likelihoods. Unlike other methods, C-HVAE does not require a distance function in the real input space. Instead, it relies on the autoencoder to measure distances in the latent space, guiding the search for counterfactuals.

Parmentier and Vidal (2021) proposed the Optimal Counterfactual ExplAiNer (OCEAN), which focuses on tree ensembles and uses efficient mixed-integer programming to search for counterfactuals. OCEAN accounts for both plausibility and actionability, providing a robust framework for generating optimal counterfactual explanations. Kanamori et al. (2021) introduced the Ordered Counterfactual Explanation (ORDCE) method, which accounts for asymmetric interactions among features by calculating a loss function that depends on the order of feature changes. This method aims to return counterfactuals that not only specify feature values but also the sequence in which features should be altered, enhancing the practical relevance of the explanations.

After overviewing various counterfactual explainability techniques, DICE has been found the most persuasive method for delivering useful insights into the dynamics of decision-making inside Schiphol Airport's operations when conducting research on the TOBT of the A-CDM process. DICE provides a extensive method for hsndling the variables influencing TOBT within the A-CDM framework by utilizing optimization techniques to guarantee the viability and diversity of generated counterfactuals. DiCE is known for actionability settings. This can be setting ranges or adjusting actionable features. This enables successful use of explainability in dashboards because it can generate valuable insights in factors that may be improved to reduce off block time delays. Furthermore, the research community has validated and adopted DICE, demonstrating its efficacy and dependability in obtaining practical insights from intricate operational data. This can be concluded when visiting the GitHub repository of DiCE.

2.6 Dashboard

In the realm of analytical dashboards, several frameworks and tools are available, each offering different advantages. Notable options include Django, Jupyter Notebooks, R Shiny, Flask and Dash. Django is a high-level Python web framework that promotes fast development and clean design. It includes many built-in

features suitable for complex applications. However, its monolithic nature can be inconvenient for projects requiring more flexibility (Holovaty & Kaplan-Moss, 2005). Jupyter Notebooks provide an interactive environment for combining code execution, rich text, and visualizations. They are popular for exploratory data analysis but lack the robustness and scalability for full-scale applications (Kluyver et al., 2016). R Shiny is a web application framework for R, designed for interactive data visualizations and dashboards. It is excellent for R users but less suitable for Python developers and may not integrate well with Python-based tools (Chang et al., 2020).

Flask is a Python framework that simplifies web development with its flexible and simple design. Initially created as an April Fool's Day joke, it gained popularity due to its simplicity and adaptability (Ronacher, 2010). Built on Werkzeug and Jinja2, Flask offers a strong foundation for both small and complex applications. Its modular design allows developers to choose necessary libraries and tools, avoiding unnecessary work. Additionally, Flask is supported by many third-party extensions, providing functionalities such as database integration, form validation, and user authentication (Grinberg, 2017). Flask is used by major companies like Netflix and LinkedIn, showcasing its reliability (Grinberg, 2018). Dash, developed by Plotly in 2017, is ideal for building analytical web applications. Dash is designed to simplify the creation of interactive, data-driven applications, making it accessible to data scientists and analysts who may not have extensive web development experience (Plotly, 2017). Dash allows developers to write the entire application in Python, eliminating the need for proficiency in HTML, CSS, and JavaScript. This feature significantly lowers the barrier to entry for creating web applications with complex data visualizations (Belorkar et al., 2020). Dash performs well in creating dashboards and data visualization applications, offering highly interactive components such as dropdowns, sliders, and graphs that respond dynamically to user inputs. The framework integrates with Plotly's graphing libraries, enabling the creation of high-quality, interactive visualizations. This makes Dash particularly useful in sectors where data interpretation and representation are important. Companies like IBM, NVIDIA, and Tesla have utilized Dash for developing their data visualization interfaces (Dash, 2019).

Combining Flask and Dash leverages the strengths of both frameworks. Flask handles backend infrastructure, such as user authentication, database interactions, and API integrations. On the other hand, Dash provides interactive data visualizations and dashboards. This combination allows developers to use Flask's robust backend capabilities while taking advantage of Dash's powerful data visualization tools (Plotly, 2019). The integration of Flask and Dash can result in a powerful web application framework that helps to both general web development needs and specialized data visualization requirements. This combination not only enhances the functionality of the application but also improves the user experience by offering interactive and visually appealing interfaces (Grinberg, 2018).

For this research, the combination of Flask and Dash offers an ideal solution. Flask's simplicity and flexibility, paired with Dash's powerful visualization capabilities, fit the project's requirements and time constraints. This combination demonstrates the power of counterfactual machine learning effectively and efficiently.

3. Methodology

This chapter aims to give an overview of the research approach used while creating this report. The first section of this chapter explains the approach of the research and its applicability to do so (3.1). The next section explains the origin of the data used in this research (3.2). Thereafter, the following section explains the processes of cleaning the data, enabling it to be analysed (3.3). Afterwards, the next section explains how the resulting data is analysed (3.4), using various techniques. Then, the creation of the dashboard is explained (3.5). This is followed by a justification of the methods that are used during the research (3.6). Then, the next section of this chapter explains the reliability and validity of this research (3.7). This chapter is ended with a section of ethical considerations of this research (3.8).

3.1 Variable selection

This research investigates the factors that leads to inconsistency between the TOBT and AOBT. This under-researched topic is of interest for LVNL since deviations of this can lead to planning disruptions for air traffic controllers, eventually leading to capacity loss at one of busiest airports of Europe (Schiphol Group, 2023). The TOBT is an important term within the A-CDM concept and is used to calculate other terms within A-CDM, such as the target take-off time (TTOT) and the target start-up approval time (TSAT). It is of interest to the company to understand how independent variables are influencing the variations between TOBT and AOBT. For this reason, independent variables are researched statistically. Furthermore, prediction models are created and evaluated to be able to predict the unalignments between TOBT and AOBT. To get a greater understanding of factors influencing the misalignment, counterfactual explainability is utilized to investigate the models' feature importance. The following independent variables are researched (table 4).

Independent variable	Explanation	Type of variable
Bax Events	Duration of baggage handling events	Continuous numeric
Catering Events	Duration of catering handling events	Continuous numeric
Line Maintenance Events	Duration of maintenance handling events	Continuous numeric
Water or Toilet Events	Duration of lavatory handling events	Continuous numeric
Pushback Events	Duration of pushback handling events	Continuous numeric
Fuel Events	Duration of fuelling handling events	Continuous numeric
Pax Events	Duration of passenger handling events	Continuous numeric
Main Handler	Responsible for main handling during turnaround	Categorical
Apron Handler	Responsible for apron handling during turnaround	Categorical
Baggage Handler	Responsible for baggage handling during turnaround	Categorical
Clean Handler	Responsible for clean handling during turnaround	Categorical
Food Handler	Responsible for catering handling during turnaround	Categorical
Freight Handler	Responsible for freight handling during turnaround	Categorical
Fuel Handler	Responsible for fuel handling during turnaround	Categorical
Pax Handler	Responsible for passenger handling during turnaround	Categorical
Tec Handler	Responsible for technical handling during turnaround	Categorical
Day	Monday, Tuesday, Wednesday, etc.	Categorical
Month	November, December, January, or February	Categorical
Hour	The hour the turnaround was completed (10:15 = 10)	Categorical
Alliance	Airline's alliance (Star Alliance, Sky Team, Oneworld)	Categorical
Carrier Type	Full-service carrier, low-cost carrier, charter	Categorical
Capacity	Number of turnarounds completed in the hour	Discrete numeric

Table 4: Independent variables

3.2 Data collection

To be able to do research, data is required to investigate feature influence, create a machine learning model, and research feature importance of the model by counterfactual explainability. These research techniques require merely quantitative data. For this research, multiple data sources are needed. First,

historic data of the turnaround processes is needed. This data is received from the DeepTurn IT department of Schiphol. As more in depth explained in chapter 2, DeepTurn is an AI image recognition technique using cameras that was introduced at Schiphol in 2021 (Schiphol Airport, n.d). The department of Schiphol shared two types of data, CDM-data, and event-data. The CDM data consist of 11 PARQUET files and included information about the aircraft that were captured by the DeepTurn cameras during their turnaround. The event-data consists of 579 PARQUET files which includes of information of every of the 72 specified events during the turnaround captured by the DeepTurn cameras. These events are documented with timestamps of when they occurred. All these files consist of a unique ID which can be used to identify events that are part of the same turnaround and enables the two different types of data to be linked together.

Although the DeepTurn data supplies a lot of valuable features, a key variable is missing, TOBT. Since this variable is mandatory for this research, historic CDM data from LVNL, which includes TOBT, is gathered. Since this database uses a different key to identify separate turnarounds, the TOBT information could not directly be merged to the data provided by Schiphol. Therefore, outbound flight data is used, which serves as intermediate data because of its corresponding features. This data is also gathered from the databases of LVNL.

3.3 Data cleaning

Since the gathered data originates from different data managements systems, the first step of cleaning consists of creating equivalent data structures. This required pivoting the events data so that every turnaround had its own datapoint, with each of the events as features including a value of time that had passed to complete that event in seconds. Since the data from Schiphol consists of event and CDM data with the same keys, this data can be merged. The same applies to LVNL's CDM-data and outbound flight data. Due to the lack of matching keys in LVNL's and Schiphol's data, the resulting data frames are merged by using unique aircraft registrations within a specific day. If an aircraft registration occurs multiple times per day, the AOBT of each data source is compared and the closest datapoint is kept. The AOBT in both data sources usual differs a couple minutes.

For this research, important independent variables to be studied are the handler companies. These variables are important because LVNL is interested in gaining insight into them. Handler information was not included in the data until 18 November 2023. Therefore, all turnarounds prior to this date are excluded from the data. Furthermore, due to the different turnaround nature of cargo and passenger flights, all cargo turnarounds are dropped. Finally, all long-haul flights are also removed from the data. These flights are also of a different nature to short-haul flights. They are often dependent on short-haul flights due to the hub and spoke network of Schiphol Airport and therefore often have delays that cannot be explained by these data.

As the 72 turnaround events are too large to be used for a machine learning model and often not all sub-events are captured in the data, the events are filtered into events as shown in Table 6. The events are grouped based on the DeepTurn event documentation (Schiphol DeepTurn department, 2023). This filtering was necessary because DeepTurn's event data does not actually record events and their durations, but only events that occur at a date and time. This often resulted in noise in the data. A sub-event captured by the DeepTurn cameras often did not necessarily lead to that event occurring. For example, certain vehicles (such as de-icing vehicles, fuel tankers or catering vehicles) would often appear and be recorded by the cameras even though they did not cause the event for which they were intended. This led to the following definitions created to document the events shown in Table 6. Each event is calculated using the first completed sub-event and the last completed sub-event. In this way the duration of the event is retained in the data. This duration is then expressed as a percentage of the total turnaround time.

Grouped events	Sub events
Fuel events	First fuel truck stops in position; Last fuel truck moves out of position
Pushback event	Tug idle connected starts; Aircraft moves out of position
Bax events	First belt loader stops in position; Last belt loader moves out of position
De-ice events	De-icing wing starts, stops;
Line maintenance events	First/Last oil check truck appears, disappears; Power connects, disconnects
Water or toilet events	First/Last water or toilet truck appears, disappears
Catering events	First catering truck completes ascent, completes descent,
Pax events	First pax bridge connects, last pax bridge stops in park position; First ambulift appears, last ambulift disappears; Rear/front pax stairs disconnects, connects; Ambulift in/out of position; Pigs connected, disconnected, First/Last pax door open, closed; Pax disembark/board starts, stops; First/Last pax door front left/right/rear left/rear right open, closed

Table 5: Grouped DeepTurn events

The final step in the data cleaning process is to create new features. Firstly, time of day is used as a feature. Secondly, some categorical features are made up of values that appear too rarely in the data. Airlines are grouped into alliances as they often share handlers. They are also grouped as either low-cost carriers, full-service carriers, or charter flights. Finally, the target variable is defined and constructed. Since this research aims to investigate the difference between AOBT and TOBT, a target variable is created called Delta Off-Block Time (DOBT), which is the difference between the two. The TOBT used in this research is the one that is defined as soon as the turnarounds begin.

3.4 Data analysis

The data analysis in this research consists of three parts: a statistical analysis (3.4.1), machine learning algorithms (3.4.2), and a counterfactual machine learning explainability analysis (3.4.3).

3.4.1 Statistical tests

The statistical analysis investigates the correlation between independent variables and the dependent variable of this research, which is DOBT. The statistical analysis approach differentiates between the categorical and the numerical variables.

To analyse the categorical variables, the study employs the ANalysis OF VAriance (ANOVA) test and the Kruskal-Wallis test, depending on whether the assumptions of ANOVA are met. The ANOVA test is used to determine if there are any statistically significant differences between the means of three or more independent groups. It assumes that the data is normally distributed and that variances are homogeneous across groups (Schmider et al., 2010). The normality of each feature's distribution is checked visually using Q-Q plots and histogram plots of the residuals, and statistically by conducting the Shapiro-Wilk test. The Shapiro-Wilk test assesses whether the data is drawn from a normally distributed population, with a p-value less than 0.05 indicating a deviation from normality (Razali & Wah, 2011).

Homogeneity of variance is evaluated using Bartlett's test and Levene's test. Bartlett's test is sensitive to departures from normality, while Levene's test is more robust for non-normal distributions (Lim & Loh, 1996). Both tests assess whether variances are equal across groups, with a significant p-value indicating unequal variances. If the assumptions of normality and homogeneity of variance are met, ANOVA is performed to compare the group means using the F-statistic, which measures the ratio of variance between the groups to the variance within the groups (Feir-Walsh & Toothaker, 1974).

When the assumptions of ANOVA are not met, the Kruskal-Wallis test, a non-parametric alternative, is used. The Kruskal-Wallis test does not assume normal distribution and is used to determine if there are statistically significant differences between the medians of three or more independent groups (McKight & Najab, 2010). This test ranks all the data points and evaluates whether the ranks differ significantly between groups, providing a method for analysing non-normally distributed data.

For numerical variables, the normality of each feature's distribution is assessed both visually, using Q-Q plots and histogram plots of the residuals, and statistically, using the Shapiro-Wilk test (Ghasemi & Zahediasl, 2012). Variance homogeneity is then evaluated visually using residual vs. fitted values plots and statistically using the Breusch-Pagan test, which tests for heteroscedasticity (Breusch & Pagan, 1979). A significant Breusch-Pagan test indicates heteroscedasticity, or non-constant variance, in the data.

Based on the outcomes of these tests, either Pearson or Spearman correlation coefficients are calculated to measure the strength and direction of the relationship between numerical features and the dependent variable, DOBT. Pearson correlation is used for normally distributed data and measures linear relationships, while Spearman correlation is used for non-normally distributed data and measures monotonic relationships (Hauke & Kossowski, 2011).

Finally, linear regression analysis is conducted to model the relationship between the dependent variable and one or more independent variables. Linear regression assumes linearity, independence, homoscedasticity, and normally distributed residuals. The regression equation models the dependent variable as a linear combination of the independent variables plus an error term (Montgomery et al., 2012). This analysis provides insights into the extent to which each independent variable predicts the dependent variable.

3.4.2 Machine learning algorithms

In this research, machine learning models were developed and optimized using pipelines to integrate data preprocessing steps and model training efficiently. This approach ensures that functions such as standard scaling and one-hot encoding are consistently applied across all models while preserving the attributes needed for counterfactual explainability analysis. This section overviews feature reduction techniques (3.4.2.1), machine learning models and the hyperparameters that are tuned (3.4.2.2), performance metrics that are used (3.4.2.3) and model feature importance (3.4.2.4)

3.4.2.1 Feature reduction techniques

In this research, two feature reduction techniques are used, Variance Inflation Factor (VIF) and Principal Component Analysis (PCA). VIF is a measure used to detect multicollinearity among the independent variables in a regression model. High VIF values indicate high correlation between variables, which can inflate the variance of regression coefficients and make them unstable (O'Brien, 2007). The VIF is calculated using the formula (equation 1):

$$VIF_i = \frac{1}{1 - R_i^2} \qquad \text{Eq. 1}$$

In this, R_i^2 is the coefficient of determination of the regression of the i-th independent variable on all the other independent variables. A VIF value greater than 10 is often used as a rule of thumb to indicate significant multicollinearity, so this is also the case for this research (Kutner et al., 2004). In this study, dummy variables were created for categorical columns, and features with VIF values greater than 10 were excluded from the model to reduce multicollinearity.

PCA is a dimensionality reduction technique that transforms a large set of variables into a smaller set of uncorrelated variables called principal components. These components capture the maximum variance in the data. The steps involved in PCA include:

1. Standardization**:** Scaling the data so that each feature has a mean of zero and a standard deviation of one (Jolliffe, 2002).
2. Covariance Matrix Computation**:** Calculating the covariance matrix to understand the variance and covariance between different features (Jolliffe, 2002).

3. Eigen Decomposition**:** Computing eigenvalues and eigenvectors from the covariance matrix to identify the principal components (Jolliffe, 2002).
4. Projection**:** Projecting the data onto the principal components to reduce dimensionality (Jolliffe, 2002).

In this research, PCA was used to retain features that accounted for 95% of the variance. This threshold ensured that the most informative features were kept, avoiding overfitting while maintaining model interpretability. The results are found in figure, 2. Due to little variation in explainability between components, no feature was reduced for this research.

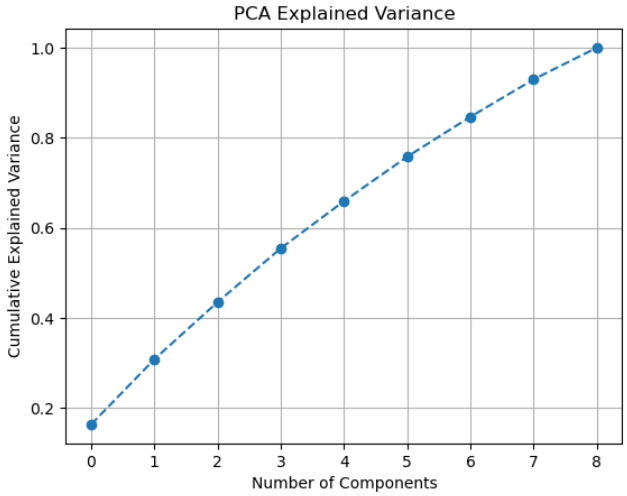


Figure 2: PCA results

3.4.2.2 Machine learning models and hyperparameters

Based on previous research, three machine learning models are created. These are Random Forest, LightGBM and XGBoosting. Furthermore, a base model is created using logistic regression to have a comparable model. The models are created by splitting test and train data and doing hyperparameter tuning for the predictors. To be able to work with counterfactuals, the target variable of DOBT is transformed in a binary classification. For this, a threshold is set at 300 seconds, meaning a DOBT of less than 300 seconds is the desired class and a DOBT that exceeds this threshold is considered undesired. After creating the four models, a fifth model is created which is an ensembled model. Its operation and the hyperparameters tuned for this study are described below.

Logistic Regression is a linear model used for binary classification problems. It estimates the probability that a given input belongs to a certain class (Hosmer et al., 2013). The logistic function is defined as (equation 2):

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n)}} \qquad \text{Eq. 2}$$

Hyperparameters tuned include:

- C**:** The inverse of regularization strength, smaller values specify stronger regularization, which helps prevent overfitting by penalizing large coefficients (Hosmer et al., 2013).
- Solver: The algorithm used in the optimization problem. 'liblinear' is suitable for small datasets or binary classification (Peduzzi et al., 1996).

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. For this research, it's designed to solve a classification problem. The general principle uses Bootstrap aggregation (bagging)**,** random subsets of the training set are sampled with replacement to train each tree, and random feature selection, each node in a tree considers a random subset of features to split on, reducing correlation between trees and improving model robustness (Liaw & Wiener, 2002). Key hyperparameters tuned in this research are:

- bootstrap: Whether bootstrap samples are used when building trees, which helps in reducing variance (Breiman, 2001).
- max_depth: Maximum depth of the tree; controls overfitting by limiting the number of splits in each tree (Liaw & Wiener, 2002).
- min_samples_split: Minimum number of samples required to split an internal node, helping to prevent overfitting (Liaw & Wiener, 2002).
- min_samples_leaf: Minimum number of samples required to be at a leaf node, ensuring each leaf has enough samples (Liaw & Wiener, 2002).
- n_estimators: Number of trees in the forest, with more trees generally leading to better performance but increased computational cost (Liaw & Wiener, 2002).

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. Gradient boosting involves model building, while adding predictors, each correcting its predecessor's errors, and optimization, minimizing a differentiable loss function using gradient descent (Ke et al., 2017). Key hyperparameters include:

- max_depth: Maximum depth of the tree, controlling the complexity of the model (Ke et al., 2017).
- min_child_samples: Minimum number of data points required in a child (leaf) node, which helps in controlling overfitting (Ke et al., 2017).
- n_estimators: Number of boosting iterations, balancing bias and variance (Ke et al., 2017).
- num_leaves: Maximum number of leaves in one tree, balancing model complexity and performance (Ke et al., 2017).

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. Key principles include regularized learning objective, incorporates regularization to prevent overfitting, shrinking, reduces the influence of each tree to allow subsequent trees to correct errors more effectively, and column sampling**, i**mproves computation speed and model robustness (Chen & Guestrin, 2016). Key hyperparameters include:

- colsample_bytree: Fraction of features to be used by each tree, which helps in preventing overfitting (Chen & Guestrin, 2016).
- learning_rate: Step size shrinkage used to prevent overfitting by lowering the influence of each individual tree (Chen & Guestrin, 2016).
- max_depth: Maximum depth of a tree, controlling the model's complexity (Chen & Guestrin, 2016).
- n_estimators: Number of trees, balancing bias and variance (Chen & Guestrin, 2016).
- subsample: Fraction of samples used for fitting individual base learners, preventing overfitting (Chen & Guestrin, 2016).

An ensemble model integrates multiple models to improve performance. In this research, the ensemble method used voting to combine the predictions of Random Forest, LightGBM, XGBoost, and Logistic

Regression classifiers. This approach leverages the strengths of each algorithm to achieve a more balanced and robust predictive capability (Sagi & Rokach, 2018).

3.4.2.3 Performance metrics

Performance metrics are essential for evaluating the effectiveness of the machine learning models. It enables to compare models and draw conclusion based on the reliability of the models. The following metrics were used:

- Accuracy: The proportion of true results (both true positives and true negatives) among the total number of cases. It is a measure of the overall correctness of the model (Sokolova & Lapalme, 2009).
- Precision: The ratio of true positives to the sum of true positives and false positives. It indicates the accuracy of the positive predictions and is crucial when the cost of false positives is high (Powers, 2011).
- Recall: The ratio of true positives to the sum of true positives and false negatives. It measures the ability of the model to identify all relevant instances and is important when the cost of false negatives is high (Sokolova & Lapalme, 2009).
- F1 Score: The harmonic mean of precision and recall. It provides a single metric that balances both concerns, especially useful when the class distribution is imbalanced (Powers, 2011).
- AUC (Area Under the ROC Curve): The area under the Receiver Operating Characteristic curve, which measures the model's ability to distinguish between classes. Higher values indicate better performance, as it considers the trade-off between true positive rate and false positive rate (Fawcett, 2006).

These selected metrics offer balanced evaluation of model performance, addressing different aspects such as overall accuracy, precision, recall, balance in imbalanced datasets, and discriminative ability across thresholds. This combination ensures that the models are fully assessed and that the results are reliable and interpretable for various use cases.

3.4.2.4 Model feature importance

Feature importance is a critical aspect of interpreting machine learning models, especially in understanding which features most significantly impact predictions. For logistic regression, feature importance can be determined by examining the magnitude of the coefficients assigned to each feature, with larger absolute values indicating greater importance. For tree-based models like Random Forest, LightGBM, and XGBoost, feature importance is determined by how much each feature contributes to reducing the impurity (Gini impurity or entropy) in the trees (Breiman, 2001; Ke et al., 2017; Chen & Guestrin, 2016). This information is valuable for identifying the key factors influencing TOBT predictions. By analyzing feature importance, stakeholders can gain insights into which variables are most impactful, enabling more informed decision-making and optimization of operations at Schiphol Airport.

3.4.3 Counterfactuals

Furthermore, the data is analysed using counterfactual machine learning. Based on the outcomes of the literature review, the DiCE ML methodology is used for this analysis. Counterfactual explanations are hypothetical scenarios that alter certain features of an instance to achieve a desired outcome while keeping other features constant. This approach provides insights into how changes in specific features can impact the prediction of the model.

Counterfactual explanations aim to answer "what if" questions, providing examples that show how a different outcome could be achieved by changing certain features of the input data. For instance, in the context of this research, a counterfactual explanation might suggest that a delay could be avoided if the

refuelling events were reduced by a certain percentage. The process involves generating counterfactuals, creating alternative scenarios where the model's prediction changes. The DiCE ML framework generates multiple counterfactual examples to offer diverse explanations. Each counterfactual instance is generated by changing one or more features of the original instance (Mothilal et al., 2020). To prevent unrealistic results, constraints are applied. In this research, numerical features are allowed to change only within specified ranges. Specifically, event times can decrease by a maximum of 25 percent to ensure the generated counterfactuals are plausible and actionable.

Local feature importance refers to the impact of features on the prediction of a single instance. By analysing counterfactuals for specific instances, we can determine which features need to change to chnage the prediction. This helps in understanding the influence of each feature on individual predictions and provides actionable insights for specific scenarios (Mothilal et al., 2020). The local feature importance of a single datapoint is integrated in the dashboard created during this research.

Global feature importance assesses the impact of features across all data points in the dataset. This is achieved by aggregating the feature importance from all counterfactual instances to identify the most influential features globally. This is done by generating ten counterfactual explanations for each datapoint in the train data. In this study, the results of the global feature importance are plotted in histograms. These histograms show the distribution of feature importance scores, enabling comparisons between different models (Mothilal et al., 2020).

3.5 Dashboard

The results of the counterfactual instances are implemented in a dashboard using a combination of Flask and Dash. Flask is used to create a backend API for handling predictions and generating counterfactuals, while Dash is used to create an interactive dashboard for users to input data and visualize results.

Flask is a lightweight web framework for Python, suitable for creating APIs and handling web requests (Grinberg, 2018). In this project, Flask is used to serve a machine learning model and handle prediction and counterfactual generation requests. The Flask application includes multiple API endpoints. The predict endpoint accepts a JSON input, preprocesses it, and returns a prediction. The counterfactuals endpoint accepts a JSON input, preprocesses it, and returns counterfactual explanations. The expected-types endpoint returns the expected data types for input features. This is important to correctly generate counterfactual explanations. Dice ML is used to generate counterfactual explanations, resulting in alternative scenarios that could lead to a different prediction (Mothilal, Sharma, & Tan, 2020). This is particularly useful for understanding model decisions and exploring how the smallest changes in input features affect the output. This means that for this dashboard, the ability is created to investigate counterfactual explicabilities for delayed individual delayed turnarounds.

Dash is a framework for building analytical web applications using Python (Plotly, 2021). It allows for the creation of interactive, web-based dashboards. The layout of the Dash application is designed to be user-friendly and intuitive, focusing on simplicity. It includes a dropdown menu for selecting flight IDs, input field for entering turnaround time, and a button for triggering predictions and counterfactual generation. Additionally, it features tables for displaying original data points and counterfactuals explanations. When a user selects a flight ID and enters a turnaround time, the data is sent to the Flask API for prediction and counterfactual generation. The results are then displayed in the dashboard. Dash uses callbacks to update components dynamically based on user interactions. This ensures that the dashboard remains responsive and interactive. Numerical features in the data are converted from percentages to actual times based on the entered turnaround time. This transformation helps in understanding the impact of different events on the turnaround time more clearly.

Integrating Flask and Dash allows for a powerful combination where Flask handles the backend logic and Dash provides a user-friendly frontend. User inputs in the Dash frontend are sent as JSON requests to the Flask API. Flask processes these inputs, performs predictions, or generates counterfactuals, and returns the results. Dash receives the results and updates the dashboard components. Both Flask and Dash are run as separate servers. Flask serves the API endpoints, while Dash runs the interactive dashboard. This separation ensures that the backend logic is sperate from the frontend, making the application modular and easier to maintain. The integration also allows for efficient handling of machine learning tasks and the presentation of results in an accessible manner. By using Dash, the results of the predictions and counterfactuals can be easily interpreted by users without requiring deep technical knowledge.

3.6 Reliability and validity

For this research on the factors influencing TOBT deviations at Schiphol Airport, several measures have been taken to ensure reliability. Firstly, the data collection process was consistent and used reliable sources, such as DeepTurn data from Schiphol Airport and historic A-CDM data from LVNL. Consistency in data collection methods ensures that the data is dependable and replicable (Lewis-Beck et al., 2003).

Secondly, the data cleaning and processing steps were standardized to handle raw data from different sources. This process included merging data based on unique identifiers, filtering out irrelevant data, and creating equivalent data structures. By applying consistent data cleaning techniques, the research minimizes the chances of introducing biases or errors, thereby enhancing the reliability of the results (Van den Broeck et al., 2005).

Thirdly, established statistical methods, such as ANOVA, Kruskal-Wallis, Shapiro-Wilk, Bartlett's, and Levene's tests, were used. These tests were chosen based on the nature of the data and the research objectives, ensuring that the results are robust and reproducible (Field, 2018). Additionally, to ensure the validity of these statistical tests, assumptions were rigorously tested. For example, normality assumptions for parametric tests were assessed using the Shapiro-Wilk test, and homogeneity of variances was evaluated using Levene's and Bartlett's tests (Bhaumik & Dey, 2022). Where assumptions were violated, appropriate non-parametric tests, such as the Kruskal-Wallis test, were employed to ensure the robustness of the findings.

Fourthly, the machine learning models used in this research, including Random Forest, LightGBM, XGBoost, and logistic regression, are well-documented and widely used in similar research contexts. The use of these models, combined with hyperparameter tuning and cross-validation, ensures that the predictions are reliable and generalizable. Model assumptions were tested using feature importance plots and VIF for multicollinearity to confirm that the models are well-specified (Bhaumik & Dey, 2022). The ensemble model further enhances reliability by leveraging the strengths of individual models (Breiman, 2001; Friedman, 2001; Chen & Guestrin, 2016; Ke et al., 2017).

Lastly, the DiCE methodology was selected for its ability to produce diverse and actionable insights. By setting constraints on counterfactuals, such as allowing event times to decrease by a range, the research ensures that the generated explanations are realistic and relevant, thereby increasing the reliability of the insights derived from the models (Mothilal et al., 2020).

This research addresses several aspects of validity to ensure that the findings are credible and applicable. Internal validity is achieved by carefully designing the research methodology to eliminate misleading variables and biases. The selection of independent variables was based on a literature review and expert input, ensuring that all relevant factors influencing TOBT are considered. Additionally, the use of appropriate statistical tests and machine learning models enhances the internal validity of the research (Shadish et al., 2002).

External validity concerns the generalizability of the findings. By using data from a major international airport (Schiphol Airport) and using widely recognized methodologies, the research findings are likely to be applicable to other similar airport environments. The fact that plenty of other airports also implemented A-CDM supports this claim. The generalizability is further supported by the inclusion of a diverse set of variables, covering different aspects of the turnaround process (Steckler & McLeroy, 2008).

Construct validity is ensured by accurately defining and measuring the constructs of interest, such as TOBT, AOBT, and the various independent variables. The use of well-defined metrics and consistent data collection methods ensures that the constructs are accurately represented in the data. The transformation of the target variable (DOBT) into a binary classification is based on a clear threshold, further supporting construct validity (Trochim & Donnelly, 2008).

Content validity is maintained through a comprehensive literature review and expert consultations, ensuring that the research covers all relevant aspects of TOBT estimation and deviations. The selection of independent variables is based on previous research and industry practices, using all features of interest that are influencing TOBT (Haynes et al., 1995).

Face validity is achieved by presenting the research methodology and findings to stakeholders and experts in the field. The involvement of LVNL and the use of DeepTurn data provide practical relevance and credibility to the research. Validation through expert feedback ensures that the results are found to be valid and relevant by people working in the field of aviation (Nunnally & Bernstein, 1994).

3.7 Ethical considerations

The research relies on historical data from Schiphol Airport and LVNL. Ensuring the privacy and confidentiality of this data is of great importance. All data used was anonymous to prevent the identification of individuals involved in the data set. The data was presented using protected and secured by being sent through a Windows Azure Private Link. For the Schiphol data, no Non-Disclosure Agreement (NDA) was required due to the agreements between LVNL and Schiphol Group of shared data.

Although the research did not involve direct interaction with individuals, informed consent principles were applied to the use of data. This included clear communication about the objectives of the research, how the data would be used, and the measures in place to protect the data (Smith, 2020). This involved weekly meetings with stakeholders wherein progression and data handling were discussed.

Detailed documentation of all processes, from data collection and cleaning to model development and validation, was maintained. This documentation ensures that the research can be audited and replicated by other researchers, enhancing the reliability and credibility of the findings. This is achieved by creating an in-depth methodology, but also by documenting the programming code extensively, so that it could easily be replicated. Additionally, any potential conflicts of interest were disclosed, and the research was conducted independently without undue influence from external stakeholders (Jones, 2019).

The deployment of machine learning models raises ethical questions about bias, fairness, and the impact of decisions based on model predictions. In this research, efforts were made to ensure that the models used were as unbiased and fair as possible, by testing for biases, balancing different model performance metrics and implementing techniques to mitigate any identified biases, such as VIF. Furthermore, the explainability of the models was prioritizedto facilitate the understanding and trust of users (Molnar, 2020).

This study, focusing on the optimization of airport operations, has potential environmental and social benefits, such as reducing delays and improving operational efficiency, which can lead to lower emissions and better service for passengers. The research was designed to contribute positively to these areas, aligning with broader societal goals of sustainability and improved quality of life (United Nations, 2015).

4. Results

In this chapter, the results of this research are presented. These results are a direct result of the described objectives, built on the analysed prior done research and created based on the explained methodology. The results are divided over four sections. In the first section, the statistical results are shown between the different independent variables and the target variable, DOBT (4.1). Thereafter, the prediction model outcomes are explained (4.2). Furthermore, the counterfactual machine learning explainability results are shown (4.3). To conclude, the last section involves the outcomes of the dashboard (4.4)

4.1 Statistical analysis

In the statistical analysis section, all results considering the correlation between the individual independent variables and DOBT are overviewed. Since the nature of researching different kind of variables are different, as also explained in the methodology, this requires different tests. Therefore, there is a separate section for continuous numeric features (4.1.1) and for the remaining features (4.1.2).

4.1.1 Numeric features

Table 6 presents the results of various statistical analyses on different event features to assess normality, heteroscedasticity, correlation, and regression significance. These analyses provide insights into the data characteristics and relationships between features and DOBT. The individual distributions of the numerical features are found in Appendix 3 & 4.

Feature	**S-W p-value**	**Normality**	**B-P p-value**	**Heterosce dasticity**	**Correlation Coeff**	**Spearman p-value**	**Correlation Significance**
Bax Events	2.1811e-39	✗	1.9882e-01	✗	4.2104e-02	1.1360e-06	✓
Catering Events	5.2198e-32	✗	8.8339e-03	✓	-4.6233e-02	3.9686e-05	✓
Line Maintenance Events	5.9686e-40	✗	2.4774e-05	✓	6.8860e-02	1.3265e-16	✓
Water or Toilet Events	5.2063e-24	✗	6.6336e-01	✗	-2.3249e-02	1.1121e-01	✗
Pushback Events	9.9754e-39	✗	1.5172e-24	✓	2.0734e-01	9.135e-140	✓
Fuel Events	3.4332e-39	✗	1.1697e-01	✗	5.1864e-02	2.4386e-09	✓
Pax Events	1.2595e-40	✗	2.7737e-04	✓	6.5336e-02	2.9259e-16	✓
Feature	**LR Intercept**	**LR Coeff**	**R-squared**	**MSE**	**F-statistic**	**Regression Significance**	
Bax Events	2.4972e+02	1.168e-02	1.8159e-03	6.585e+04	2.4286e+01	✓	
Catering Events	2.9754e+02	-4.59e-02	2.1454e-03	6.633e+04	1.6972e+01	✓	
Line Maintenance Events	2.3969e+02	7.497e-03	2.8667e-03	6.559e+04	4.1387e+01	✓	
Water or Toilet Events	2.7942e+02	-4.22e-02	5.6572e-04	6.460e+04	2.6576e+00	✗	
Pushback Events	2.4914e+02	3.486e-02	9.6047e-03	6.516e+04	1.3981e+02	✓	
Fuel Events	2.3382e+02	2.855e-02	1.9802e-03	6.535e+04	2.6220e+01	✓	
Pax Events	2.4509e+02	6.096e-03	1.7688e-03	6.567e+04	2.7696e+01	✓	

Table 6: Correlation and linear regression results between independent numeric variables and the dependent variable

The Shapiro-Wilk test indicates that none of the features follow a normal distribution, suggesting the need for non-parametric tests or data transformations for further analysis. Normality was also checked visually. Heteroscedasticity was assessed using the Breusch-Pagan test. Features like Catering Events, Line Maintenance Events, Pushback Events, and Pax Events showed heteroscedasticity, while Bax Events, Water or Toilet Events, and Fuel Events show homoscedasticity. Heteroscedasticity implies varying variance in the dependent variable across values, affecting regression validity. Visual inspection of residual plots for heteroscedasticity is provided in Appendix 5. Spearman's rank correlation coefficient was used to evaluate the correlation between event occurrences and DOBT. Most features show significant but weak correlations, except Water or Toilet Events, which does not show a significant correlation.

Linear regression analysis explored the relationship between event occurrences and DOBT. Despite some violations of assumptions, using logistic regression can still provide valuable insights into the relationships between predictors and the binary outcome. The significant p-values and F-statistics suggest that the model identifies important predictors. While most features resulted in statistically significant models, the low R-squared values indicate that the models explain only a small fraction of the variance, suggesting the presence of additional influencing factors. An exception is Water or Toilet Events, which does not significantly explain the variance. Residuals versus fitted values and linear regression plots are found in Appendices 6 and 7.

In summary, none of the event features follow a normal distribution, several features exhibit heteroscedasticity, and most show weak but significant correlations with DOBT. The regression models, although statistically significant, have low explanatory power, indicating the need to consider additional factors.

4.1.2 Categorical features

Table 7 presents the results of several statistical tests conducted on various event features to assess normality, homogeneity of variances, and overall group differences. The Shapiro-Wilk test was performed to evaluate the normality of the data in each column. The results show that all columns have p-values significantly less than 0.05, indicating that none of the columns follow a normal distribution.

Feature	S-W p-value	Normality residuals	Bartlett p-value	Bartlett Result	Levene p-value	Levene Result	K-W p-value	K-W Result
Main handler	2.28e-10	✗	3.12e-03	✗	1.62e-04	✗	4.47e-57	✓
Apron handler	6.69e-10	✗	4.09e-03	✗	3.67e-04	✗	7.82e-55	✓
Baggage handler	6.58e-09	✗	3.08e-03	✗	1.62e-04	✗	3.67e-57	✓
Clean handler	1.27e-10	✗	7.21e-02	✓	9.55e-02	✓	5.14e-15	✓
Food handler	1.36e-10	✗	5.27e-01	✓	5.28e-01	✓	9.06e-24	✓
Freight handler	3.19e-10	✗	1.33e-01	✓	1.32e-02	✗	4.20e-54	✓
Fuel handler	7.21e-09	✗	5.31e-01	✓	1.45e-01	✓	1.11e-02	✓
Pax handler	2.57e-08	✗	3.12e-03	✗	1.62e-04	✗	4.47e-57	✓
Technical handler	4.96e-10	✗	1.61e-02	✗	3.72e-04	✗	6.71e-43	✓
Day	5.69e-09	✗	3.68e-06	✗	2.44e-05	✗	3.53e-16	✓
Month	1.31e-09	✗	4.37e-09	✗	3.52e-08	✗	8.85e-12	✓
Alliance	6.57e-10	✗	1.56e-01	✓	2.91e-02	✗	7.74e-14	✓
Carrier type	3.73e-10	✗	2.88e-0	✓	8.39e-01	✓	8.19e-20	✓
Hour	1.39e-06	✗	2.85e-12	✗	1.66e-13	✗	3.15e-42	✓
Capacity	3.13e-10	✗	1.49e-02	✗	2.97e-02	✗	3.86e-03	✓

Table 7: Statistical results of the categorical independent variables with the dependent variable

Two tests were employed to assess the homogeneity of variances across groups: Bartlett's test and Levene's test. The Bartlett's test p-values are significantly less than 0.05 for most features, indicating that the variances are not homogeneous across groups. However, features like Clean Handler, Food Handler, Freight Handler, Fuel Handler, Alliance, and Carrier Type show p-values greater than 0.05, suggesting homogeneity of variances. Levene's test results show that all columns, except Fuel Handler and Clean Handler, have p-values less than 0.05, indicating unequal variances across groups for these features. Homogeneity was also visually checked using boxplots of the top 10 most occurring values per feature, found in Appendix 8.

The results from the Shapiro-Wilk test clearly indicate that none of the columns meet the assumption of normality, as further shown in the residual plots in Appendix 9. Given these violations of assumptions for

parametric tests, the use of the non-parametric Kruskal-Wallis test is justified. The Kruskal-Wallis test results indicate significant differences between the groups for all columns analysed.

The p-values for all features, based on the Kruskal-Wallis test, are significantly less than 0.05. This indicates that there are significant differences between the groups for each feature. The null hypothesis, stating that the medians of all groups are equal, can be rejected for each column. The small p-values suggest strong evidence against the null hypothesis, implying that at least one group median is significantly different from the others. These findings reveal statistically significant differences in the distributions of the groups being compared, highlighting variability in the data due to actual differences between the groups rather than random chance.

In summary, none of the event features follow a normal distribution, several features include heteroscedasticity, and the Kruskal-Wallis test confirms significant differences between groups for all analysed features.

4.2 Prediction models

The logistic regression model, optimized with hyperparameter tuning, resulted in moderate performance in predicting delays. The optimal parameters were determined C=0.01 and the solver set to 'liblinear', balancing the bias-variance trade-off, and ensuring computational efficiency. For the Random Forest model, a comprehensive hyperparameter tuning process identified the best configuration: bootstrap set to True, no restriction on maximum depth, minimum samples per leaf set to 4, minimum samples per split set to 10, and 300 estimators. This setup aims to balance bias and variance for optimal prediction accuracy.

The LightGBM model also followed hyperparameter tuning and optimal hyperparameters included a maximum depth of 10, a minimum of 30 child samples, 300 estimators, and 31 leaves. The XGBoost model was also optimized, with the best hyperparameters identified as follows: colsample_bytree set to 1.0, learning rate set to 0.3, max depth set to 3, number of estimators set to 300, and subsample set to 1.0. An ensemble model integrating Random Forest, LightGBM and XGBoost classifiers was constructed using voting. This ensemble approach uses the strengths of each algorithm, achieving a more balanced and robust predictive capability than any single model. In table 8, the performance of the various machine learning models is overviewed based on the metrics that are discussed in the methodology. These results are emerged from testing the tuned models on the test set of the data. This is 20 percent of the dataset, which is equal to 3250 datapoints, and was split from the train data

Model	Accuracy	Precision	Recall	F1 Score	AUC
Logistic Regression	0.6218	0.5550	0.5881	0.5710	0.66
Random Forest	0.6388	0.5737	0.6075	0.5901	0.69
LightGBM	0.6526	0.5889	0.6240	0.6059	0.70
XGBoosting	0.6588	0.6179	0.5313	0.5713	0.69
Ensembled model	0.6575	0.6030	0.5852	0.5939	0.71

Table 8: Prediction model performance

The confusion matrices for the prediction models can be found in Appendix 10, providing a visual representation of their performance in terms of true positives, false positives, true negatives, and false negatives. The Random Forest and LightGBM models show balanced performance with a good mix of true positives and true negatives. XGBoost resulted in a higher number of true negatives but struggles with true positives, indicating a bias towards predicting non-delays. The Ensemble model, leveraging the strengths of all three models, achieves a balanced performance with improved accuracy in both true positive and true negative predictions, demonstrating its robustness in delay prediction.

Feature importance for the models is visualized and found in Appendix 11. The importance values were normalized to ensure a fair comparison. This is because feature importance calculations for different models varies, as explained in the methodology chapter. The ensemble model's feature importance demonstrates a balanced contribution from various features since it aggregates the other models' feature importance. The results indicates that the models prefer different features. The feature importance analysis reveals that operational and maintenance events, such as Pushback Events, Bax Events, and Water or Toilet Events, are critical predictors of delays across all models. The XGBoost model also highlights calendar features like Ramp, Day, and Month as significant. The Ensemble model, combining insights from all individual models, identifies Ramp and Pushback Events as the most influential features. The ramp feature is prioritized in the XGBoost classifier, causing it to be the highest influencing feature in the ensembled model as well. These findings underscore the importance of focusing on specific operational activities and time-related factors to mitigate delays effectively.

4.3 Counterfactual explanations

Counterfactual explanations are techniques of machine learning explainability which serve to create insights into important features and how small changes in these features can change the predictions of a model. This study utilizes two types of counterfactual explanations, local feature importance of individual datapoints (4.3.1) and global feature importance (4.3.2).

4.3.1 Counterfactual instances

This section presents the outcomes of the analysis, focusing on the generation of counterfactual instances for delay predictions at Schiphol Airport. A specific instance from the dataset was selected to generate counterfactual explanations (table 9). This query instance, consisted of continuous and categorical features, is an example of an instance where the model predicted a deviation of more than 5 minutes between the TOBT and the AOBT. Keep in mind that the events are translated to the percentage this event took place within the complete turnaround. Using the DiCE ML library, counterfactual instances in table 10 were generated by altering the feature values to achieve the opposite class. These counterfactuals are designed to highlight the minimal changes required to switch the prediction, providing clear guidance on which aspects of the turnaround process are most impactful. The variable that changed in the counterfactual explanation consists of two stars behind their value.

Bax Events	Catering Events	Line Maint. Events	Water Toilet Events	Push back Events	Fuel Events	Pax Events	Capa city	Ramp	Month	Day	Hour
8.32	18.91	73.57	0.00	20.55	0.46	89.90	18	D27	February	Thursday	19.0

Table 9: Counterfactual instance example

Bax Events	Catering Events	Line Maint. Events	Water Toilet Events	Push back Events	Fuel Events	Pax Events	Capa city	Ramp	Month	Day	Hour
8.32	18.91	73.57	0.00	20.55	0.46	89.90	3**	D27	February	Thursday	19.0
8.32	18.91	73.57	0.00	20.55	0.46	89.90	7**	D27	February	Thursday	19.0
8.32	18.91	73.57	0.00	14.30**	0.46	89.90	18	D27	February	Thursday	19.0
8.32	18.91	73.57	0.00	12.10**	0.46	89.90	2**	D27	February	Thursday	19.0
5.75**	18.91	73.57	0.00	20.55	0.46	89.90	18	D27	February	Thursday	10.00**
8.32	18.91	73.57	0.00	20.55	0.46	89.90	4**	D27	February	Thursday	10.00**
8.32	18.91	73.57	0.00	20.55	0.33**	89.90	7**	D27	February	Thursday	19.0
8.32	18.91	73.57	0.00	20.55	0.46	89.90	11**	D27	February	Thursday	19.0
8.32	18.91	73.57	0.00	20.55	0.46	89.90	3**	D27	January**	Thursday	19.0
8.32	18.91	61.08**	0.00	20.55	0.46	89.90	18	D27	February	Monday*	19.0

Table 10: Counterfactual explanations for instance example

These insights are intended to guide operational adjustments aimed at minimizing delays. While not all features are practically changeable, counterfactuals still offer valuable insights into the root causes of DOBT delays. The tables in this section present only the counterfactuals of a single datapoint, but using this example and implementing it in a dashboard were other datapoints and even new datapoints can be used creates unlimited possibilities in post analyses into turnaround delays.

4.3.2 Global feature importance

The global counterfactual feature importance plot for the XGBoost model (figure 5) reveals the key factors influencing delay predictions. This plot was generated by analysing the training data using the DiCE ML's global counterfactual feature importance method, which calculates the importance of each feature based on counterfactual explanations. Specifically, it used 10 counterfactuals for each datapoint to understand how changes in each feature affect the model's predictions, without applying any post hoc sparsity parameter.

The plot indicates that Pax Events is the most influential feature in predicting delays, followed by Water or Toilet Events and Pushback Events for the XGBoost model. The Random Forest model also found Pax Events as an important global counterfactual feature, but also presents catering events as an important feature Other significant features include Fuel Events, and Line Maintenance Events. The global feature importance plots define the events durations as the most important features for creating counterfactual explanations. These insights are crucial as they highlight the operational areas that have the most impact on delay predictions, thereby allowing stakeholders to focus on these aspects to improve on-time performance.

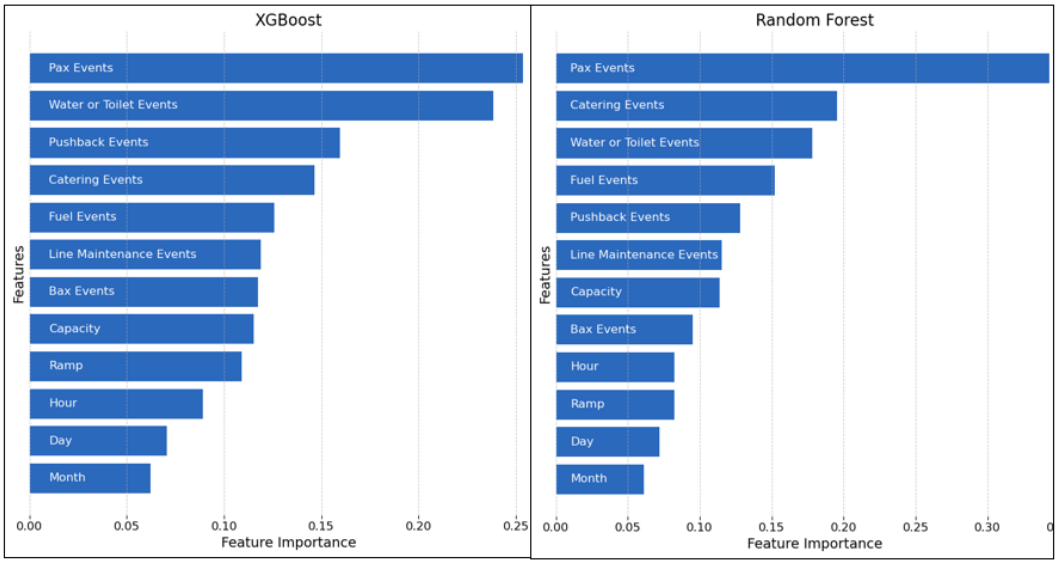


Figure 3: Global counterfactual feature importance

The LightGBM model, and therefore also the ensembled model could not be utilized for counterfactual generations. This is because, if 2 to the power of max depth is greater than number of leaves, the model does not fully utilize the depth of the tree, resulting in lower accuracy. By default, number of leaves is set to 31, which might be insufficient if you have a large max depth. This is the case since the max depth was tuned to 10.

4.4 Dashboard

In this section, the focus lies on the dashboard application developed for predicting and analysing turnaround delays in airport operations. Screenshots of the dashboard is presented in Appendix 13. In the

screenshots, the functionality of the dashboard is displayed by selecting a flight ID, predicting delay, and generating counterfactual explanations. The dashboard functions as an intuitive and interactive interface, enabling stakeholders to input specific flights and obtain predictions and counterfactual analyses. The dashboard was designed to provide a user-friendly experience. Users have the option to select a flight ID and input the turnaround time. Upon clicking the "Predict and Generate Counterfactuals" button, the dashboard communicates with the backend server, which employs the Random Forest machine learning model to predict the turnaround delay and generate counterfactual scenarios. Results are presented in two sections: the Original Data Point and Counterfactuals. The Original Data Point section displays the details of the selected flight. The Counterfactuals section showcases alternative scenarios generated based on the input parameters, enabling stakeholders to explore potential outcomes under different conditions. Changes between the original datapoint and the counterfactual explanation values are marked orange for easy readability.

5. Discussion

This discussion chapter addresses the challenges and insights gained from predicting the delay of turnarounds at Schiphol Airport. It discusses literature findings (5.1), which were the foundation for this research. Furthermore, it goes of the results found in investigating feature importance (5.2). Thereafter, it explains the problems found from a data perspective (5.3). Then, it discusses external factor that influence turnaround times (5.4). Finally, it discusses the results concerning the created dashboard (5.5). The section answers the research sub questions and provides practical implications for improving operational efficiency and resource allocation.

5.1 Literature findings

The first sub question aims to research TOBT and how it is estimated. The research questions states: “What is Target Off-Block Time and how is it estimated?”. The literature review outlines that the TOBT is a crucial metric in the Airport Collaborative Decision-Making A-CDM process, which is designed to enhance airport operations through improved predictability and efficiency. TOBT represents the planned time at which an aircraft is expected to be ready for departure from its parking position. The initial TOBT is equal to the SOBT, which is based on the flight’s scheduled departure time. TOBT can be updated several times during the turnaround process based on the progression through various milestones. Each milestone reflects key stages in the aircraft’s arrival and turnaround, and any delays in these stages can trigger updates to the TOBT. This system ensures that TOBT is dynamically adjusted to reflect real-time conditions and operational status, thereby optimizing departure schedules and runway utilization (EUROCONTROL, 2017b; Schiphol Airport, 2024).

The second sub-question states: “Which prediction machine learning models are currently used for Target Off-Block Time estimation?”. Based on the literature review, multiple models were considered in the pro-con analysis, but Random Forest, LightGBM, and XGBoost were found to be the most applicable due to their performance and suitability for the nature of the data. Random Forest uses an ensemble learning method based on decision trees, offering high accuracy and robustness against overfitting. It is particularly useful for handling large datasets with many features, making it a reliable choice for predicting DOBT (Breiman, 2001; Liaw & Wiener, 2002). LightGBM is known for its efficiency and scalability, LightGBM can handle large-scale datasets and provides faster training speed compared to traditional gradient boosting methods (Ke et al., 2017). XGBoost enhances the gradient boosting algorithm's performance and scalability by incorporating regularization techniques to control overfitting. It combines the strengths of both boosting and bagging techniques (Chen & Guestrin, 2016). Given that the prediction involves a classification problem, a voting system was created to facilitate the development of an ensemble model. This approach combines the strengths of the individual models, balancing their biases and leveraging their collective strengths to improve prediction accuracy and robustness (Dietterich, 2000).

The third sub question concerns the case of Schiphol and how for this airport a successful prediction model can be designed. It states: “How can machine leanirng mdoels be used to predict Target Off-Block Time at Schiphol Airport?”. Schiphol Airport provides a unique environment for implementing machine learning models to predict DOBT due to its adoption of the A-CDM process and the availability of data from the Deep Turnaround system. This AI-driven solution offers real-time insights and predictive capabilities by capturing and analysing over 70 distinct turnaround events through image-based processing. The Deep Turnaround system utilizes two cameras that take snapshots of the turnaround activities every five seconds, collecting data on various operational events. This extensive dataset includes historical, real-time, and predictive information that can be used to train machine learning models. By leveraging the data from the Deep Turnaround system and employing advanced machine learning models, Schiphol Airport can significantly enhance its ability to predict and manage TOBT. This approach not only improves the accuracy

of TOBT predictions but also enhances overall operational efficiency and reduces delays, ultimately contributing to better airport management and passenger satisfaction.

5.2 Feature importance

This study was designed to research the important factors during the aircraft turnarounds of Schiphol Airport. Therefore, the important of features influencing turnaround delays are measured by statistically analysing their impact on delay (5.2.1). Furthermore, machine learning models are created to predict this delay. Researching these model's most important features creates insights in important variables during the turnaround (5.2.2). Understanding feature importance is also crucial for interpreting predictive models and improving their accuracy. When creating counterfactual explanations using the DiCE framework, more feature importance was researched by calculating the most important features for creating counterfactuals (5.2.3). In this section, the implications of using these different interpretations of feature importance are also discussed.

5.2.1 Statistical results

The statistical analysis conducted in this study provided insights into the significance of various continuous event features. Two types of analyses are conducted. First, the correlation analysis which employs Spearman's rank correlation coefficient to evaluate the correlation between numerical features and the target variable (DOBT). Second, linear regression analysis was used to indicate their effect on the target variable. However, the low R-squared values indicated that these models explain only a small fraction of the variance in DOBT, suggesting the presence of additional influencing factors.

Using this, the following results are found:

- Bax Events: Although Bax Events showed a significant correlation with DOBT (p-value < 0.001), the correlation coefficient is very weak (0.0421), and the linear regression model only explains 0.18% of the variance (R-squared = 0.0018). This indicates that while Bax Events are statistically significant, they contribute minimally to the variability in DOBT.
- Catering Events: Catering Events have a negative correlation with DOBT (-0.0462), suggesting that more catering events might slightly reduce the off-block time. However, like Bax Events, the R-squared value is low (0.0021), explaining just 0.21% of the variance, which highlights limited predictive power.
- Line Maintenance Events: These events showed a weak positive correlation (0.0689) and had slightly higher explanatory power in the regression model (R-squared = 0.0029), indicating a somewhat stronger influence on DOBT compared to Bax and Catering Events, yet still minimal overall impact.
- Pushback Events: This feature has the strongest correlation (0.2073) among the analysed features and a relatively higher R-squared value (0.0096), explaining 0.96% of the variance in DOBT. Although this is still a small percentage, it underscores Pushback Events as a more critical factor in determining TOBT.
- Pax Events: Both features are statistically significant with weak positive correlations (0.0653). The R-squared value is low (0.0018), indicating limited but notable contributions to the variability in DOBT.
- Water or Toilet Events: This feature did not show a significant correlation with DOBT and had the lowest R-squared value (0.0006), indicating it does not meaningfully explain the variance in TOBT.
- Fuel events**:** These events showed correlation significance, also being a weak positive correlation (0.0519). The R-squared value is very low (0.0020), indicating limited but some contributions to the variability in TOBT and AOBT.

In summary, while the statistical analysis shows that several event features have significant relationships with DOBT, the overall explanatory power of these individual features is low. This indicates that these factors have limited influence in the off-block delay in the way that they are measured. There have been several difficulties in accurately documenting the events' durations, which may have degraded the correlation results, which will be discussed in paragraph 5.3. Nevertheless, these events also indicate that there are other external factors influencing the delays.

The second part of the statistical analysis involved categorical and discrete numeric variables. For these variables, the Kruskal-Wallis Tests was used since the ANOVA assumptions were not met. The Kruskal-Wallis test results indicate significant differences between the groups for all categorical features. This implies that there are substantial differences in TOBT across different categories of handlers, days, months, alliances, carrier types, hours, and capacities. For instance:

- Handlers: Significant differences in TOBT were observed across different types of handlers (e.g., main handler, apron handler, baggage handler), suggesting that the efficiency and methods of different handlers substantially affect turnaround times.
- Time-Related Features: Day, month, and hour of the day showed significant variability in TOBT, indicating that temporal factors play a crucial role in determining off-block times. These findings suggest that operational strategies might need to be adjusted based on time-related patterns.
- Operational Context: Variability in DOBT across different alliances and carrier types highlights the impact of operational practices and policies specific to different airlines and their partners.

The results for categorical features show significant differences in DOBT across various groups, indicating that these factors have a meaningful impact on turnaround times. The variability between groups can be found in Appendix 7, which displays boxplots of different groups. For the handler companies, generally KLM's handlers outperform the rest of the handler companies. They are the most occurring company since this research was executed on KLM's hub airport. Therefore, these results show that they have finetuned their operations. Time-related events indicate the effect of rush hours at Schiphol Airport's turnaround ramps. Boxplots show a general rise in DOBT for capacity in its mean and standard deviation, indicating that when more turnarounds happen simultaneously, DOBT delay has a greater change of occurring. Furthermore, the peak times at Schiphol also show higher changes of delay. This also indicates that the factor of more operational activity at the airport influences DOBT. The alliance where KLM belongs to, SkyTeam, also slightly outperforms the other alliances.

5.2.2 Feature importance from prediction models

The feature importance plots from the predictive models provide critical insights into which features most significantly impact the prediction of DOBT. These models, using different algorithms and techniques, using different aspects of the data that are influential in making accurate predictions. Understanding the differences is important for interpreting the models' feature importance.

The Random Forest model uses an ensemble learning method based on decision trees, identifies Pushback Events, Bax Events, and Water or Toilet Events as the top three most important features. This model aggregates the predictions of multiple decision trees to improve accuracy and control overfitting. In Random Forests, feature importance is often measured by the average decrease in impurity (Gini importance) across all trees (Breiman, 2001; Liaw & Wiener, 2002):

- Pushback Events: Consistently recognized as the most critical feature, likely due to its direct impact on the final phase of the turnaround process.
- Bax Event and Water or Toilet Events: These operational activities are crucial, reflecting significant steps in the preparation and servicing of the aircraft.

- Ramp, Fuel Events, and Catering Events: These features also show high importance, indicating their roles in the variability of TOBT.
- Time-Related Features: Day, month, and hour provide temporal context, affecting operational efficiency and resource allocation.

LightGBM is known for its efficiency and scalability, particularly with large datasets. It uses a histogram-based algorithm to bucket continuous feature values into discrete bins, speeding up the training process. The feature importance in LightGBM is derived from the number of times a feature is used to split the data across all trees (Ke et al., 2017):

- Bax Events and Pushback Events: Like Random Forest, these events are top features, show their central role in determining TOBT for this model.
- Water or Toilet Events and Catering Events: Highlighted due to their operational significance during the aircraft's turnaround.
- Fuel Events and Ramp: Important for their contributions to overall turnaround efficiency. The ramp is again the most important categorical feature.
- Time-Related Features: Again, these features are least significant, underscoring the limited importance of temporal patterns.

XGBoost enhances the gradient boosting algorithm's performance and scalability. It uses a more regularized model formalization to control overfitting. In XGBoost, feature importance can be measured by the gain, cover, or frequency of splits involving each feature (Chen & Guestrin, 2016):

- Ramp: By far the most dominant features in XGBoost, indicating that the operational ramp environment heavily influences DOBT for this model.
- Calendar Features: The inclusion of day and month signifies the model's sensitivity to periodic patterns and schedules.
- Others: Different from other models, all other features are significantly less important to this model.

The ensemble model combines insights from Random Forest, LightGBM, and XGBoost, balancing their individual biases and leveraging their collective strengths (Dietterich, 2000). The averaged feature importance values provide a comprehensive view:

- Ramp and Pushback Events: These features emerge as the most influential, reflecting their critical roles across different modeling techniques. The feature importance of Ramp can be explained by the high importance it received from the XGBoost model.
- Bax Events and Water or Toilet Events: Consistently important, indicating key operational activities.
- Day, Catering Events, and Fuel Events**:** These features also show significant influence, highlighting the multifaceted nature of turnaround processes.
- Time-Related Features**:** Found in the middle pact in terms of importance for the ensembled model, mostly because of their importance in the XGBoost model.

In summary, while there is a consensus on the importance of certain features like Pushback Events and time-related features, the specific ranking and emphasis vary due to the technical nuances of each model. The consistency of the Pushback, Bax and Water or Toilet events as important model features indicate their influence on the predictions of the model. These insights help in understanding the models' behaviour and indicate operational improvements at Schiphol Airport.

5.2.3 Feature importance from global counterfactuals

Global counterfactual explanations provide a clear perspective on feature importance by showing how small changes in specific features can change model predictions. This method enhances the interpretability of machine learning models, allowing stakeholders to understand which factors most influence the outcome and how adjustments can impact turnaround times. By examining global counterfactual explanations, we can identify the features that play important roles in determining DOBT and discuss their implications. The feature importance of counterfactual explanations is only researched for the Random Forest and XGBoost models, as LightGBM has complexities with its hyperparameters to run DiCE. The insights gained from counterfactual explanations improves transparency, making it easier for stakeholders to understand the rationale behind predictions and creating greater trust in automated decision-making systems, thereby facilitating their integration into airport operations (Ribeiro, Singh, & Guestrin, 2016).

Pax Events (boarding and disembarking times) were shown to have the most significant impact on predictions. This highlights the importance of efficient passenger handling in minimizing delays. The feature importance plots from both XGBoost and Random Forest models confirm Pax Events as the top feature, showing their substantial influence. Improving passenger handling processes, such as optimizing boarding and disembarking procedures, can lead to significant reductions in turnaround times. Training ground staff and utilizing advanced boarding techniques can enhance efficiency in this area. This is a known bottleneck during the turnaround and there are several studies and experiments that tries to improve this. For instance, a study by Cook et al. (2021) found that implementing a dynamic boarding system, which adjusts the boarding process based on the real-time status of the aircraft and passenger flow, significantly reduced boarding times compared to the traditional methods. Another example is the use of biometric boarding gates, as experimented by Delta Air Lines, which streamlined the boarding process by reducing the need for manual checks and thereby speeding up the entire boarding procedure (Delta Air Lines, 2020). Additionally, a research study by Milne and Kelly (2020) on the "WilMA" (Window-Middle-Aisle) boarding method demonstrated it to be more efficient than the back-to-front method traditionally used by many airlines.

Pushback Events consistently emerged as a critical factor in both the predictive models and counterfactual explanations. Adjusting the duration of pushback events significantly influenced the predictions, underscoring its important role in determining DOBT. This finding aligns with the operational importance of pushback as it marks the final phase of the turnaround process, directly preceding departure. Efficient management of pushback events can directly reduce turnaround times. Implementing standardized procedures and ensuring timely execution can mitigate delays associated with this critical phase. Features such as the day of the week, month, and hour of the day were less important in the counterfactual explanations. Adjusting these temporal features did often not result in significant changes in the predicted turnaround times, underscoring the low influence of scheduling and time management. This can be explained by Schiphol's ability to have enough resources ready for turnarounds. Furthermore, since the research was don from data of November until May, summer holiday peaks are not considered. Other operational events, such as Line Maintenance Events and Fuel Events, had medium significance in the counterfactual explanations. Adjustments in these events for some datapoints led to changes in predictions, indicating their importance in the overall turnaround process. Efficient management of maintenance and fuelling operations is found an important factor in turnaround delays.

Global counterfactual explanations provide valuable insights into the factors influencing TOBT, highlighting the importance of pushback events, passenger handling, and temporal features. The visual analysis of feature importance further supports these findings, offering a understanding of the key drivers of turnaround times at Schiphol Airport.

5.2.4 Most important features

To answer the research question "Which metrics are important when estimating Target Off-Block Time?", the research conducted identified several key features that are crucial for estimating DOBT. Through statistical analysis, machine learning models, and counterfactual explanations, the most important metrics have been highlighted. Pushback Events emerged as the most influential factor across all methods, significantly affecting TOBT predictions due to their role in the final phase of the turnaround process. Pax Events also demonstrated substantial impact, emphasizing the need for efficient passenger handling to minimize delays. Other notable metrics include Bax Events**,** Water or Toilet Events, and Catering Events, which, although showing weaker correlations individually, contribute meaningfully to the overall prediction models. Time-related features such as the day of the week**,** month, and hour of the day showed variability in importance, often reflecting operational patterns and peak times at the airport. Overall, understanding these key metrics can lead to improvements in TOBT predictions and airport turnaround efficiency.

5.3 External Factors Influencing Turnaround Delay

External factors play a crucial role in influencing turnaround delays at airports. These are not taken in consideration for this research, which aimed to research delay purely on the data mentioned. These factors, often beyond the control of the airport or airlines, can significantly impact the efficiency of turnaround operations.

Adverse weather conditions, such as heavy rain, snow, fog, or thunderstorms, can severely disrupt airport operations. These conditions can delay flights, affect ground handling activities, and reduce visibility, making it difficult for ground staff to perform their tasks efficiently. Studies have shown that weather is a major cause of delays at airports worldwide (Schultz et al., 2018; Sanz et al., 2021). Although using robust machine learning models with loads of data should generally not be affected by weather conditions, doing post turnaround analysis for individual turnarounds should take weather conditions during the turnaround into consideration.

ATC delays occur when there is congestion in the airspace or at the airport. These delays can result from high traffic volumes, limited runway capacity, or ATC strikes. Such delays can disrupt the planned sequence of arrivals and departures, leading to cascading delays in turnaround operations (Hao & Hansen, 2018; Schultz et al., 2018). This occurs from time to time at Schiphol, due to the limited capacity it can house. Schiphol Airport is one of the busiest airports of Europe and is often challenging its own runway capacity in peak departure hours. ATC delay can therefore more regularly occur due to a domino effect of delays, requiring aircraft to wait for each other.

Compliance with aviation regulations, such as security checks and maintenance requirements, can introduce delays. These regulatory requirements, while essential for safety and security, can be time-consuming and impact the overall turnaround time (Holloway, 2016; Tretheway & Andriulaitis, 2015). The so-called preflight checks must be carefully executed before an aircraft can be cleared to leave the gate and can therefore be a factor in turnaround delay.

Unexpected technical issues, such as mechanical failures or equipment malfunctions, can cause significant delays. These issues often require immediate attention and repair, which can extend the turnaround time considerably. The availability of technical support and spare parts also influences the duration of these delays (Rebollo & Balakrishnan, 2014; Zhang & Lian, 2019). These delays are often found inside the body of the plane rather than outside, making it hard to detect for AI imagery detection cameras.

Delays caused by passengers, such as late arrivals at the gate, issues during boarding, or last-minute ticketing and baggage problems, can also impact turnaround times. Efficient passenger management is crucial to minimize these delays (Wang et al., 2018; Lee & Lee, 2016). Furthermore, Schiphol Airport is part

of a hub and spoke network. This means that flights all over Europe arrive at Schiphol Airport to make a transfer too for example transatlantic destinations. Schiphol Airport is also often used as a transfer airport inside Europe. This means that often departing flights are depended on arriving flights to fill the plane with passengers. If one of the arriving flights is delayed, it can cause delay for the departing flight.

Limited airport infrastructure, such as insufficient gates, parking spots, or ground handling equipment, can cause bottlenecks during peak times. These constraints can slow down the turnaround process and increase the likelihood of delays (de Neufville & Odoni, 2016; Delcea et al., 2018). At Schiphol Airport, these constraints are particularly significant during peak travel seasons or times of high passenger traffic. The demand for gates and parking spots often exceeds the available supply, causing aircraft to wait for a spot to become available. This waiting time can disrupt the planned sequence of arrivals and departures, leading to cascading delays. Additionally, the efficiency of ground handling operations, which include baggage handling, fuelling, and catering, can be impacted by the availability and capacity of ground handling equipment (Schiphol Group, 2021).

5.4 Post analysis dashboard for turnaround delay

The fifth research question concerned: “How can machine learning explainability lead to counterfactual insights?” To be able to present counterfactual insights to the focal company of LVNL a dashboard is created to enable operational usages of generating counterfactuals. The dashboard developed for predicting aircraft turnaround times at LVNL is a tool designed to improve operational efficiency and decision-making processes. It captures detailed data on various turnaround events such as Bax Events, Fuel Events, and Catering Events. Events are recorded with precise timestamps, enabling accurate duration tracking and analysis.

One of the dashboard’s core functionalities is its predictive analysis capability. Using machine learning algorithms, the dashboard predicts the difference between the TOBT and the AOBT. These predictions help in identifying potential delays in the turnaround process, thereby enabling more effective scheduling and resource allocation. The predictive feature enables proactive decision-making, allowing stakeholders to anticipate and mitigate delays before they impact flight schedules.

The dashboard also generates counterfactual scenarios, which are alternative scenarios that show how different variables might impact turnaround times. This functionality is particularly useful for identifying the key factors that contribute to delays. By understanding these factors, LVNL can implement targeted interventions to improve operations and reduce turnaround times delay occurrences. The counterfactual analysis provides a deeper insight into the potential improvements that can be made in the turnaround process.

At LVNL, the dashboard that is created during this research can be utilized to improve decision-making by providing data and can create a different way of doing post hoc analysis. This capability supports more informed decisions, ultimately improving turnaround efficiency. Furthermore, by identifying the causes of delays, the dashboard aids in optimizing resource allocation, which helps in reducing operational costs. Accurate predictions also facilitate better scheduling, minimizing delays and enhancing the reliability of airport operations. Continuous monitoring of performance metrics ensures that LVNL can make data-driven decisions to improve turnaround processes continually.

The final sub question stated: “How can a dashboard be created to overview the models’ predictions?”. A dashboard to overview the models' predictions can be created by combining Flask and Dash frameworks. Flask serves as the backend to handle API requests, manage machine learning models, and generate predictions and counterfactual explanations. Dash provides the front-end interface, enabling users to interact with the data, view predictions, and visualize results through an intuitive and interactive web-

based dashboard. This setup ensures a robust and user-friendly platform for real-time decision-making. Screenshots of the tool can be found in Appendix 13.

5.5 Limitations

The accuracy and reliability of predictive models for aircraft turnaround times at airports are significantly affected by various forms of noise and hard-to-define events in the data. In this discussion, these issues are explained, outlining the specific challenges encountered and their implications for data analysis and model performance. It first discusses which noise was found in the data (5.3.1). Thereafter, it discusses the various challenges in defining events (5.3.2). Then, there is a section that discusses how these issues impact model performance (5.3.3).

5.5.1 Noise in the Data

Noise in data refers to random variability or errors that obscure the true signal. The main noise in the data that is an issue is the inconsistent data recording. All events that are captured by the DeepTurn cameras are documented in the data as a DateTime datatype. This means that the cameras detect an event occurring and record that events' date and time of that camera frame. These cameras have an update interval of five seconds, after each interval the camera looks for new events on the ramp to record. The problem with this is that the camera often detects the start of the event but does not capture the end. An example of this problem occurring is during the refuel process of the turnaround. There are six subevents for this main event recorded in the data: First Fuel Truck Appears, First Fuel Truck Connects, First Fuel Truck Stops in Position, Last Fuel Truck Disappears, Last Fuel Truck Finalizes, Last Fuel Truck Moves Out of Position. The problem does not occur in capturing the appearance and disappearance of the first and the last fuel truck, as this is correctly captured for most of the turnarounds. It does occur for the other three events, which is an issue since these events are more accurate features to define the fuelling event. There are multiple datapoints where the fuel truck is in position and is connected but have no recorded data of the fuel truck finalizing and moving out of position. Similarly, this also happens the other way around. And then there are also instances where the fuel truck does stop in position, does not connect, does finalize but does not move out of position. This indicates that the DeepTurn cameras and there AI image recognition systems have some error in their system. This is not an issue exclusive to the fuelling event, it occurs for almost every event it captures.

5.5.2 Hard-to-Define Events

Certain events in the turnaround process are difficult to define and capture accurately due to multiple reasons. The most important problem is defining start and end points of events from the data. All turnaround activities are composed of several sub-events. For example, passenger boarding involves multiple steps such as connecting the passenger bridge, opening the aircraft door, and actual boarding of passengers. The difficulty in defining when an event truly starts and ends adds to the complexity. For some events, such as the previous example of the fuel truck, there can be clear start and end definition. This can be done based on the connect and finalize data, but it can also be defined using the in and out of position timestamps. For other events, this is not the case. The best example is the water and toilet event during the turnaround. There are only two timestamps captured for this event in the DeepTurn data, the truck appearing and disappearing. It does not mention anything about the start and end of the event. Using these sub events to measure the duration of time the water and toilet events presents multiple problems. First, when this truck drives somewhere within the range of the cameras, these events are captured, even when the truck was not involved in the plane's turnaround. Measuring the event time for these occurrences resulted in an event of less than two minutes, creating error in the dataset. Second, often this event took way longer than it should have lasted, due to that the truck was not disappearing from the

ramp. Water and toilet trucks often appeared on the ramp and stayed there for an hour, which is an unrealistic duration of the event. Third, there is no reliability in measuring this event. There are no indicators of the event's duration. There could be instances where the truck drove into the camera's view and parked there for a couple minutes until eventually disappearing again. This would lead to a falsely recorded event.

Events like maintenance checks or cleaning may not have clear start and end points. These events often happen from inside the aircraft, making it impossible to be captured by the outside cameras. Of course, there could be event registration done based on vehicles outside of the aircraft, but there is no certainty that these events are happening and there are no clear start and end points. Furthermore, unexpected maintenance can cause a lot of extra delay during the turnaround. This is difficult for the cameras to detect and can influence prediction model drastically without capturing the reason of delay.

Turnaround processes involve multiple concurrent activities, such as refuelling, loading baggage, and boarding passengers. These activities are often interdependent, and delays in one can cascade into delays in others. Capturing and modelling these interdependencies accurately is challenging. An example is that when a flight arrives at the gates, there may be problems with deboarding the plane. This could be due to wounded people or due to limited deboard entrances available. When people are still within the aircraft, it is prohibited to refuel the plane due to safety reasons. But when the refuelling truck is already in position and defining this as the start of the fuelling event, this can cause misleading event durations. Another problem can arise from boarding the plane, since passengers arriving late at the gate is an often-occurring problem. This may lead to event durations, such as pushback events, to last longer due to external reasons.

Based on the documentation of DeepTurn (Schiphol DeepTurn department, 2023) there should be 72 different sub events captured within the data. Unfortunately, in the data presented for this research, there were only 68 events available. The missing events were found in the start of boarding and end of boarding of passengers, and in the start and end of loading baggage into the aircraft. While just missing four sub events does not look that bad, these sub events would be of create importance for creating events. If they were available in the data, deboarding events and boarding events could be split in two. The same is the case for unloading and loading of baggage into the plane. These events should be handled as separate events, they do not directly rely on each other and happen during the start and the end of the turnaround phase. In the current research, as explained in the methodology, these events are regarded as a combined pax events and bax events, due to the limited event information. This is technically unfair to do and splitting these events into two would greatly add value to this research.

5.5.3 Effect on research

The presence of noise and hard-to-define events poses several challenges to the development and performance of predictive models. Noise and ambiguous events reduce the data quality, making it difficult for models to learn the true patterns in the data. This can lead to lower prediction accuracy and higher error rates. Models trained on noisy data are likely to overfit the noise rather than the underlying patterns (Van Buuren, 2018). Models trained on noisy and ambiguous data are likely to be less robust. They may perform well on training data but fail to generalize to new, unseen data. The variability in the data can cause the models to make inconsistent predictions under slightly different conditions (Tercan & Meisen, 2022). The interpretability of models is compromised when the input data is noisy or poorly defined. It becomes challenging to draw meaningful insights from the model outputs or to provide actionable recommendations based on the predictions. The presence of noise can obscure the true relationships

between features and the target variable, making it difficult to explain why the model made a particular prediction (Van Buuren, 2018). To conclude, having these issues greatly affected the predictability performance of the models. This also caused unreliable feature importance results. The inaccurate way of measuring event duration also influenced the statistical results of this research. Finally, the counterfactuals explainability outcomes may exposes small changes in variables that are unreliable due to the performance of the prediction models. This is the case for the global counterfactual feature importance as well as the outcomes from individual datapoint counterfactuals, for example when utilizing the dashboard.

6. Conclusion

This research investigates the deviations between Target Off-Block Time (TOBT) and Actual Off-Block Time (AOBT) at Schiphol Airport. For this, a new variable is defined, called Delta Off-Block Time (DOBT) which implies the difference between the two. Through a combination of statistical analysis, machine learning modelling, and counterfactual explanations, several critical findings and practical recommendations have emerged. The main research question of this report aims to answer is:

What are the main factors influencing Target Off-Block Time estimation deviations during airport turnaround processes at Schiphol Airport?

The study found key operational activities, such as the duration of baggage handling, pushback, and passenger handling events, as significant predictors of TOBT deviations. Additionally, temporal factors, including the time of day and specific days, substantially influence TOBT accuracy, with peak hours showing higher deviations.

Noise in the data, particularly due to inconsistent event recording by the DeepTurn cameras, poses significant challenges. Events such as refuelling are often incompletely captured, leading to inaccuracies in the data. Furthermore, hard-to-define events, like the exact start and end times of certain turnaround activities, add complexity to the data analysis. For instance, the water and toilet servicing events are poorly defined with only appearance and disappearance timestamps, leading to unrealistic duration measurements. Maintenance checks and cleaning activities, often happening inside the aircraft, are difficult to capture accurately, affecting the reliability of the data.

The study also notes missing sub-events in the DeepTurn data, which are crucial for accurately defining and splitting turnaround activities. The absence of these sub-events necessitates combining certain events, which could have been more precisely analysed if the missing data were available.

The ensemble model, integrating Random Forest, LightGBM, and XGBoost, demonstrated robust performance in predicting TOBT deviations. Counterfactual explanations provided clear, actionable insights, highlighting the minimal changes needed to alter outcomes. For example, reducing the duration of specific turnaround events by a few minutes can significantly align TOBT with AOBT, thus minimizing delays.

A user-friendly dashboard, developed using Flask and Dash frameworks, allows stakeholders to visualize model predictions and counterfactual insights. This tool supports informed decision-making and operational adjustments, enabling proactive measures to mitigate potential delays and enhance overall efficiency. By integrating predictive models and counterfactual explanations into operational systems, Schiphol Airport can improve TOBT accuracy, leading to better planning and coordination among various stakeholders.

This research significantly advances the understanding and management of TOBT deviations at Schiphol Airport. Combining statistical analysis, machine learning, and explainability tools provides a comprehensive framework for improving turnaround processes. Implementing these insights will enhance operational efficiency and decision-making, benefiting Schiphol Airport and potentially serving as a model for other airports.

7. Recommendations

Based on the findings of this research, several key recommendations are proposed to enhance the accuracy of TOBT predictions and improve the overall efficiency of turnaround processes at Schiphol Airport. These recommendations focus on improving data collection, accounting for external factors, and redefining events for better precision.

1. **Improving Data Collection**

One significant improvement would be enhancing data collection by the cameras. It is recommended to explore ways to reduce noise in the data, which is further explained in the discussion chapter (5.5.1). Possibilities to improve this are using cameras with higher updated frames, research improvements in cameras coverage and generally keep monitoring the performance of the AI recognition cameras.

Cameras with higher frame rates can capture events more frequently and accurately, reducing the chances of missing critical sub-events. Additionally, enhancing the AI algorithms for better detection and classification of events is essential. Advanced machine learning models trained to recognize the different turnaround activities with higher accuracy can significantly improve data quality. Furthermore, keep training the AI recognition model based on more data can improve the performance.

Ensuring complete coverage is another potential improvement. The cameras should cover all critical areas and angles of the ramp to capture all relevant activities without blind spots. In the current situation, two cameras are used to detect the turnaround events, but there should be explored if more cameras could improve this. Enhanced coverage will help in capturing the start and end of all events more accurately, thereby improving the overall reliability of the data.

Regular maintenance and calibration of the cameras and AI systems should be implemented to ensure they function optimally and provide accurate data consistently. A regular maintenance and calibration schedule will help in maintaining the quality of data collection, minimizing errors, and ensuring the systems are up to date with the latest technological advancements.

2. **Redefining Events Using Start and End Sub-Events**

A critical aspect of improving data accuracy is the detailed definition of each event in the turnaround process by clearly specifying start and end sub-events. Developing standardized protocols for identifying the exact start and end points of each event is essential and is insufficiently possible with the current timestamps. More on this is discussed in chapter 5.2.2. These protocols should aim to avoid using appear and disappear as event identifiers. Using these features as start and end definitions for events proved to be unreliable.

- Fuel events: First Fuel Truck Stops in Position → Last Fuel Truck Moves Out of Position

The fuel events can successfully be recorded with the uses of these two subevents. These events are already captured by the cameras and therefore can easily be used. These events should be used over the First Fuel Truck Connects and Last Fuel Truck Finalizes due to the higher amount of noise in capturing those events.

- Pushback events: Tug Idle Connected Starts → Tug Idle Connected Ends

During this research, the pushback events were measured based on the Tug Idle Connected Starts and Aircraft Moves Out of Position events. This was because Tug Idle Connected Ends was regularly missing in the data. By improving the data collection as recommended, this can be improved so that this event is more accurately defined.

- Bax events: Bax Unloading Starts → Bax Unloading Stops
  Bax Loading Starts → Bax Loading Stops

One of the problems faced with the data was that documentation of DeepTurn included events that were missing in the data. If these events are included in the data, it would enable to split the bax events into unloading and loading. This would greatly improve the prediction models since these are separate events during the turnaround.

- Line Maintenance events: First Line Maintenance event → Last Line Maintenance event

This event will be dependent on the possibilities in detecting sub events. In the current data, it defines First Oil Check Truck Appears, Last Oil Check Truck Disappears, Power Connects and Power Disconnects. As stated, it should be avoided to use appear and disappear as event definitions. Therefore, there should be explored to capture other events, such as First Oil Truck Stops in Position. Furthermore, it should be explored if more line maintenance events can be captured.

- Water/Toilet Events: First Water/Toilet Truck in Position → Last Water/Toilet Truck In Position

These events are not yet available in the data since they are not captured by the DeepTurn cameras. For now, only the appears and disappears events are registered, but as stated this is not acceptable as accurate start and end events. Therefore, it should be explored to find ways to recognize the positional events.

- Catering Events: First Catering Truck Starts Ascent → Last Catering Truck Completes Descent

For catering events, the start of the first truck ascending is missing. This should be added to the DeepTurn data since it is a more accurate way of identifying the start of catering events then by using First Catering Truck Completes Ascent. This is because when the ascent of the truck is completed, the catering events has already started.

- Pax events: Pax Disembark Board Starts → Bax Disembark Board Stops
  Pax Boarding Starts → Pax Boarding Stops

Just like bax events, pax events should be considered as two separate events. Disembarking is one of the first processes during the turnaround, while boarding happens at the later stages. Accurately recording these events has shown trouble, as there are different ways the plane can be boarded. For example, KLM aircraft prioritize boarding with the use of bridge, focusing over the quality of their services. Low-cost carriers will look for the most efficient way of boarding, which can be for example with stairs leading to multiple entrances of the aircraft. Therefore, different starting and end points need to be identified for these processes. And if bridges are used, it may be hard to recognize when boarding starts, or disembarking is finished. Therefore, for accurate data extern data sources from Schiphol should be used.

By focusing on these two key areas, Schiphol Airport can significantly improve the accuracy and reliability of this predictive models for TOBT, enhance operational efficiency, and reduce turnaround delays. These measures will also provide a more detailed and accurate understanding of the factors influencing turnaround times, facilitating better decision-making, and planning. Improving the data collection and event definitions will likely improve the business issue since research results will be more accurate. Moreover, predictive models will potentially be more accurate and counterfactual explanations will be more accurate.

References

Arrieta, A. B., & Ser, J. D. (2020). Plausible counterfactuals: Auditing deep learning classifiers with realistic adversarial examples. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1-7.

Artelt, A. (2019). Ceml: Counterfactuals for explaining machine learning models—a python toolbox. Retrieved from https://www.github.com/andreArtelt/ceml

Ates, E., Aksar, B., Leung, V. J., & Coskun, A. K. (2021). Counterfactual explanations for multivariate time series. In *2021 International Conference on Applied Artificial Intelligence (ICAPAI)* (pp. 1-8). https://doi.org/10.1109/ICAPAI49758.2021.9462056

Balakrishna, P., Ganesan, R., & Sherry, L. (2010). Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of Tampa Bay departures. *Transportation Research Part C: Emerging Technologies, 18*(6), 950-962.

Baron, S. (2023). Explainable AI and causal understanding: Counterfactual approaches considered. *Minds & Machines, 33*, 347-377. https://doi.org/10.1007/s11023-023-09637-x

Belorkar, A., Guntuku, S. C., & Hora, S. (2020). *Interactive data visualization with Python - Second edition: Present your data as an effective and compelling story*. Packt Publishing.

Bhaumik, D., & Dey, D. (2022). An audit framework for technical assessment of binary classifiers. *Amsterdam University of Applied Sciences*. Retrieved from https://arxiv.org/abs/2207.01611

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32.

Breusch, T. S., & Pagan, A. R. (1979). A simple test for heteroscedasticity and random coefficient variation. *Econometrica, 47*(5), 1287-1294.

Brughmans, D., & Martens, D. (2021). NICE: An algorithm for nearest instance counterfactual explanations. *Data Mining and Knowledge Discovery*, 1-39.

Buuren, S. van. (2018). *Flexible imputation of missing data*. Chapman and Hall/CRC.

Carvalho, L., Sternberg, A., Gonçalves, L. M., Cruz, A. B., Soares, J. A., Brandão, D., Carvalho, D., & Ogasawara, E. (2020). On the relevance of data science for flight delay research: A systematic review. *Transport Reviews, 41*(4), 499-528. https://doi.org/10.1080/01441647.2020.1861123

Carreira-Perpiñán, M., & Hada, S. (2021). Counterfactual explanations for oblique decision trees: Exact, efficient algorithms.

Chang, W., Cheng, J., Allaire, J., Xie, Y., & McPherson, J. (2020). *Shiny: Web application framework for R*. Retrieved from https://shiny.rstudio.com/

Chapman-Rounds, M., Schulz, M., Pazos, E., & Georgatzis, K. (2019). EMAP: Explanation by minimal adversarial perturbation.

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* (pp. 785-794).

Cheng, F., Ming, Y., & Qu, H. (2021). DECE: Decision explorer with counterfactual explanations for machine learning models. *IEEE Transactions on Visualization and Computer Graphics, 27*(2), 1438-1447. https://doi.org/10.1109/TVCG.2020.3030342

Chou, Y.-L., Moreira, C., Bruza, P., Ouyang, C., & Jorge, J. (2022). Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications. *Information Fusion, 81*, 59-83. https://doi.org/10.1016/j.inffus.2021.11.003

Cui, Z., Chen, W., He, Y., & Chen, Y. (2015). Optimal action extraction for random forests and boosted trees. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 179-188). ACM. https://doi.org/10.1145/2783258.2783264

Dalmau, R., Ballerini, F., Naessens, H., Belkoura, S., & Wangnick, S. (2019). Improving the predictability of take-off times with machine learning: A case study for the Maastricht upper area control centre area of responsibility.

Dandl, S., Molnar, C., Binder, M., & Bischl, B. (2020). Multi-objective counterfactual explanations. *Parallel Problem Solving from Nature*, 424-438. https://doi.org/10.1007/978-3-030-58112-1_30

Dandl, S., & Molnar, C. (2023). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Independently published.

Dash. (2019). *Dash User Guide and Documentation*. Plotly. Retrieved from https://dash.plotly.com/

De Falco, P., Kubat, J., Kuran, V., Varela, J., Plutino, S., & Leonardi, A. (2023). Probabilistic prediction of aircraft turnaround time and target off-block time: Case studies for Prague, Geneve, Arlanda and Fiumicino international airports using operational data.

Delcea, C., Cotfas, L. A., & Dragos, C. M. (2018). Airport ground operations modeling. *Procedia Computer Science, 138*, 623-630. https://doi.org/10.1016/j.procs.2018.10.082

Deutschmann, A. (2012). Prediction of airport delays based on non-linear considerations of airport systems. In *28th Congress of the International Council of the Aeronautical Sciences 2012 (ICAS 2012)* (pp. 4108-4114).

Dhurandhar, A., Chen, P., Luss, R., Tu, C., Ting, P., Shanmugam, K., & Das, P. (2018). Explanations based on the missing: Towards contrastive explanations with pertinent negatives. *Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018)*.

Dhurandhar, A., Pedapati, T., Balakrishnan, A., Chen, P., Shanmugam, K., & Puri, R. (2019). Model agnostic contrastive explanations for structured data. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems* (pp. 1-15). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45014-9_1

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*. Retrieved from https://arxiv.org/abs/1702.08608

Downs, M., Chu, J. L., Yacoby, Y., Doshi-Velez, F., & Pan, W. (2020). CRUDS: Counterfactual recourse using disentangled subspaces. *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*.

EUROCONTROL. (2017, March 31). *Airport Collaborative Decision-making*. Retrieved from https://www.eurocontrol.int/concept/airport-collaborative-decision-making

EUROCONTROL. (2017). *Airport CDM Implementation Manual* (5th ed.). Retrieved from https://www.eurocontrol.int/sites/default/files/publication/files/airport-cdm-manual-2017.PDF

European Union. (2016). General Data Protection Regulation (GDPR). *Official Journal of the European Union*.

Feir-Walsh, B. J., & Toothaker, L. E. (1974). An empirical comparison of the ANOVA F-test, normal scores test and Kruskal-Wallis test under violation of assumptions. *Educational and Psychological Measurement, 34*(4), 789-799.

Fan, Y., & Zhuang, J. (2020). Mitigating air traffic delays through improved airport operations. *Journal of Air Transport Management, 85*, 101796.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861-874.

Fernández, R. R., Martín de Diego, I., Aceña, V., Fernández-Isabel, A., & Moguerza, J. M. (2020). Random forest explainability using counterfactual sets. *Information Fusion, 63*, 196-207. https://doi.org/10.1016/j.inffus.2020.07.001

Ferrario, A., & Loi, M. (2022). The robustness of counterfactual explanations over time. *IEEE Access*, 1-1. https://doi.org/10.1109/ACCESS.2022.3196917

Field, A. (2018). *Discovering statistics using IBM SPSS statistics*. Sage.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189-1232.

Gao, Y., Huyan, Z., & Ju, F. (2015). A prediction method based on neural network for flight turnaround time at airport. In *2015 8th International Symposium on Computational Intelligence and Design (ISCID)* (pp. 44-47). https://doi.org/10.1109/iscid.2015.44

Gcattan. (n.d.). GitHub - gcattan/git-quality-check: Simple tool to check quality of git commits, and build indicators on it. GitHub. Retrieved from https://github.com/gcattan/git-quality-check

Ganesan, R., Balakrishna, P., & Sherry, L. (2010). Improving quality of prediction in highly dynamic environments using approximate dynamic programming. *Quality and Reliability Engineering International, 26*(7), 717-732.

George, E., & Khan, S. (2015). Reinforcement learning for taxi-out time prediction: An improved Q-learning approach. In *2015 International Conference on Computing and Network Communications (CoCoNet)* (pp. 757-764). IEEE.

Ghasemi, A., & Zahediasl, S. (2012). Normality tests for statistical analysis: A guide for non-statisticians. *International Journal of Endocrinology and Metabolism, 10*(2), 486-489.

Ghazimatin, A., Balalau, O., Saha Roy, R., & Weikum, G. (2019). PRINCE: Provider-side interpretability with counterfactual explanations in recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (pp. 196-204). https://doi.org/10.1145/3289600.3290982

Gomez, O., Holter, S., Yuan, J., & Bertini, E. (2020). ViCE: Visual counterfactual explanations for machine learning models. In *Proceedings of the 25th International Conference on Intelligent User Interfaces* (pp. 531-535). https://doi.org/10.1145/3377325.3377516

Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., & Lee, S. (2019). Counterfactual visual explanations. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 2376-2384).

Grinberg, M. (2017). *The Flask Mega-Tutorial: A Working Guide to Web Application Development with Flask and Python*. Leanpub.

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.

Gui, Q., Zhang, Y., & Li, S. (2019). Predicting flight delays using neural networks and random forests. *Journal of Air Transport Management, 75*, 80-89. https://doi.org/10.1016/j.jairtraman.2018.12.010

Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., & Turini, F. (2019). Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, 34(6), 14-23. https://doi.org/10.1109/MIS.2019.2957223

Haynes, S. N., Richard, D. C., & Kubany, E. S. (1995). Content validity in psychological assessment: A functional approach to concepts and methods. *Psychological Assessment, 7*(3), 238-247. https://doi.org/10.1037/1040-3590.7.3.238

Holloway, S. (2016). *Straight and Level: Practical Airline Economics* (4th ed.). Routledge.

Hao, L., & Hansen, M. (2018). Air traffic flow management under uncertainty: Flight-specific delay estimation. *Transportation Research Part C: Emerging Technologies, 86*, 151-175. https://doi.org/10.1016/j.trc.2017.11.020

Hashemi, M., & Fathi, A. (2020). PermuteAttack: Counterfactual explanation of machine learning credit scorecards. *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*, 1902-1911. https://doi.org/10.1109/BigData50022.2020.9378225

Hauke, J., & Kossowski, T. (2011). Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data. *Quaestiones Geographicae, 30*(2), 87-93. https://doi.org/10.2478/v10117-011-0021-1

Holovaty, A., & Kaplan-Moss, J. (2005). *The Definitive Guide to Django: Web Development Done Right*. Apress.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd ed.). John Wiley & Sons.

Huang, X., & Marques-Silva, J. (2024). On the failings of Shapley values for explainability. *International Journal of Approximate Reasoning, 160*, 97-108. https://doi.org/10.1016/j.ijar.2023.109112

IATA Airline A-CDM Coordination Group. (2018). *Airport – Collaborative Decision Making (A-CDM): IATA Recommendations*. International Air Transport Association.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer.

Jones, R. (2019). *Transparency and Accountability in Research*. Academic Press.

Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., & Ghosh, J. (2019). Towards realistic individual recourse and actionable explanations in black-box decision making systems. *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 263-270). https://doi.org/10.1145/3306618.3314281

Jung, H.-G., Kang, S.-H., Kim, H.-D., Won, D.-O., & Lee, S.-W. (2022). Counterfactual explanation based on gradual construction for deep networks. *Pattern Recognition, 132*, 108958. https://doi.org/10.1016/j.patcog.2022.108958

Kanamori, K., Takagi, T., Kobayashi, K., & Arimura, H. (2020). DACE: Distribution-aware counterfactual explanation by mixed-integer linear optimization. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (pp. 2857-2864). https://doi.org/10.24963/ijcai.2020/395

Kanamori, K., Takagi, T., Kobayashi, K., Ike, Y., Uemura, K., & Arimura, H. (2021). Distribution-aware counterfactual explanation by mixed-integer linear optimization. *Transactions of the Japanese Society for Artificial Intelligence, 36*(6), C-L44_1. https://doi.org/10.1527/tjsai.36-6_C-L44

Karimi, I., Barthe, G., Schölkopf, B., & Valera, I. (2020). Model-agnostic counterfactual explanations for consequential decisions. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics* (pp. 895-905).

Karimi, A., Schölkopf, B., & Valera, I. (2021). Algorithmic recourse: From counterfactual explanations to interventions. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 353-362). https://doi.org/10.1145/3442188.3445899

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems, 30*.

Keane, M. T., & Smyth, B. (2020). Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI). In I. Watson & R. Weber (Eds.), *Case-Based Reasoning Research and Development. ICCBR 2020. Lecture Notes in Computer Science* (Vol. 12311). Springer, Cham. https://doi.org/10.1007/978-3-030-58342-2_11

Kenny, E., & Keane, M. (2021). On generating plausible counterfactual and semi-factual explanations for deep learning. *Proceedings of the AAAI Conference on Artificial Intelligence, 35*, 11575-11585.

Khanmohammadi, S., Chou, C. A., Lewis, H. W., & Elias, D. (2014). A systems approach for scheduling aircraft landings in JFK airport using ANFIS. In *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1578-1585). https://doi.org/10.1109/FUZZ-IEEE.2014.6891597

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., ... & Willing, C. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (pp. 87-90). IOS Press.

Kovalev, M., Utkin, L., Coolen, F., & Konstantinov, A. (2021). Counterfactual explanation of machine learning survival models. *Informatica, 32*(4), 817-847. https://doi.org/10.15388/21-INFOR468

Kutner, M. H., Nachtsheim, C. J., & Neter, J. (2004). *Applied Linear Regression Models*. McGraw-Hill/Irwin.

Lash, M. T., Lin, Q., Street, W. N., Robinson, J. G., & Ohlmann, J. W. (2016). Generalized inverse classification. In *Proceedings of the 2016 SIAM International Conference on Data Mining* (pp. 162-170). https://doi.org/10.1137/1.9781611974348.19

Laugel, T., Lesot, M. J., Marsala, C., Renard, X., & Detyniecki, M. (2018). Comparison-based inverse classification for interpretability in machine learning. In *Proceedings of the 2018 International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (pp. 100-111). Springer, Cham. https://doi.org/10.1007/978-3-319-91479-4_9

Le, T., Wang, S., & Lee, D. (2019). GRACE: Generating concise and informative contrastive sample to explain neural network model's prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 238-248). https://doi.org/10.1145/3292500.3330937

Lee, Y., & Lee, J. (2016). Airport passenger service quality and passenger satisfaction: The case of Korean international airports. *International Journal of Services, Technology and Management, 22*(1-2), 1-15. https://doi.org/10.1504/IJSTM.2016.075692

Lewis-Beck, M. S., Bryman, A., & Liao, T. F. (2003). *The Sage Encyclopedia of Social Science Research Methods*. Sage.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News, 2*(3), 18-22.

Lim, T. S., & Loh, W. Y. (1996). A comparison of tests of equality of variances. *Computational Statistics & Data Analysis, 22*(3), 287-301. https://doi.org/10.1016/0167-9473(95)00044-5

Lulli, G., & Odoni, A. (2007). The European air traffic flow management problem. *Transportation Science, 41*(4), 431-443. https://doi.org/10.1287/trsc.1070.0214

McKight, P. E., & Najab, J. (2010). Kruskal-Wallis test. In *The Corsini Encyclopedia of Psychology* (pp. 1-1). Wiley. https://doi.org/10.1002/9780470479216.corpsy0491

Molnar, C. (2020). *Interpretable Machine Learning*. Leanpub.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis* (5th ed.). John Wiley & Sons.

Lipton, Z. C. (2018). The mythos of model interpretability. *Communications of the ACM, 61*(10), 36-43. https://doi.org/10.1145/3233231

Lu, X., Wang, Y., & Wu, C. (2016). Forecasting delays at airports due to propagation effects using k-nearest neighbor. *Journal of Air Transport Management, 56*, 120-128. https://doi.org/10.1016/j.jairtraman.2016.04.006

Lucic, A., Oosterhuis, H., Haned, H., & de Rijke, M. (2019). FOCUS: Flexible optimizable counterfactual explanations for tree ensembles. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 36, pp. 5313-5322). https://doi.org/10.1609/aaai.v36i5.20468

Lucic, A., Haned, H., & de Rijke, M. (2020). Why does my model fail? Contrastive local explanations for retail forecasting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1869-1872). Association for Computing Machinery. https://doi.org/10.1145/3351095.3372824

Lucic, A., ter Hoeve, M., Tolomei, G., de Rijke, M., & Silvestri, F. (2021). CF-GNNExplainer: Counterfactual explanations for graph neural networks. In *DLG-KDD'21: Deep Learning on Graphs, August 14-18, 2021, Online* (Article 3). ACM. https://doi.org/10.1145/1122445.1122456

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (pp. 4765-4774).

Mahajan, D., Tan, C., & Sharma, A. (2019). Preserving causal constraints in counterfactual explanations for machine learning classifiers. *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 287-292). https://doi.org/10.1145/3306618.3314274

Mamdouh, M., Al-Mansour, F., & Khalaf, A. (2020). Predicting required ground handling resources using SVM. *Journal of Air Transport Management, 85*, 101792. https://doi.org/10.1016/j.jairtraman.2020.101792

Martens, D., & Provost, F. (2014). Explaining data-driven document classifications. *MIS Quarterly, 38*(1), 73-100.

Mohammadi, K., Karimi, A., Barthe, G., & Valera, I. (2021). Scaling guarantees for nearest counterfactual explanations. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 306-316). https://doi.org/10.1145/3442188.3445921

Molnar, C. (2022). *Interpretable Machine Learning* (2nd ed.). Leanpub.

Moore, J. P., Hammerla, N. Y., & Watkins, C. (2019). Explaining deep learning models with constrained adversarial examples. *Proceedings of the 2019 International Joint Conference on Neural Networks* (pp. 1-8). https://doi.org/10.1109/IJCNN.2019.8852467

Mothilal, R. K., Mahajan, D., Tan, C., & Sharma, A. (2021). Towards unifying feature attribution and counterfactual explanations: Different means to the same end. In *AAAI/ACM Conference on AI, Ethics, and Society (AIES)* (pp. 261-271). https://doi.org/10.1145/3461702.3462533

Mothilal, R. K., Sharma, A., & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual examples. In *Proceedings of the 2020 ACM Conference on Fairness, Accountability, and Transparency* (pp. 607-617). https://doi.org/10.1145/3351095.3372830

Neufville de, R., & Odoni, A. (2016). *Airport Systems: Planning, Design, and Management* (2nd ed.). McGraw-Hill.

Numeroso, D., & Bacciu, D. (2021). MEG: Generating molecular counterfactual explanations for deep graph networks. In *2021 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). https://doi.org/10.1109/IJCNN52387.2021.9533940

Nunnally, J. C., & Bernstein, I. H. (1994). *Psychometric Theory* (3rd ed.). McGraw-Hill.

O'Brien, R. M. (2007). A caution regarding rules of thumb for variance inflation factors. *Quality & Quantity, 41*(5), 673-690. https://doi.org/10.1007/s11135-006-9018-6

Parmentier, A., & Vidal, T. (2021). Optimal counterfactual explanations in tree ensembles. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 252-261). https://doi.org/10.1145/3442188.3445905

Pawelczyk, M., Broelemann, K., & Kasneci, G. (2020). Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 33-44). https://doi.org/10.1145/3287560.3287563

Pearl, J. (2000). *Causality: Models, reasoning, and inference*. Cambridge University Press.

Peduzzi, P., Concato, J., Kemper, E., Holford, T. R., & Feinstein, A. R. (1996). A simulation study of the number of events per variable in logistic regression analysis. *Journal of Clinical Epidemiology, 49*(12), 1373-1379. https://doi.org/10.1016/S0895-4356(96)00236-3

Postorino, M. N., Mantecchini, L., Malandri, C., & Paganelli, F. (2020). A methodological framework to evaluate the impact of disruptions on airport turnaround operations: A case study. *Case Studies on Transport Policy, 8*(2), 429-439. https://doi.org/10.1016/j.cstp.2020.03.007

Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies, 2*(1), 37-63.

Poyiadzi, R., Sokol, K., Santos-Rodríguez, R., Bie, T. D., & Flach, P. A. (2019). FACE: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (pp. 344-350). https://doi.org/10.1145/3306618.3314289

Plotly. (2017). Introducing Dash. Plotly. Retrieved from https://medium.com/plotly/introducing-dash-5ecf7191b503

Plotly. (2021). *Dash User Guide & Documentation*. Plotly. Retrieved from https://dash.plotly.com/

Ramakrishnan, G., Lee, Y. C., & Albarghouthi, A. (2020). Synthesizing action sequences for modifying model decisions. In *Proceedings of the 2020 International Conference on Learning Representations* (pp. 1-14). https://doi.org/10.5555/3495724.3495899

Ramon, Y., Martens, D., Provost, F., & Evgeniou, T. (2020). A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: SEDC, LIME-C and SHAP-C. *Advances in Data Analysis and Classification, 14*, 801-819. https://doi.org/10.1007/s11634-020-00389-0

Rathi, S. (2019). Generating counterfactual and contrastive explanations using SHAP. *ArXiv*. https://doi.org/10.48550/arXiv.1906.09293

Rawal, K., & Lakkaraju, H. (2020). Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. *Proceedings of the 2020 International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2010.02599

Razali, N. M., & Wah, Y. B. (2011). Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests. *Journal of Statistical Modeling and Analytics, 2*(1), 21-33.

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

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135-1144). https://doi.org/10.1145/2939672.2939778

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. *Proceedings of the AAAI Conference on Artificial Intelligence, 32*(1). https://doi.org/10.1609/aaai.v32i1.11491

Ronacher, A. (2010). Flask documentation. Flask. Retrieved from https://flask.palletsprojects.com/en/1.1.x/

Rott, J., König, F., Häfke, H., Schmidt, M., Böhm, M., Kratsch, W., & Krcmar, H. (2023). Process mining for resilient airport operations: A case study of Munich Airport's turnaround process. *Journal of Air Transport Management, 112*, 102451. https://doi.org/10.1016/j.jairtraman.2023.102451

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206-215. https://doi.org/10.1038/s42256-019-0048-x

Russell, C. (2019). Efficient search for diverse coherent explanations. *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 20-28). https://doi.org/10.1145/3306618.3314273

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8*(4), e1249. https://doi.org/10.1002/widm.1249

Sanz, Á., Cano, J., & Fernández, B. (2021). Impact of weather conditions on airport arrival delay and throughput. *IOP Conference Series: Materials Science and Engineering, 1024*, 012107. https://doi.org/10.1088/1757-899X/1024/1/012107

Schiphol Airport. (2021). Schiphol Airport Collaborative Decision Making (A-CDM).

Schiphol Airport. (2024, February 6). *A-CDM Manual* (Version 1.0). Amsterdam Airport Schiphol.

Schiphol Airport. (n.d.). Deep turnaround. Schiphol Aviation Solutions. Retrieved April 15, 2024, from https://www.schiphol.nl/en/aviation-solutions/page/deep-turnaround/

Schiphol DeepTurn department. (2023, April 14). DeepTurn event definitions (Configuration Version 1.2) [Documentation].

Schiphol Group. (2023). *Annual report 2023*. Retrieved from https://www.schiphol.nl/nl/schiphol-group/pagina/jaarverslagen/

Schleich, M., Geng, Z., Zhang, Y., & Suciu, D. M. (2021). GeCo: Quality counterfactual explanations in real time. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 1-10). https://doi.org/10.1145/3442188.3445939

Schmider, E., Ziegler, M., Danay, E., Beyer, L., & Bühner, M. (2010). Is it really robust? Reinvestigating the robustness of ANOVA against violations of the normal distribution assumption. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences, 6*(4), 147-151. https://doi.org/10.1027/1614-2241/a000016

Schultz, M., Lorenz, S., Schmitz, R., & Delgado, L. (2018). Weather impact on airport performance. *Aerospace, 5*(4), 109. https://doi.org/10.3390/aerospace5040109

Schultz, M., Atkin, J., & Arnaldo Val, L. (2018). Airport slot allocation considering runway configuration changes. *Transportation Research Part E: Logistics and Transportation Review, 118*, 315-330. https://doi.org/10.1016/j.tre.2018.08.007

Yıldız, S., Aydemir, O., Memiş, A., & Varlı, S. (2022). A turnaround control system to automatically detect and monitor the time stamps of ground service actions in airports: A deep learning and computer vision-based approach. *Engineering Applications of Artificial Intelligence, 114*, 105032. https://doi.org/10.1016/j.engappai.2022.105032

Shadish, W. R., Cook, T. D., & Campbell, D. T. (2002). *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin.

Shakhnarovich, G., Darrell, T., & Indyk, P. (2008). Nearest-neighbor methods in learning and vision. *IEEE Transactions on Neural Networks, 19*(2), 377-389. https://doi.org/10.1109/TNN.2008.919409

Sharma, S., Henderson, J., & Ghosh, J. (2019). CERTIFAI: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 1-10). https://doi.org/10.1145/3306618.3314287

Sisson, P. (2017, November 20). Schiphol Airport's savvy design shows flying can actually be enjoyable. *Curbed*. Retrieved from https://archive.curbed.com/2017/11/20/16676482/airport-security-holiday-travel-schiphol-amsterdam

Smith, J. (2020). Principles of informed consent in research. *Journal of Ethical Research, 12*(3), 45-60.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management, 45*(4), 427-437. https://doi.org/10.1016/j.ipm.2009.03.002

Steckler, A., & McLeroy, K. R. (2008). The importance of external validity. *American Journal of Public Health, 98*(1), 9-10. https://doi.org/10.2105/AJPH.2007.126847

Strohmeier, M., Lenders, V., & Martinovic, I. (2018). Security of ADS-B: State of the art and beyond. *International Journal of Critical Infrastructure Protection, 19*, 34-46. https://doi.org/10.1016/j.ijcip.2017.05.002

Tolomei, G., Silvestri, F., Haines, A., & Lalmas, M. (2017). Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 465-474). https://doi.org/10.1145/3097983.3098039

Tretheway, M., & Andriulaitis, R. (2015). Airport regulation and the evolution of airline networks. *Journal of Air Transport Management, 42*, 243-253. https://doi.org/10.1016/j.jairtraman.2014.11.009

Trochim, W. M., & Donnelly, J. P. (2008). *Research Methods Knowledge Base*. Atomic Dog.

United Nations. (2015). *Transforming our world: The 2030 agenda for sustainable development*. Retrieved from https://sustainabledevelopment.un.org/post2015/transformingourworld

Ustun, B., Spangher, A., & Liu, Y. (2018). Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (pp. 10-20). https://doi.org/10.1145/3287560.3287566

Van den Broeck, J., Argeseanu Cunningham, S., Eeckels, R., & Herbst, K. (2005). Data cleaning: Detecting, diagnosing, and editing data abnormalities. *PLoS Medicine, 2*(10), e267. https://doi.org/10.1371/journal.pmed.0020267

Van Looveren, A., & Klaise, J. (2021). Interpretable counterfactual explanations guided by prototypes. In N. Oliver, F. Pérez-Cruz, S. Kramer, J. Read, & J. A. Lozano (Eds.), *Machine Learning and Knowledge Discovery in Databases. Research Track. ECML PKDD 2021. Lecture Notes in Computer Science* (Vol. 12976, pp. 301-316). Springer, Cham. https://doi.org/10.1007/978-3-030-86520-7_40

Verma, S., Dickerson, J., & Hines, K. (2020). Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*. Retrieved from https://arxiv.org/abs/2010.10596

Volt, J., Stojić, S., & Had, P. (2023). Possibilities of quantification of factors influencing the aircraft ground handling process and TOBT prediction. *Transportation Research Procedia, 75*, 68-76. https://doi.org/10.1016/j.trpro.2023.12.009

Waa, J. V., Robeer, M., Diggelen, J. V., Brinkhuis, M. J., & Neerincx, M. A. (2018). Contrastive explanations with local foil trees. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 303-309). https://doi.org/10.1145/3278721.3278729

Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology, 31*, 841-887.

Wang, P., & Vasconcelos, N. (2020). SCOUT: Self-aware discriminant counterfactual explanations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020* (pp. 8978-8987). https://doi.org/10.1109/CVPR42600.2020.00900

Wang, J., Yang, H., & Yang, D. (2018). Factors affecting passenger satisfaction in airport terminal: A case study of Shanghai Pudong Airport. *Journal of Air Transport Management, 74*, 22-30. https://doi.org/10.1016/j.jairtraman.2018.09.001

WG CDM. (2023). Deep Turnaround: Improving aircraft turnaround processes based on real-time insights for all stakeholders leveraging data and AI [PowerPoint slides].

White, A., & d'Avila Garcez, A. S. (2020). Measurable counterfactual local explanations for any classifier. In *Frontiers in Artificial Intelligence and Applications, 325* (pp. 2529-2535). https://doi.org/10.3233/FAIA200387

Wu, T., Ribeiro, M. T., Heer, J., & Weld, D. S. (2021). Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 6703-6719). https://doi.org/10.18653/v1/2021.emnlp-main.540

Yang, L., Kenny, E. M., Ng, T. L., Yang, Y., Smyth, B., & Dong, R. (2020). Generating plausible counterfactual explanations for deep transformers in financial text classification. In *Proceedings of the 28th International Conference on Computational Linguistics* (pp. 2843-2853). https://doi.org/10.18653/v1/2020.coling-main.258

Yildiz, B., Yildirim, U., & Aydin, O. (2022). Automating ground service monitoring during aircraft turnaround processes using deep learning and computer vision. *Journal of Air Transport Management, 95*, 102013. https://doi.org/10.1016/j.jairtraman.2021.102013

Zhang, X., & Lian, J. (2019). An analysis of the causes of aircraft turnaround delays in China's civil aviation industry. *Journal of Air Transport Management, 77*, 35-41. https://doi.org/10.1016/j.jairtraman.2019.03.005

Zhang, X., Solar-Lezama, A., & Singh, R. (2018). Interpreting neural network judgments via minimal, stable, and symbolic corrections. In *Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS 2018)* (pp. 4874-4885).

Zhang, Z., & Wang, L. (2021). *Machine Learning Techniques for Predictive Modeling*. Springer. https://doi.org/10.1007/978-3-030-67177-8

Zhou, Q., Li, X., & Ding, L. (2019). Predicting aircraft turnaround time: A machine learning approach. *Transportation Research Part E: Logistics and Transportation Review, 130*, 90-107. https://doi.org/10.1016/j.tre.2019.08.006

Zhao, Y. (2020). Fast real-time counterfactual explanations. Department of Civil and Environmental Engineering, University of California, Irvine, USA. Retrieved from https://doi.org/10.48550/arXiv.2006.12308

List of Appendices

Appendix 1: Extended literature review counterfactual XAI 61
Appendix 2: DeepTurn cameras 67
Appendix 3: Distribution of event features 68
Appendix 4: Scatterplots of numeric features versus DOBT 69
Appendix 5: Residual plots of numerical features 70
Appendix 6: Residual versus fitted values plots numeric features 71
Appendix 7: Linear regression plots numeric features 72
Appendix 8: Boxplots of categorical/discrete features 73
Appendix 9: Residual plots categorical features 75
Appendix 10: Confusion matrices 77
Appendix 11: Model feature importance 78
Appendix 12: Counterfactual explanations 79
Appendix 13: Dashboard 80

Appendix 1: Extended literature review counterfactual XAI

Dandle and Molnar (2023) describe counterfactual explanations as the smallest modification to the feature values that transforms a prediction into a predetermined output. With their logical understanding into the reasons behind specific outcomes, counterfactual explanations present a reliable method for understanding the predictions made by machine learning models. Counterfactual explanations demonstrate how modifications to input variables impact model predictions by modelling alternative scenarios in which important features are adjusted. This approach not only improves interpretability and transparency, but it also gives users the ability to investigate practical insights for better results. The Rashomon effect presents a problem, since there could be several possible counterfactuals for a single data point, which could cause interpretation to become unclear. However, counterfactual explanations are a useful tool for clearing up complex model behaviour and directing decision-making procedures due to their simplicity and clarity (Dandle & Molnar, 2023). In their exploration of post-hoc explanations within the domain of explainable Artificial Intelligence (XAI), Ferrario & Loi (2022) highlight the importance of counterfactual explanations as essential human-machine learning model interfaces that clarify model results and provide practical guidance on how to get different outcomes. Nonetheless, recent results from a systematic review by Chou et al. (2022) creates doubt on the counterfactual approach's ability to explain causality in XAI. They found that because the counterfactual approach's underlying framework was not developed in accordance with accepted causality theories, it is unclear how to establish causality. Baron (2023) suggests utilizing the classic causality approach to improve the counterfactual approach's ability to provide causal understanding in XAI, building on the work of Chou et al (2022). In other words, counterfactual XAI is a key tool to get a better understanding of feature importance in prediction models. Utilizing this is essential to get a better understanding of features influencing TOBT predictions and therefore utilized in this study. Furthermore, counterfactual XAI in TOBT or A-CDM in general has not been published in a research paper and therefore reveal a research gab. This further underlines the relevance of this research.

Various methodologies have been proposed to address the challenge of creating counterfactual explanations in machine learning models, offering insights into why certain predictions are made and what changes could lead to different outcomes. One approach, introduced by Cui et al. (2015), is Optimal Action Extraction (OAE), which is designed for models like ensembles. OAE uses Integer Linear Programming (ILP) with a solver to find optimal solutions, specifically aiming to provide explanations in instances where the desired outcome is not achieved. In a similar way, Wachter et al. (2017) presented WACH, which minimizes a loss function incorporating quadratic distance terms between desired and predicted outcomes, along with distance terms between input instances and counterfactuals. By balancing these terms, WACH aims to find counterfactual instances that are close to the original but lead to the desired outcome, thereby providing insightful explanations. Dhurandhar et al. (2018) proposed the Contrastive Explanation Method (CEM), which focuses on generating contrastive explanations by playing around with input instances to ensure a change in the predicted class. CEM uses an autoencoder to evaluate instances to known data, ensuring posibility in the generated explanations. By optimizing a loss function that encourages different predictions while minimizing reconstruction error, CEM generates counterfactual explanations that contrast with the original prediction, creating insights on the decision-making process of machine learning models. In contrast, Artelt (2019) introduced Counterfactual Explanations via CEML, providing a toolbox for producing counterfactual explanations across different types of black-box models. While not formally presented in a paper, CEML offers solutions based on optimization designed for different model types, thereby contributing to the generation of insightful explanations in diverse scenarios. Additionally, Chapman-Rounds et al. (2019) proposed Explanation by Minimal Adversarial Perturbation (EMAP), also known as FIMAP, which is a model and data-agnostic approach. By training a model to mimic the behaviour

of the black-box model, EMAP identifies minimal perturbations needed to change the model's prediction while ensuring validity, thereby offering interpretable explanations.

Dhurandhar et al. (2019) extended CEM to Model Agnostic Contrastive Explanations Method (MACEM), focusing on providing model-agnostic contrastive explanations by estimating gradients instead of directly calculating them. MACEM offers an approach to generating contrastive explanations across different model types, aiming to highlight actionable insights by identifying perturbations that lead to different model predictions. Furthermore, Downs et al. (2020) developed Counterfactual Reasoning Using Disentangled Subspace (CRUDS), which extends the concept of counterfactual explanations by using Conditional Subspace VAE (CSVAE) to generate explanations. CRUDS identifies relevant features for generating counterfactuals while filtering out irrelevant ones, thereby providing interpretable explanations by focusing on uncomplicated feature representations and causal relationships. Joshi et al. (2019) proposed Recourse Exploration via Variational Inference and Search (REVISE), designed to provide counterfactual explanations that account for actionability and causality. REVISE identifies counterfactual instances that not only change the predicted outcome but also respect causal relationships, thereby offering actionable results in complex decision-making scenarios. Lucic et al. (2019) introduced Flexible Optimizable Counterfactual Explanations for Tree Ensembles (FOCUS), extending counterfactual explanation methods to non-differentiable tree ensembles by leveraging probabilistic model approximations. FOCUS aims to generate counterfactual explanations for tree-based models by replacing terms in the loss function with differentiable approximations, providing interpretable insights into model decisions. The Example-Based Counterfactual explainer (EBCF) introduced by Mahajan et al. (2019), uses a variational autoencoder (VAE) to regularize the generation of counterfactuals. It uses a fine-tuning phase that adjusts model parameters to make sure feasibility through causality, adding a regularization term to the loss function to enforce plausibility by checking known causal relationships. EBCF uses the Adam optimizer and handles categorical features with one-hot encoded vectors, controlling their feasibility by the VAE. Another method, the efficient search for Diverse Coherent Explanations (DCE) proposed by Russell (2019), builds upon WACH to find diverse counterfactuals. DCE formulates the problem as a linear program, with features treated as integers through one-hot encoding. Plausibility is ensured by a set of linear constraints, and diversity is induced through additional constraints that reduce the possible values with respect to already generated counterfactuals.

Ustun et al. (2019) introduced the Actionable Recourse (ACTREC) method, which addresses the problem of actionability in counterfactual explanations by constraining generated counterfactuals to make sure that variations do not change unchangeable features. ACTREC formulates the problem through mixed integer programming, with constraints on actionable features ensuring that valid solutions remain actionable. While designed for tabular data and differentiable classifiers, ACTREC can handle categorical features. In contrast, Kanamori et al. (2020) proposed the Distribution-Aware Counterfactual Explanation method (DACE), based on mixed integer linear optimization. DACE uses a loss function incorporating the Mahalanobis distance and the Local Outlier Factor (LOF) to evaluate the plausibility of candidate counterfactuals. By simultaneously minimizing distance and maintaining plausibility, DACE provides explanations for linear classifiers and tree ensembles, handling categorical features with one-hot encoding. Karimi et al. (2020) introduced the Model-Agnostic Counterfactual Explanation (MACE) approach, which operates on diverse tabular data with any given distance function. MACE maps the problem into a sequence of satisfiability problems, expressing black-box models, distance functions, and constraints as logic formula. By using satisfiability modulo theories solvers, MACE generates counterfactual explanations, facilitating interpretable insights into model predictions. Moreover, Mothilal et al. (2020) proposed Diverse Counterfactual Explanations (DICE), which solves an optimization problem with various constraints to ensure feasibility and diversity in the generated counterfactuals. DICE encourages actionability and

feasibility by penalizing solutions formed by counterfactuals that are too similar, thereby promoting diversity. It handles categorical features through one-hot encoding and adopts the Adam optimizer.

Furthermore, Pawelczyk et al. (2020) presented the Counterfactual Conditional Heterogeneous Autoencoder (C-HVAE), a model-agnostic explainer for tabular data that utilizes an autoencoder for modelling heterogeneous data. C-HVAE does not require a distance function in the real input space, relying on the autoencoder to measure distances in the latent space and guide the search for counterfactuals. Additionally, Ramakrishnan et al. (2020) introduced SYNTH, a method for synthesizing action sequences to modify model decisions, combining search-based program synthesis and optimization-based adversarial example generation. SYNTH constructs action sequences over a domain-specific set of actions, enabling changes in differentiable black-box and categorical data, and it is designed for tabular data, though tested on simple images. Moreover, Rawal and Lakkaraju (2020) proposed the Actionable REcourse Summaries approach (ARES), which constructs global counterfactual explanations providing interpretable summaries of recourses for entire reference populations. ARES optimizes for validity and interpretability while minimizing changes with respect to the reference population, with initial rules provided by the user or extracted using Apriori. Wang and Vasconcelos (2020) developed the Self-aware disCriminant cOUnterfactual explanaTion method (SCOUT) for generating discriminant counterfactual explanations for image classifiers. SCOUT computes explanations by identifying informative pixels for the predicted class and uninformative pixels for other classes, obtained through an optimization process. Moreover, Zhao (2020) proposed FRACE (Fast ReAl-time Counterfactual Explanation), an explainer for neural network classifiers for images. FRACE uses a neural network architecture and minimizes a loss function accounting for validity and minimal perturbation, generating counterfactuals through a residual generator and accounting for plausibility through adversarial training. Furthermore, Carreira-Perpiñán and Hada (2021) presented CEODT, a Counterfactual Explanation method for Oblique Decision Trees, specifically designed for classification trees, including both traditional axis-aligned and oblique trees. CEODT computes exact solutions within each leaf region, optimizing a mixed integer optimization problem to find feasible counterfactuals.

Cheng et al. (2021) introduced DECE, an interactive Decision Explorer with Counterfactual Explanation that provides explanations through a visualization system, retrieving multiple counterfactuals by optimizing a loss function for validity, distance minimality, number of changes, and diversity, with the possibility of specifying feature constraints for actionability. Moreover, Mohammadi et al. (2021) presented SGNCE, a counterfactual explanation approach specifically designed for neural networks, providing guarantees for minimality and coverage of returned counterfactuals through mixed-integer programming. SGNCE also ensures plausibility and actionability in generated counterfactuals. Additionally, Parmentier and Vidal (2021) proposed OCEAN, an Optimal Counterfactual ExplAiNer for tree ensembles, utilizing efficient mixed-integer programming to search for counterfactuals and account for both plausibility and actionability. Kanamori et al. (2021) introduced ORDCE, the Ordered Counterfactual Explanation method, which accounts for asymmetric interaction among features by calculating a loss function that depends on the order of changing features, aiming to return counterfactuals with not only feature values but also the order in which they should be altered. Moreover, Karimi et al. (2021b) presented ALGREC, which uses causal reasoning to find recourse through minimal interventions, leveraging known causal models to ensure valid counterfactuals respecting causality. Furthermore, Kenny and Keane (2021) illustrated the PIECE method for generating contrastive explanations for CNNs working on image data, identifying exceptional features, and modifying them to generate plausible counterfactuals, leveraging a GAN to generate counterfactual images. Van Looveren and Klaise (2021) proposed CEGP, a method for Counterfactual Explanations Guided by Prototypes, which employs a loss function based on prototypes to guide perturbations toward counterfactuals that respect class distributions, accounting for categorical features by inferring distances between categories. In a different domain, Wu et al. (2021) introduced POLYJUICE, a general-purpose

counterfactual generator for textual data. POLYJUICE returns a diverse set of realistic textual counterfactuals, ensuring similarity and minimality while guaranteeing grammatical correctness and endogenous counterfactuals.

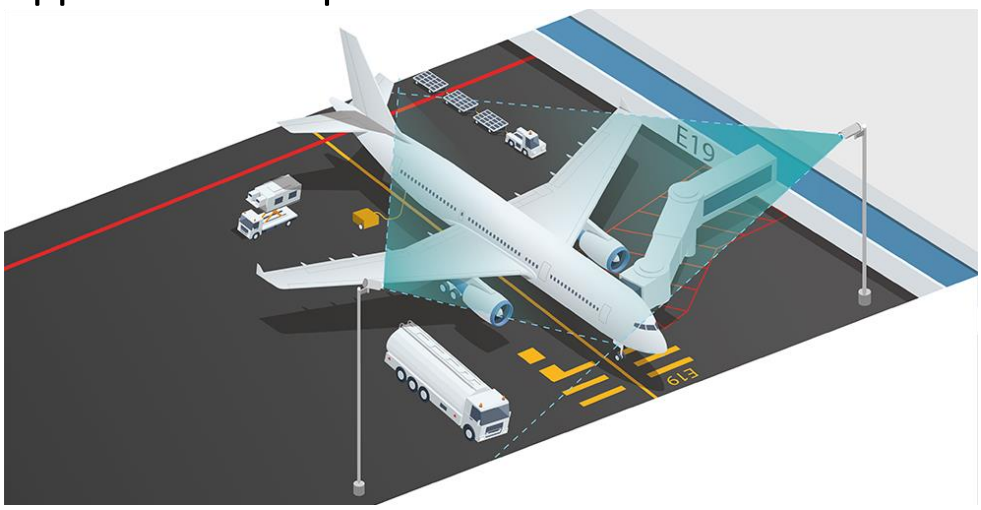
Additionally, NNCE (Nearest-Neighbor Counterfactual Explainer), introduced by Shakhnarovich et al. (2008), selects instances most like the input instance and with different labels, offering simplicity and interpretability in generating counterfactuals for tabular data. CBCE (Case-Based Counterfactual Explainer), refined by Keane & Smyth (2020) from NNCE, uses explanation cases to guide the generation of counterfactuals, providing adaptability and customization in explanations. FACE (Feasible and Actionable Counterfactual Explanations), presented by Poyiadzi et al. (2020), focuses on uncovering feasible paths for generating counterfactuals, ensuring coherence with the input data distribution, and promoting actionability in the generated explanations. NICE (Nearest Instance Counterfactual Explainer), proposed by Brughmans & Martens (2021), offers a versatile approach to generate diverse and plausible counterfactuals for tabular data, considering various trade-offs in the generated explanations. Shakhnarovich et al. (2008) introduced TBCE (Tree-Based Counterfactual Explainer), a method that using surrogate decision trees trained on reference datasets to mimic classifier behaviour. By utilizing decision tree paths leading to different predictions, TBCE offers customizable counterfactuals for tabular data, ensuring interpretability and actionability in its explanations. Tolomei et al. (2017) proposed FT (Feature Tweaking) as a method to understand which features of a given instance should be modified to alter predictions of tree-based ensembles. FT ensures validity across all trees in the ensemble, providing actionable recommendations for transforming instances while covering the entire feature domain. Guidotti et al. (2019) presented LORE (LOcal Rule-based Explainer), a method that provides rule-based explanations for tabular data. By generating synthetic neighbours and training decision trees on them, LORE extracts factual and counterfactual rules, enabling interpretable and actionable insights into model behaviour.

Waa et al. (2019) introduced FOILTREE, which generates contrastive explanations using local surrogate trees. By focusing on differences between decision paths leading to factual and foil outcomes, FOILTREE offers interpretable explanations for model predictions. Fernández et al. (2020) proposed RF-OCSE (Random Forest Optimal Counterfactual Set Extractor) to extract counterfactual sets from Random Forests. RF-OCSE ensures consensus among individual tree predictors by converting Random Forests into single decision trees, offering actionable counterfactual explanations for tabular data. Ribeiro et al. (2018) introduced ANCHOR, aiming to retrieve explanations as sufficient conditions for classification. ANCHOR offers an alternative approach to counterfactual explanations by focusing on defining conditions for predictions rather than changes to input instances. Ghazimatin et al. (2020) presented PRINCE, a counterfactual explainer for recommender systems. By describing possible user interactions, PRINCE provides explanations as minimal sets of actions on heterogeneous information networks, offering actionable insights into model recommendations. Kovalev et al. (2021) proposed SURV-CF to address the challenges of counterfactual explanation in machine learning survival models. SURV-CF offers interpretable explanations by reducing the problem to a optimization problem with linear constraints and employing Particle Swarm Optimization for survival model predictions. Ates et al. (2021) introduced COMTE, a counterfactual explanation method for multivariate time series data. COMTE provides explanations by minimizing a loss function while ensuring validity and similarity in generated counterfactuals, offering insights into model behaviour. Lucic et al. (2021) proposed CF-GNNEXPLAINER, a counterfactual explainer designed for classifiers operating on graphs. By identifying minimal perturbations to the graph structure that change predictions, CF-GNNEXPLAINER offers interpretable explanations for graph-based models. Numeroso and Bacciu (2021) introduced MEG (Molecular Explanation Generator), providing counterfactual explanations for graph neural networks by generating valid compounds with high structural similarity and different predicted properties. MEG ensures validity while exploring diverse counterfactual explanations through reinforcement learning.

Counterfactual XAI	Paper	Advantage
OAE	Cui et al. (2015)	ILP ensures optimal solutions
WACH	Wachter et al. (2017)	Balances quadratic distance terms for close to original counterfactuals
CEM	Dhurandhar et al. (2018)	Uses autoencoder for plausible explanations
CEML	Artelt (2019)	Offers optimization-based solutions for diverse model types
EMAP	Chapman-Rounds et al. (2019)	Provides minimal disruptions for interpretable explanations
MACEM	Dhurandhar et al. (2019)	Estimation of gradients for scalability
CRUDS	Downs et al. (2020)	Focuses on relevant feature representations
REVISE	Joshi et al. (2019)	Actionable insights respecting causal relationships
FOCUS	Lucic et al. (2019)	Provides insights into non-differentiable models
EBCF	Mahajan et al. (2019)	Ensures feasibility through causality in counterfactual generation
DCE	Russell (2019)	Formulates linear program for coherent counterfactuals
ACTREC	Ustun et al. (2019)	Handles actionability constraints with mixed integer programming
DACE	Kanamori et al. (2020)	Evaluates plausibility through novel loss function
MACE	Karimi et al. (2020)	Facilitates interpretable insights using SMT solvers
DICE	Mothilal et al. (2020)	Promotes diversity in generated counterfactuals
C-HVAE	Pawelczyk et al. (2020)	Utilizes latent space for counterfactual search
SYNTH	Ramakrishnan et al. (2020)	Constructs action sequences for model decision modification
ARES	Rawal & Lakkaraju (2020)	Provides interpretable summaries for entire populations
SCOUT	Wang & Vasconcelos (2020)	Identifies informative pixels for discriminant explanations
FRACE	Zhao (2020)	Minimizes perturbation for fast explanations
CEODT	Carreira-Perpiñán & Hada (2021)	Computes exact solutions for oblique decision trees
DECE	Cheng et al. (2021)	Provides interactive explanations with specified feature constraints
SGNCE	Mohammadi et al. (2021)	Guarantees minimality and coverage in returned counterfactuals
OCEAN	Parmentier & Vidal (2021)	Accounts for plausibility and actionability in tree ensembles
ORDCE	Kanamori et al. (2021)	Returns counterfactuals with ordered feature change
ALGRE	Karimi et al. (2021b)	Embeds casual reasoning for valid counterfactuals
PIECE	Kenny & Keane (2021)	Generates contrastive explanations for CNN's
CEGP	Van Looveren & Klaise (2021)	Guides disruptions based on prototypes for class distribution
POLYJUICE	Wu et al. (2021)	Generates textual counterfactuals with grammatical correctness
SEDC	Martens & Provost (2014)	Model-agnostic, tailored for textual data
GIC	Lash et al. (2017)	Handles actionable features and causal relationships
GSG	Laugel et al. (2018)	Generates diverse counterfactuals for image data
POLARIS	Zhang et al. (2018)	Versatile, stable, and model agnostic
CVE	Goyal et al. (2019)	Provides visually plausible explanations for image classifiers
CADEX	Moore et al. (2019)	Considers sparsity and plausibility for tabular data
CFSHAP	Rathi (2019)	Heuristic approach based on SHAP for tabular data
CERTIFAI	Sharma et al. (2019)	Utilizes a genetic algorithm for robust explanations
PCATTGAN	Arrieta & Ser (2020)	Relies on adversarial examples for plausible explanations
MOC	Dandl et al. (2020)	Generates diverse counterfactuals with different trade-offs
VICE	Gomez et al. (2020)	Focuses on visual explanations with minimal changes
PERMUTEATTACK	Hashemi & Fathi (2020)	Model-agnostic approach based on adversarial perturbation
GRACON	Kang et al. (2020)	Considers internal characteristics of deep neural networks
GRACE	Le et al. (2020)	Heuristic contrastive sample generation for tabular datasets
MCBRP	Lucic et al. (2020)	Exploits Monte Carlo simulation for unusual properties identification
LIME-C/SHAP-C	Ramon et al. (2020)	Adaptations of LIME/SHAP for counterfactual explanations
CLEAR	White & d'Avila Garcez (2020)	Provides local explanations via regression coefficients
PCIG	Yang et al. (2020)	Generates grammatically plausible counterfactuals for textual data
GECO	Schleich et al. (2021)	Utilizes PLAF constraints for diverse and plausible counterfactuals
NNCE	Shakhnarovich et al. (2008)	Simplicity and interpretability for tabular data
CBCE	Keane & Smyth (2020)	Adaptability and customization in explanations
FACE	Poyiadzi et al. (2020)	Ensures coherence and actionability in explanations
NICE	Brughmans & Martens (2021)	Versatile approach with diverse trade-offs
TBCE	Shakhnarovich et al. (2008)	Uses explanations with surrogate decision trees
FT	Tolomei et al. (2017)	Maintains validity across ensemble trees
LORE	Guidotti et al. (2019)	Derives rule-based insights from synthetic neighbours
FOILTREE	Van Der Waa et al. (2019)	Focuses on differences for interpretability
OCSE	Fernández et al. (2020)	Converts forests for consensus explanations
ANCHOR	Ribeiro et al. (2018)	Provides "sufficient" conditions for classification
PRINCE	Ghazimatin et al. (2020)	Offers minimal action sets for recommendations
SURV-CF	Kovalev et al. (2021)	Utilizes optimization for survival model insights
COMTE	Ates et al. (2021)	Balances accuracy and validity in time series

CF-GNNEXPLAINER	Lucic et al. (2021)	Identifies minimal graph perturbations
MEG	Numeroso & Bacciu (2021)	Generates diverse counterfactual compounds

Appendix 2: DeepTurn cameras

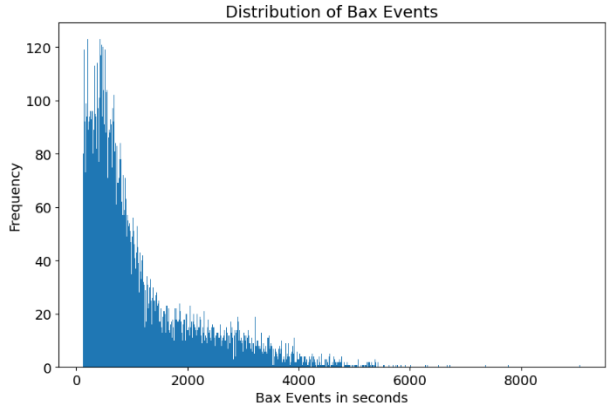


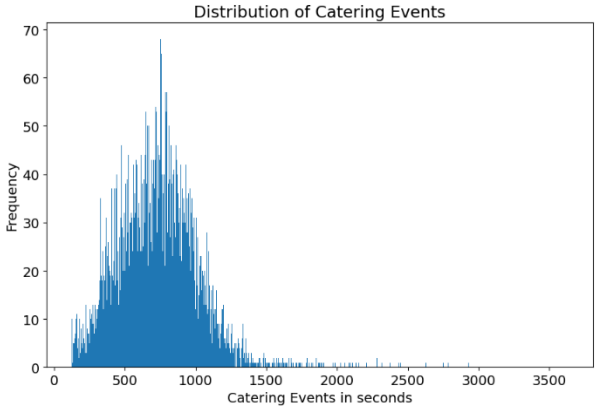
Schiphol DeepTurn department (2023)

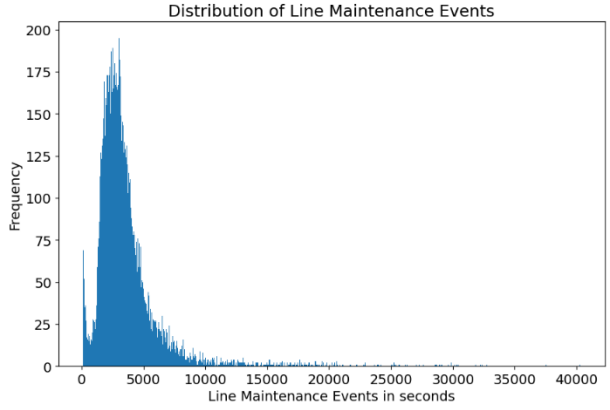
Schiphol DeepTurn department (2023)

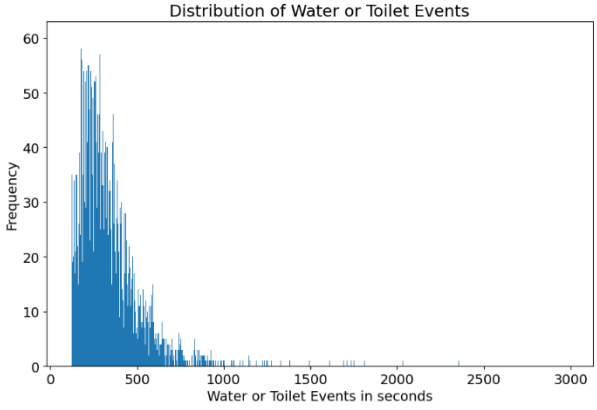
Schiphol DeepTurn department (2023)

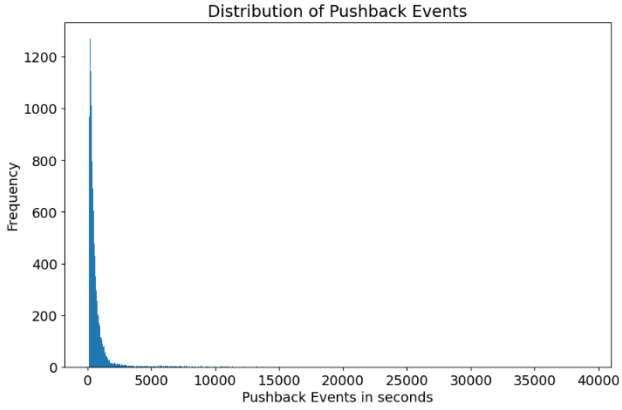
Appendix 3: Distribution of event features

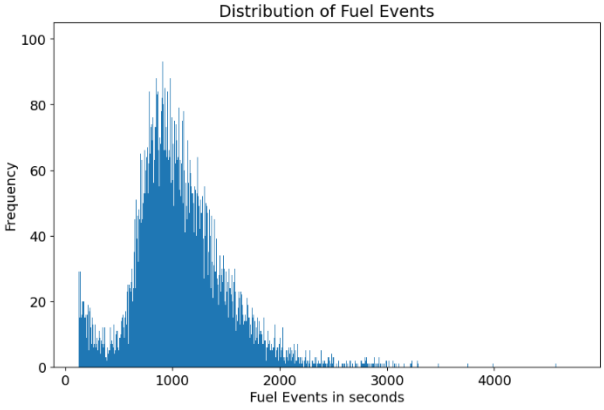


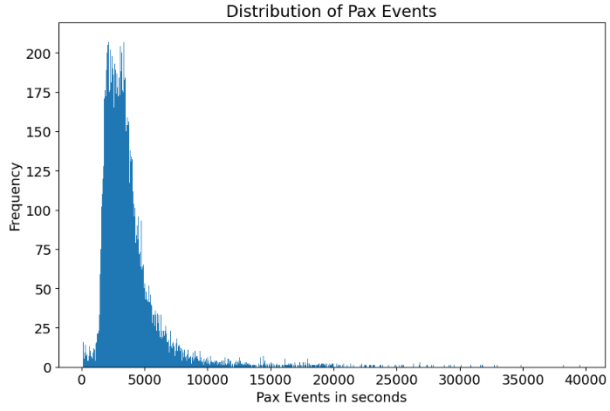




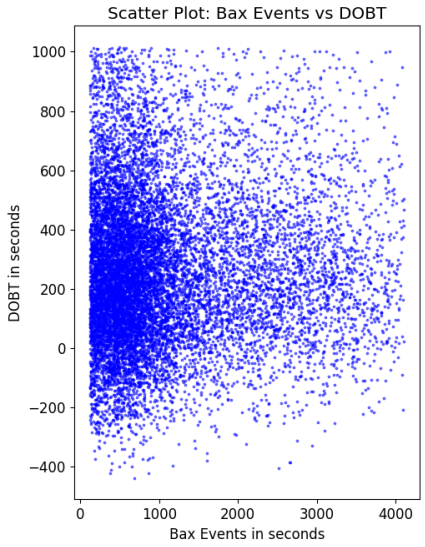


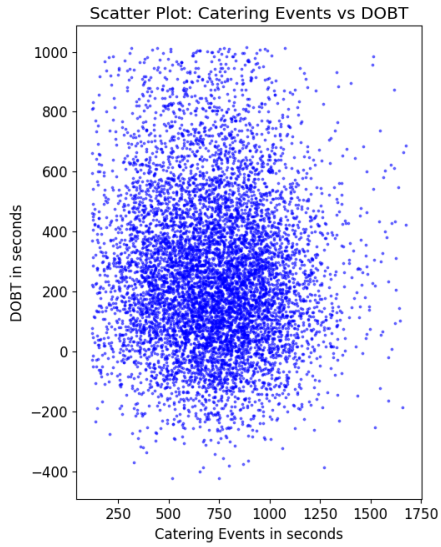


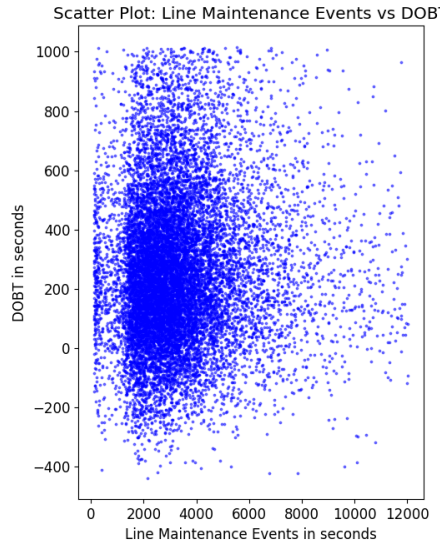


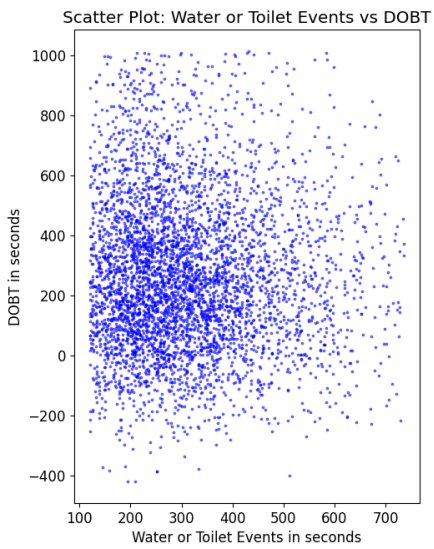


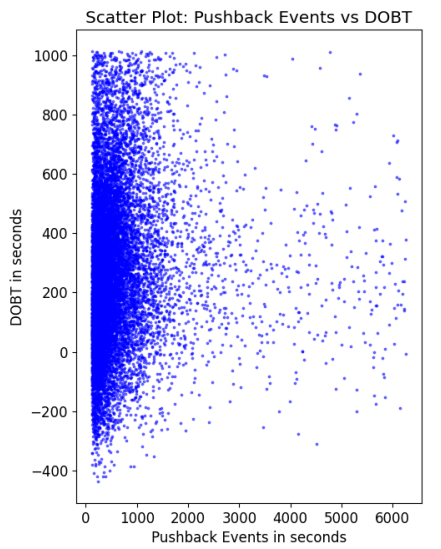
Appendix 4: Scatterplots of numeric features versus DOBT

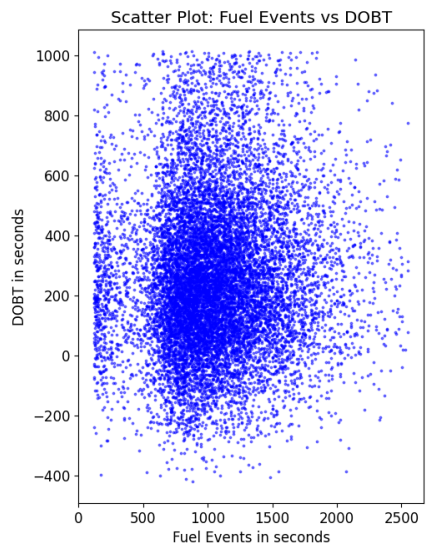


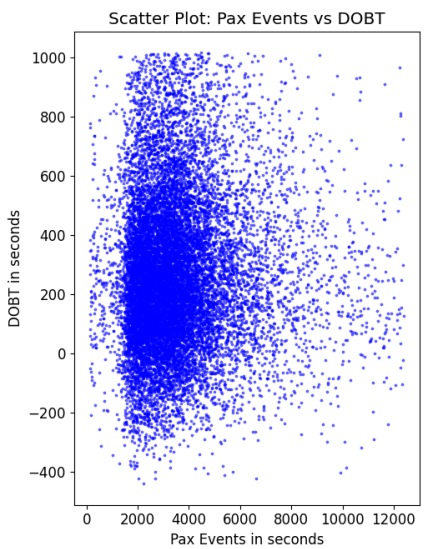




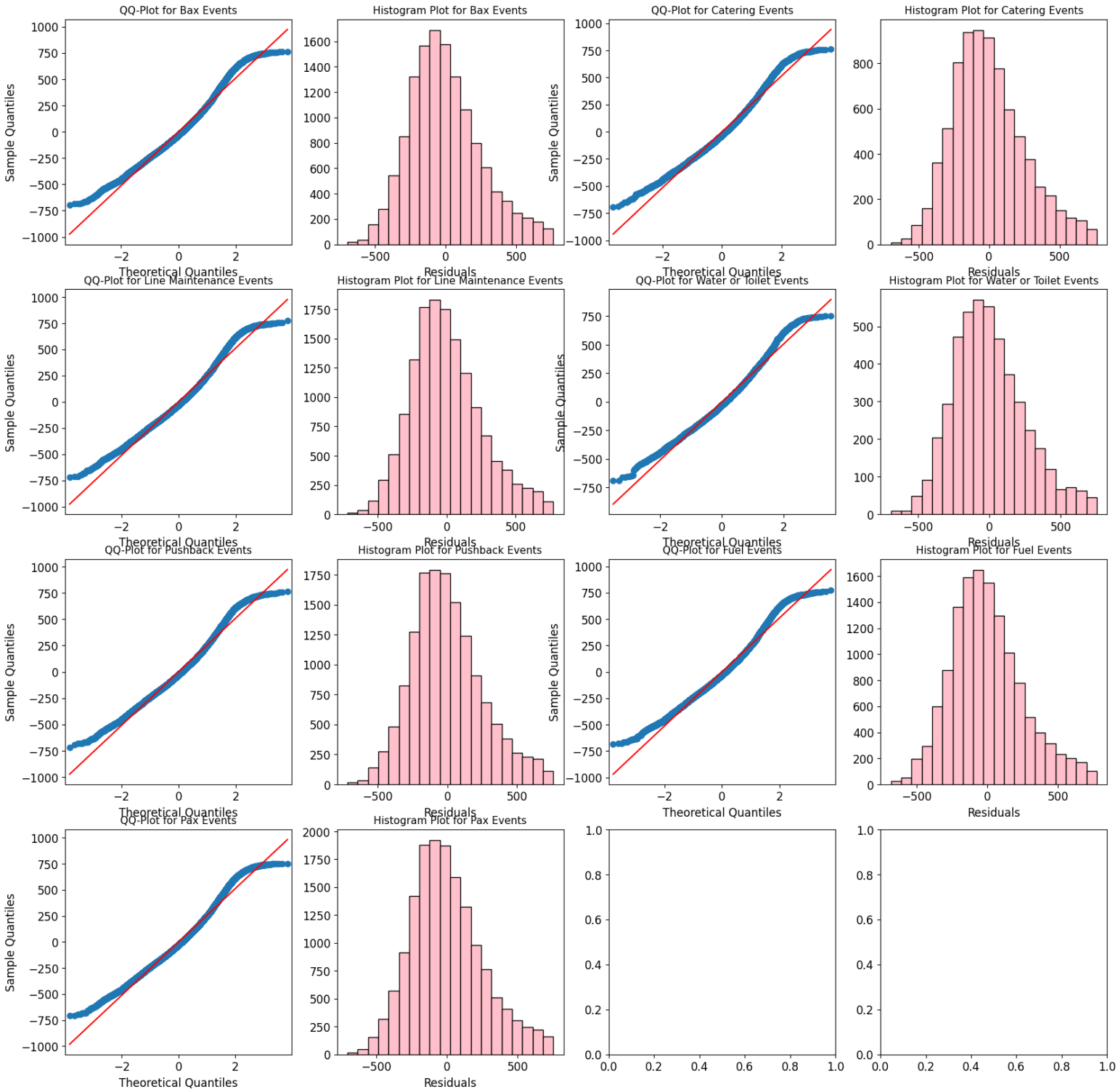




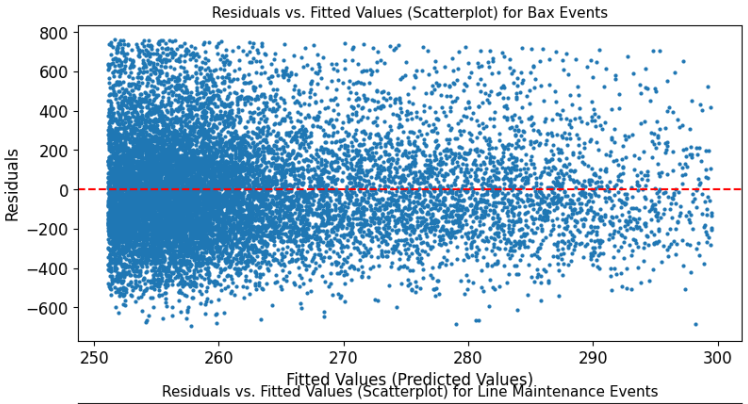


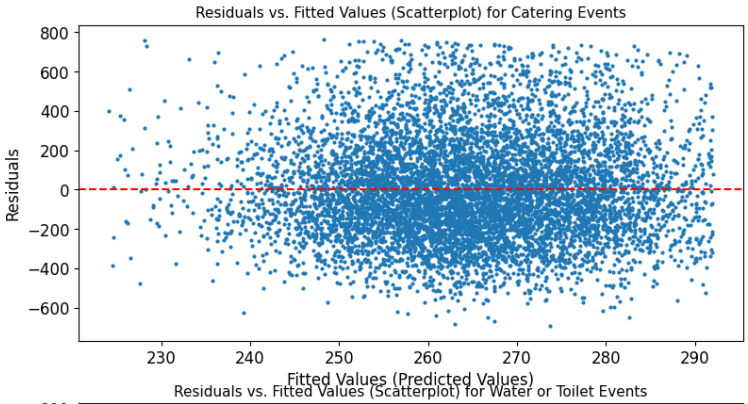


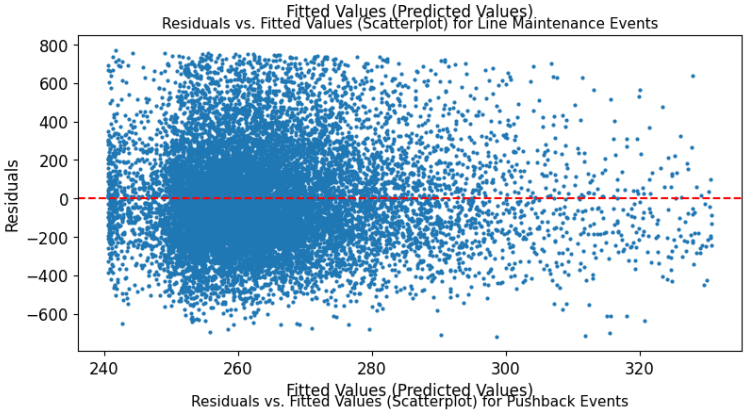
Appendix 5: Residual plots of numerical features

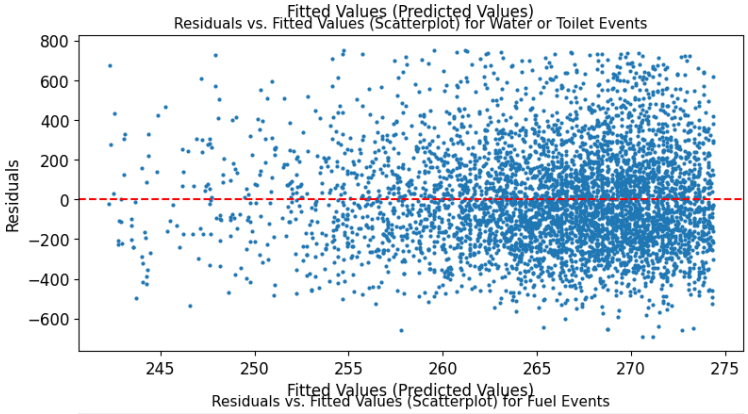


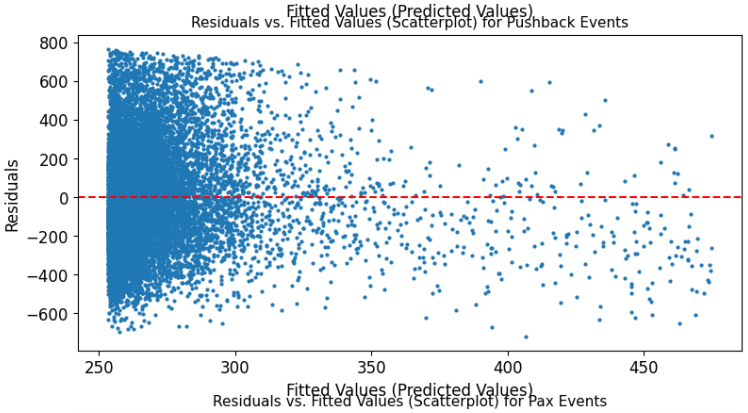
Appendix 6: Residual versus fitted values plots numeric features

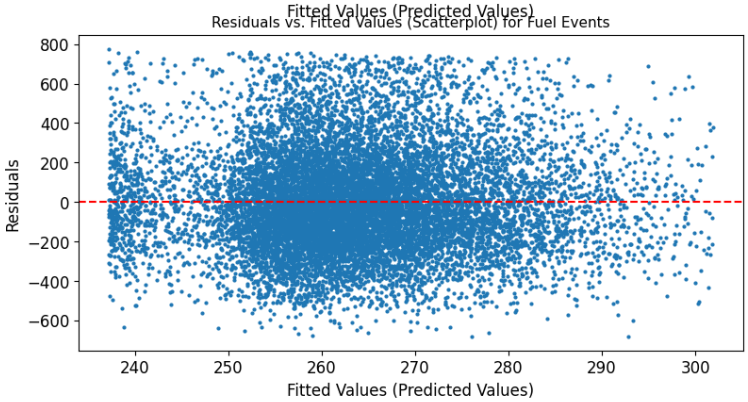


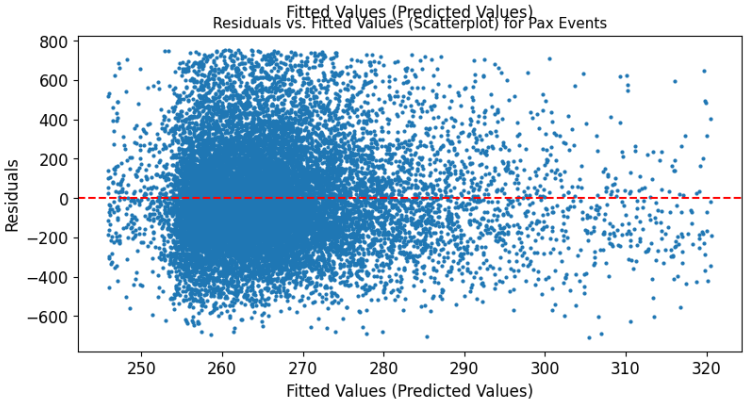


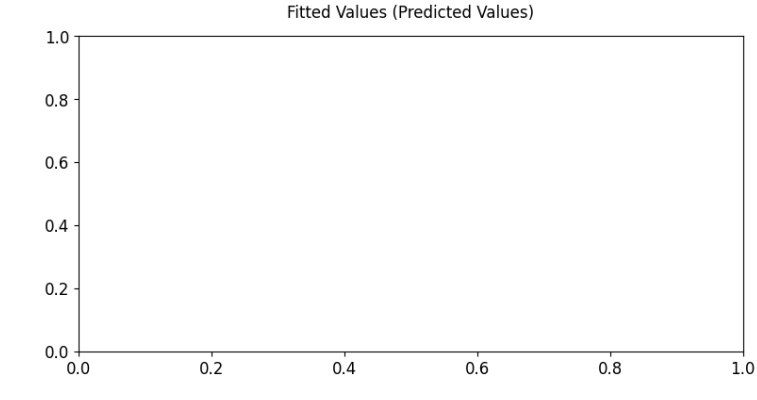




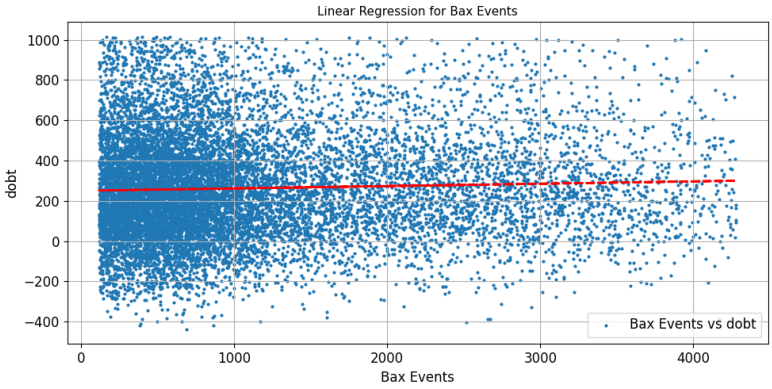


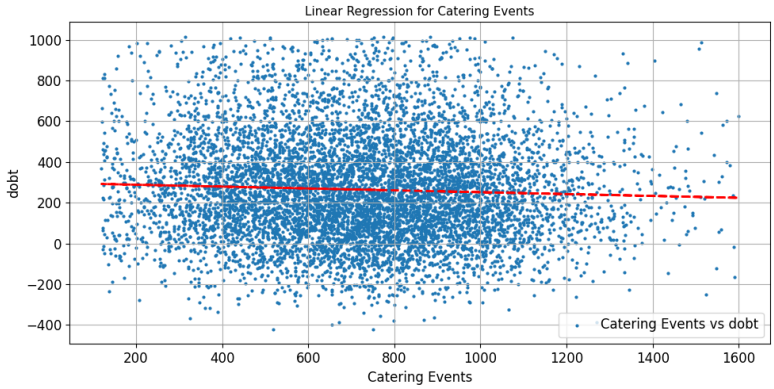


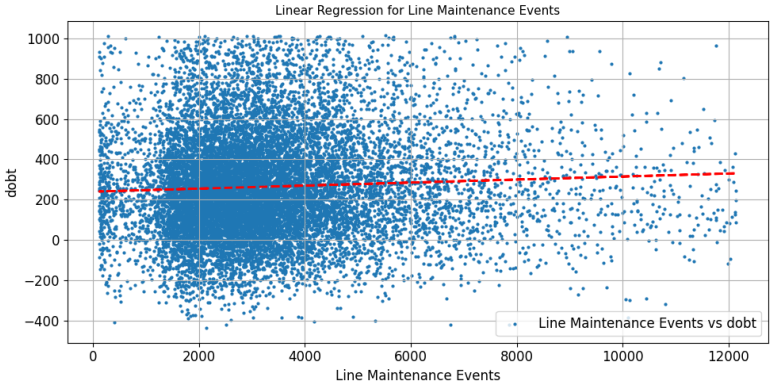


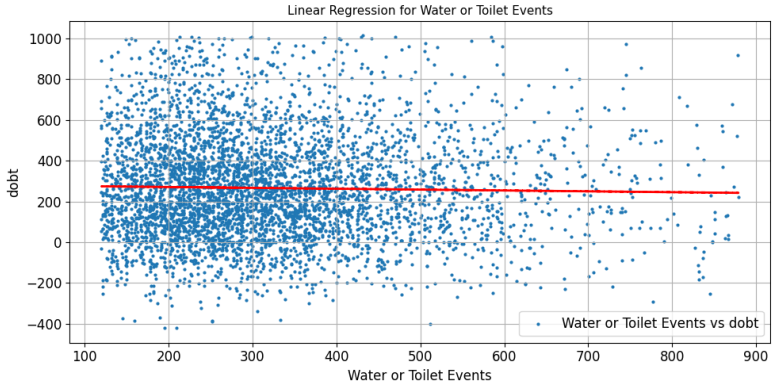


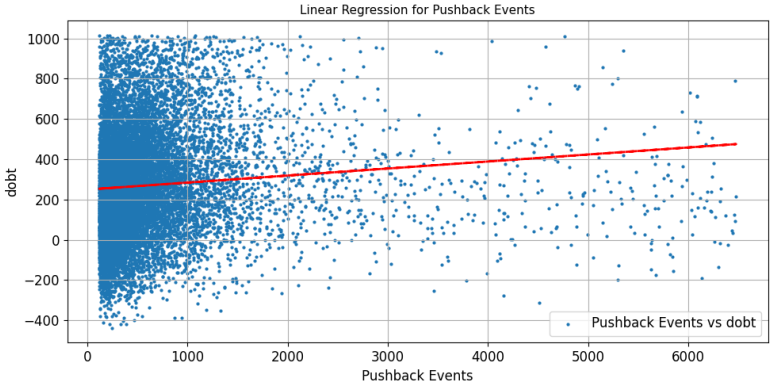
Appendix 7: Linear regression plots numeric features

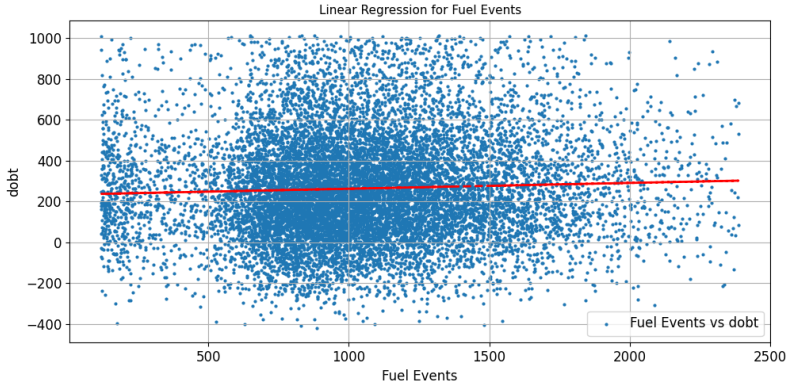


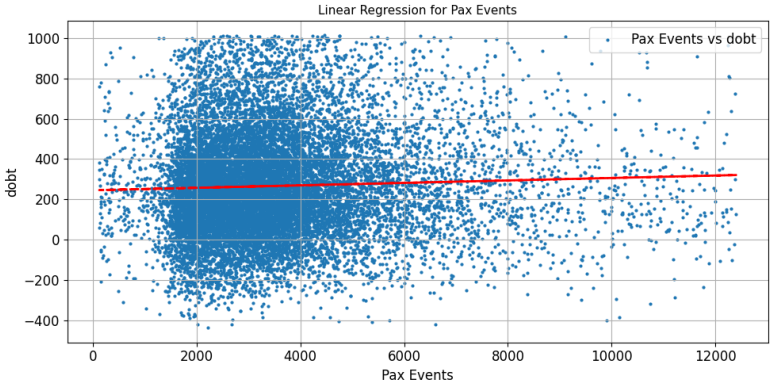


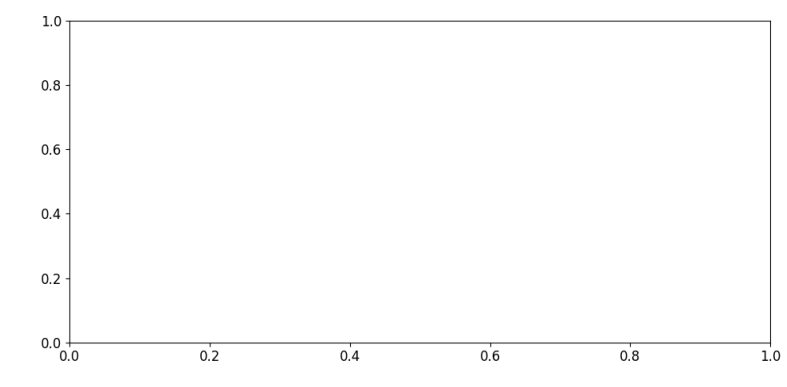




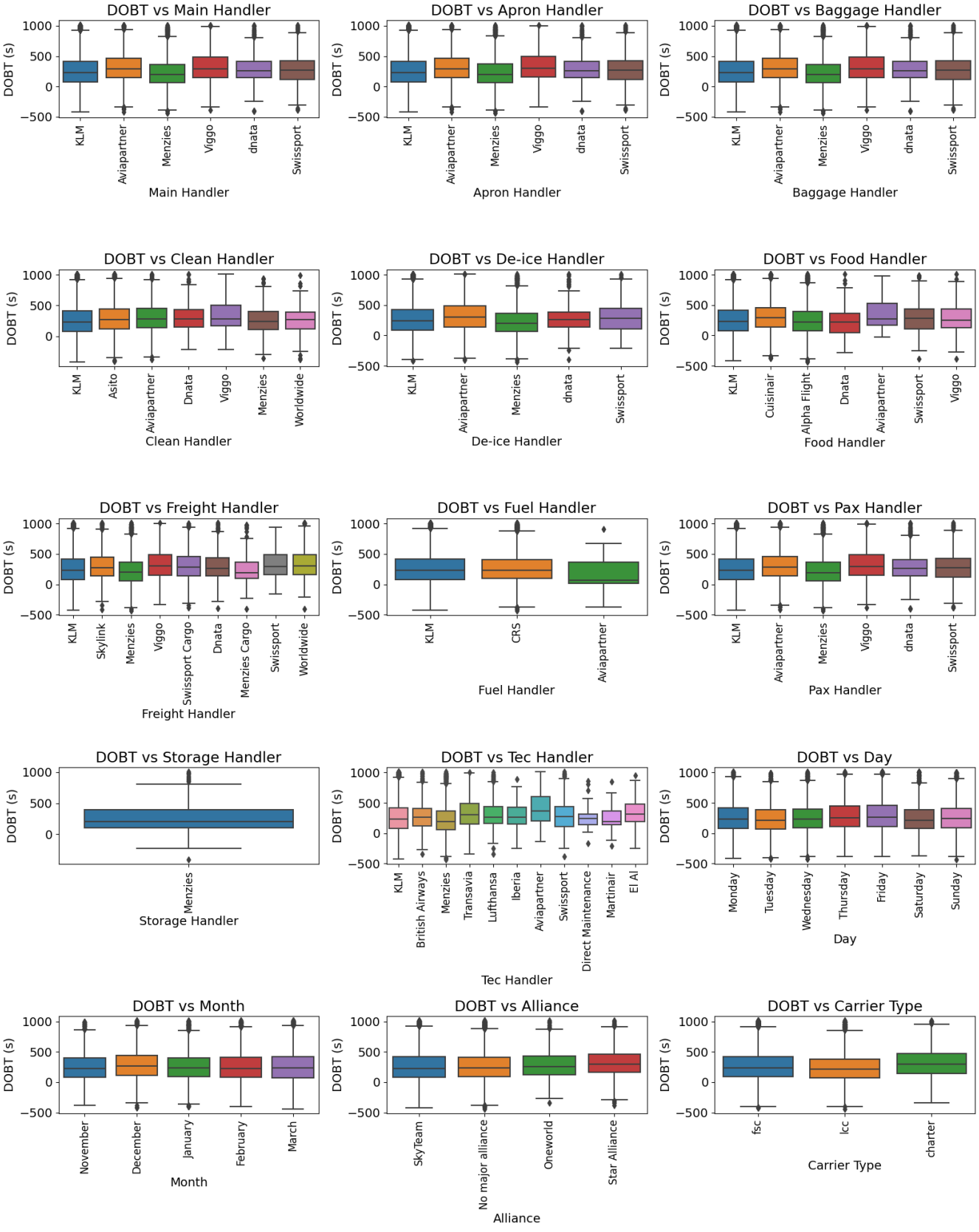


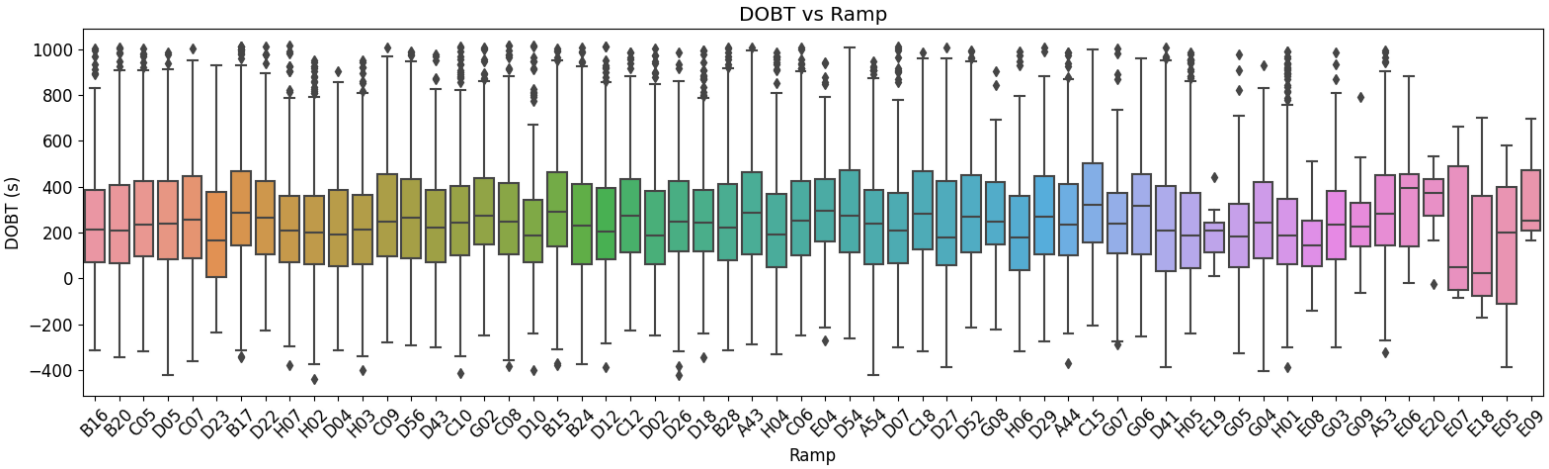


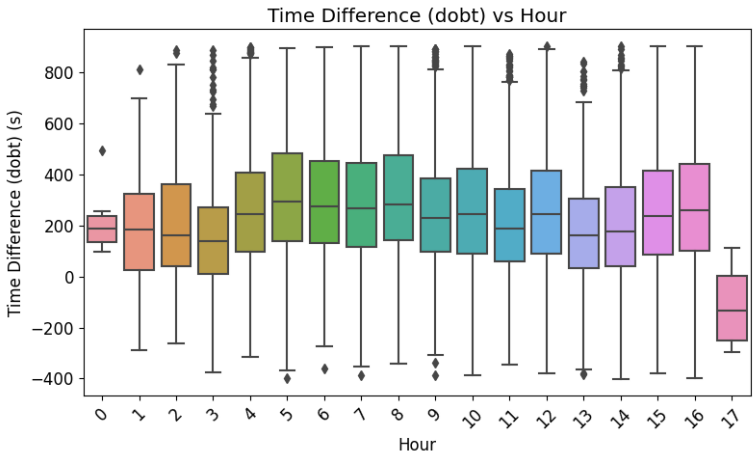


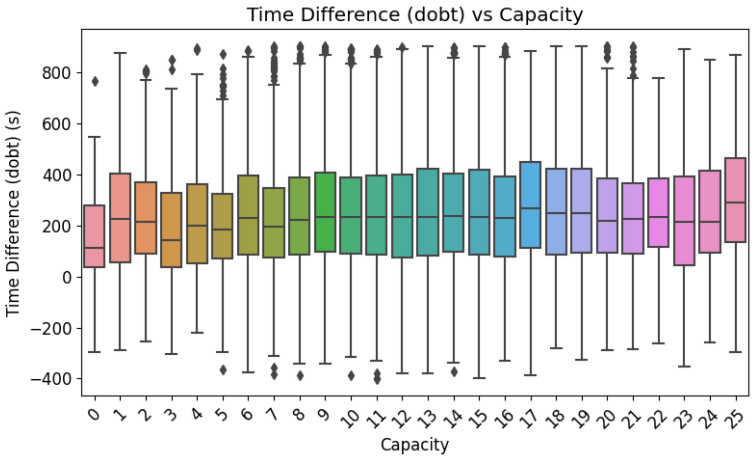


Appendix 8: Boxplots of categorical/discrete features

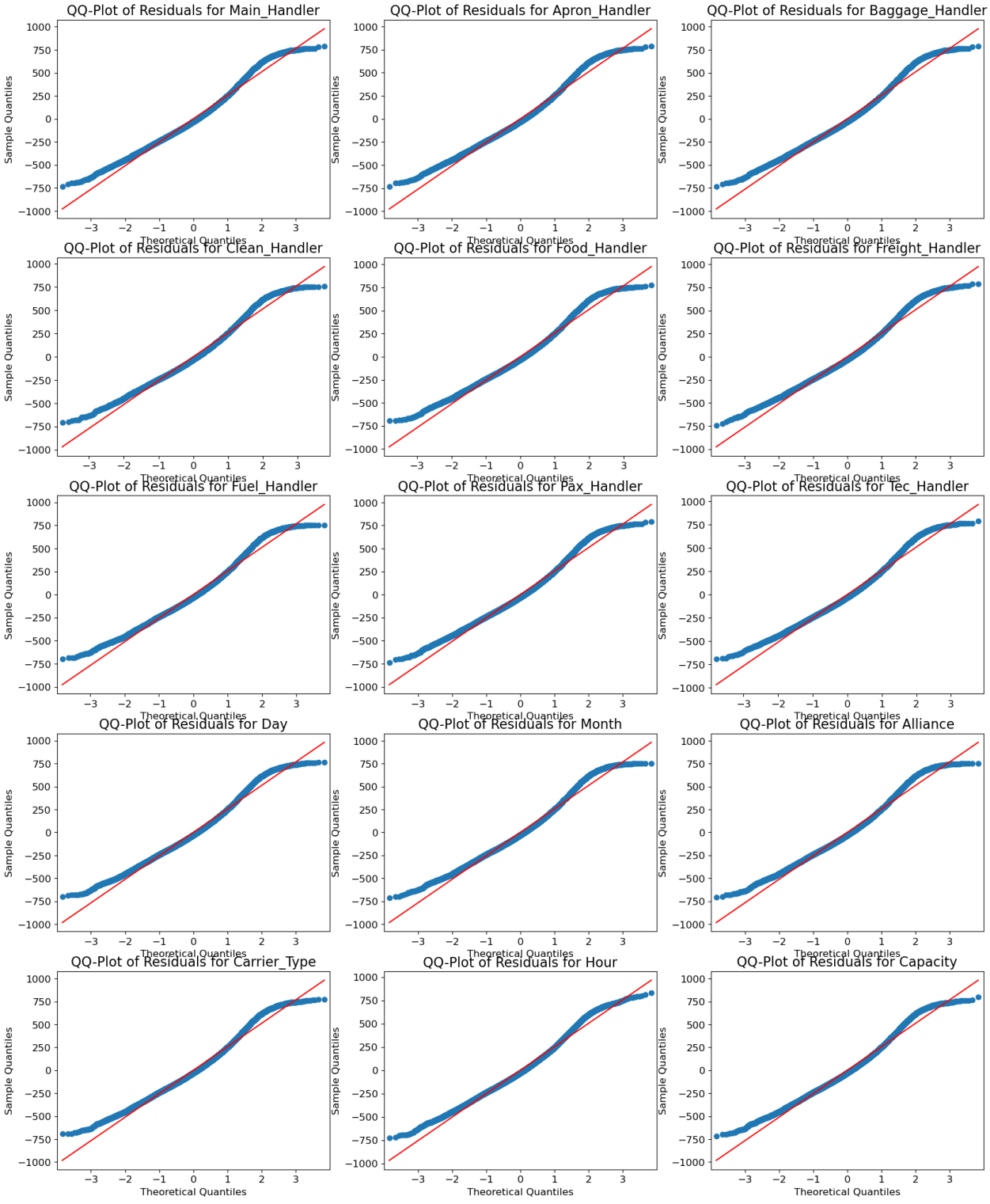


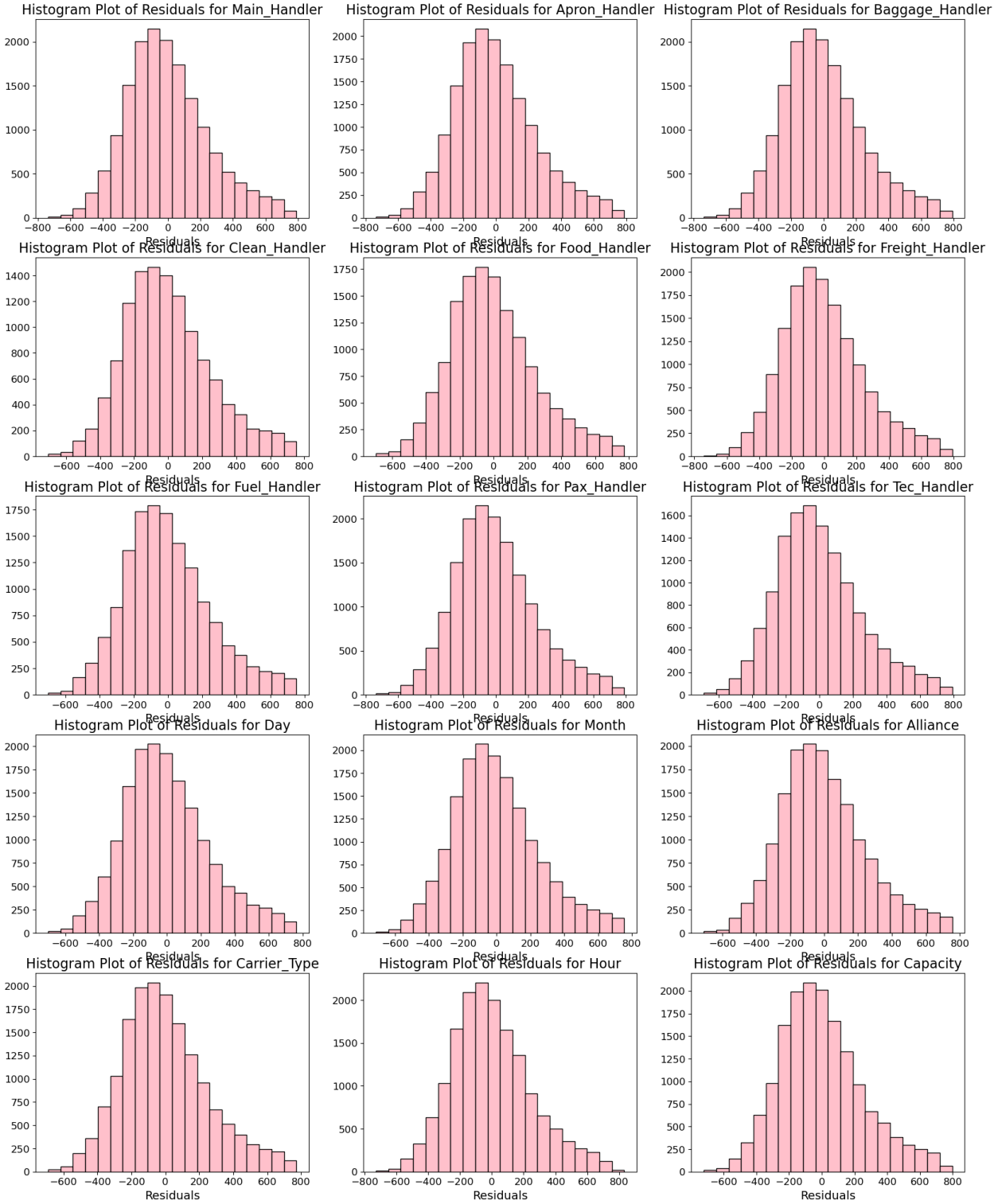

DOBT vs Ramp
DOBT (s)
Ramp


Time Difference (dobt) vs Hour
Time Difference (dobt) (s)
Hour

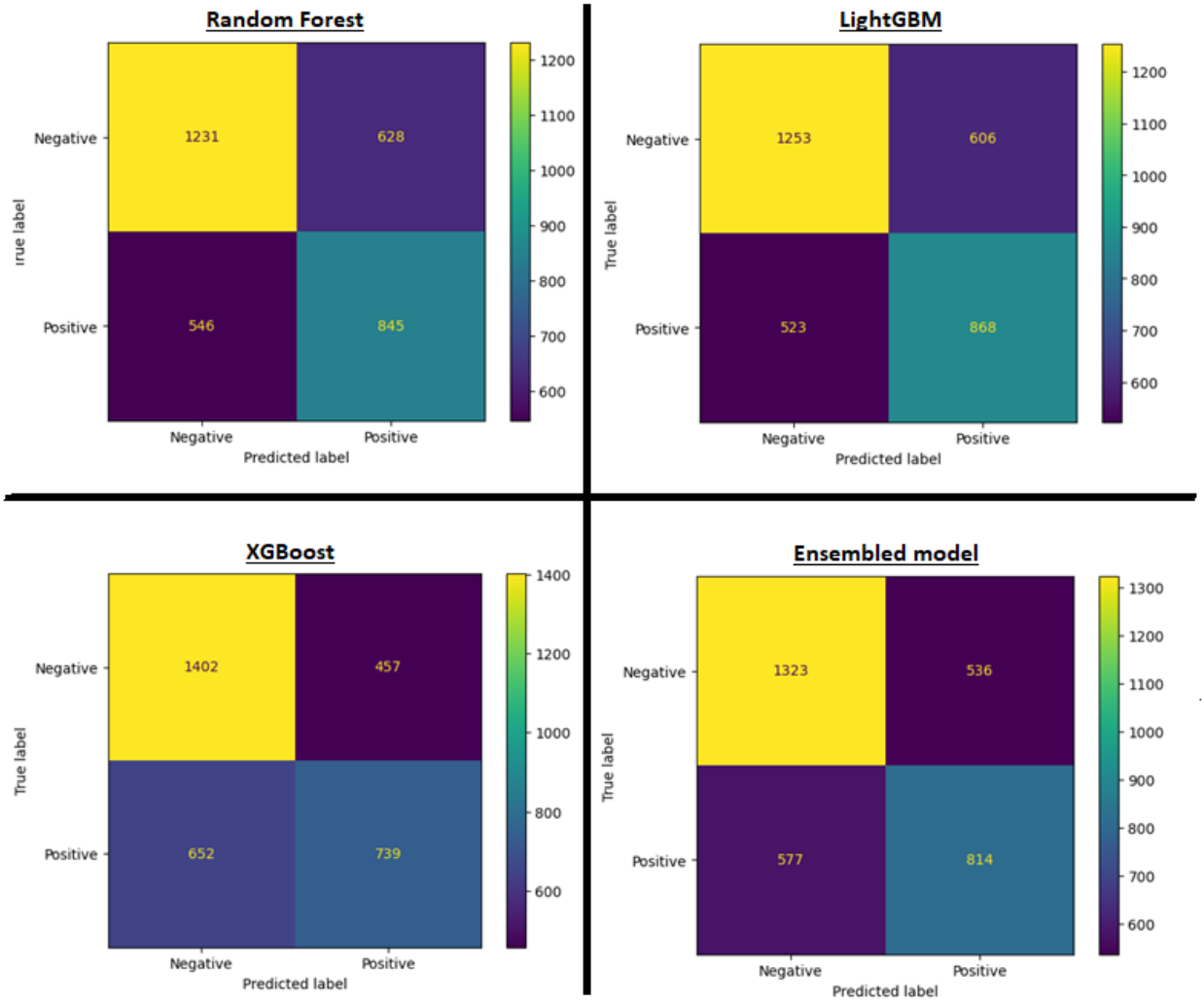

Time Difference (dobt) vs Capacity
Time Difference (dobt) (s)
Capacity

Appendix 9: Residual plots categorical features

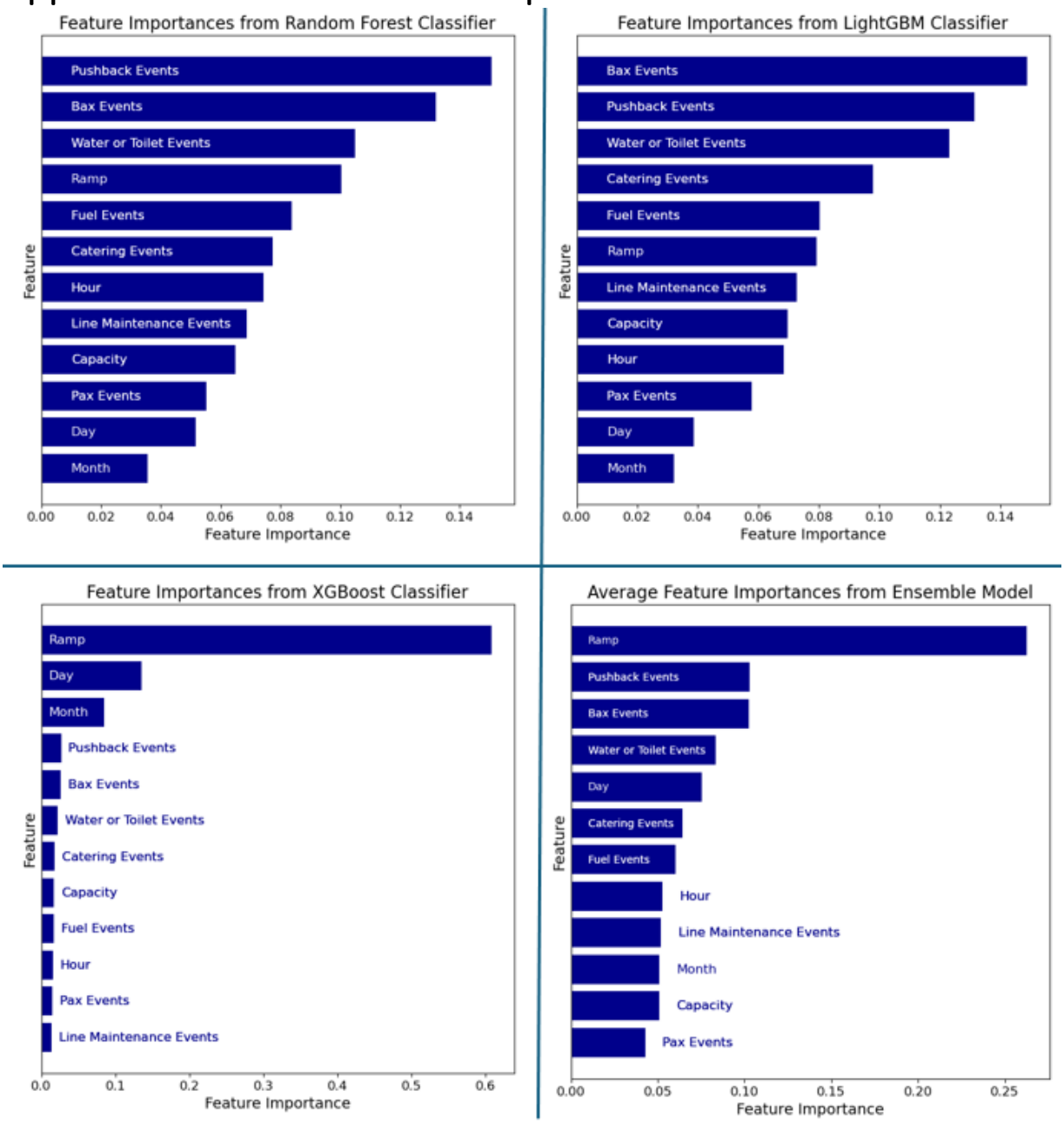



Histogram Plot of Residuals for Main_Handler
Histogram Plot of Residuals for Apron_Handler
Histogram Plot of Residuals for Baggage_Handler
Histogram Plot of Residuals for Clean_Handler
Histogram Plot of Residuals for Food_Handler
Histogram Plot of Residuals for Freight_Handler
Histogram Plot of Residuals for Fuel_Handler
Histogram Plot of Residuals for Pax_Handler
Histogram Plot of Residuals for Tec_Handler
Histogram Plot of Residuals for Day
Histogram Plot of Residuals for Month
Histogram Plot of Residuals for Alliance
Histogram Plot of Residuals for Carrier_Type
Histogram Plot of Residuals for Hour
Histogram Plot of Residuals for Capacity
Residuals

Appendix 10: Confusion matrices



Appendix 11: Model feature importance



Appendix 12: Counterfactual explanations

Bax Events	Catering Events	Line Maintenance Events	Water or Toilet Events	Pushback Events	Fuel Events	Pax Events	Capacity	Ramp	Month	Day	Hour
87.24	0	56.25	8.07	24.08	23.71	36.89	16	G05	December	Wednesday	15

100%| 1/1 [00:01<00:00, 1.48s/it]
Counterfactuals (highlighted changes):

Bax Events	Catering Events	Line Maintenance Events	Water or Toilet Events	Pushback Events	Fuel Events	Pax Events	Capacity	Ramp	Month	Day	Hour
87.24	0	56.25	8.07	24.08	23.71	36.89	2	G05	**January**	Wednesday	**7.00**
87.24	0	56.25	8.07	24.08	**0.62**	**10.30**	3	G05	December	Wednesday	15.0
87.24	0	56.25	8.07	24.08	23.71	36.89	4	G05	December	Wednesday	**7.00**
87.24	0	56.25	8.07	24.08	23.71	**15.20**	4	G05	**February**	Wednesday	15.0
87.24	0	56.25	8.07	24.08	23.71	36.89	0	G05	December	**Monday**	**5.00**
87.24	0	56.25	8.07	24.08	23.71	36.89	3	G05	December	Wednesday	**7.00**
87.24	0	**4.24**	8.07	24.08	23.71	36.89	16	G05	**November**	Wednesday	**4.00**
87.24	0	56.25	8.07	24.08	23.71	36.89	1	G05	**November**	Wednesday	15.0
87.24	0	56.25	8.07	24.08	23.71	36.89	6	G05	**January**	Wednesday	**6.00**
87.24	0	56.25	8.07	24.08	**8.67**	36.89	3	G05	December	Wednesday	**7.00**

Bax Events	Catering Events	Line Maintenance Events	Water or Toilet Events	Pushback Events	Fuel Events	Pax Events	Capacity	Ramp	Month	Day	Hour
26.88	0	88.17	0	7.01	35.39	84.15	10	H02	February	Friday	12

100%| 1/1 [00:01<00:00, 1.44s/it]
Counterfactuals (highlighted changes):

Bax Events	Catering Events	Line Maintenance Events	Water or Toilet Events	Pushback Events	Fuel Events	Pax Events	Capacity	Ramp	Month	Day	Hour
26.88	0	88.17	0	7.01	**22.89**	84.15	10	H02	February	Friday	**14.00**
26.88	0	88.17	0	**3.10**	35.39	84.15	10	H02	February	Friday	12.0
6.64	0	88.17	0	7.01	35.39	84.15	10	H02	February	Friday	12.0
26.88	0	**23.17**	0	7.01	35.39	84.15	10	H02	February	Friday	**16.00**
26.88	0	88.17	0	7.01	35.39	84.15	1	H02	February	Friday	12.0
26.88	0	88.17	0	**2.20**	35.39	84.15	3	H02	February	Friday	12.0
26.88	0	88.17	0	7.01	35.39	84.15	10	H02	February	Friday	**17.00**
26.88	0	88.17	0	**3.70**	35.39	84.15	10	H02	February	Friday	**13.00**
26.88	0	88.17	0	7.01	35.39	84.15	9	H02	February	Friday	12.0
14.15	0	88.17	0	7.01	35.39	84.15	10	H02	February	Friday	12.0

Bax Events	Catering Events	Line Maintenance Events	Water or Toilet Events	Pushback Events	Fuel Events	Pax Events	Capacity	Ramp	Month	Day	Hour
0.14	19.39	84.42	3.95	14.96	0.92	79.26	21	C15	February	Friday	19

100%| 1/1 [00:01<00:00, 1.72s/it]
Counterfactuals (highlighted changes):

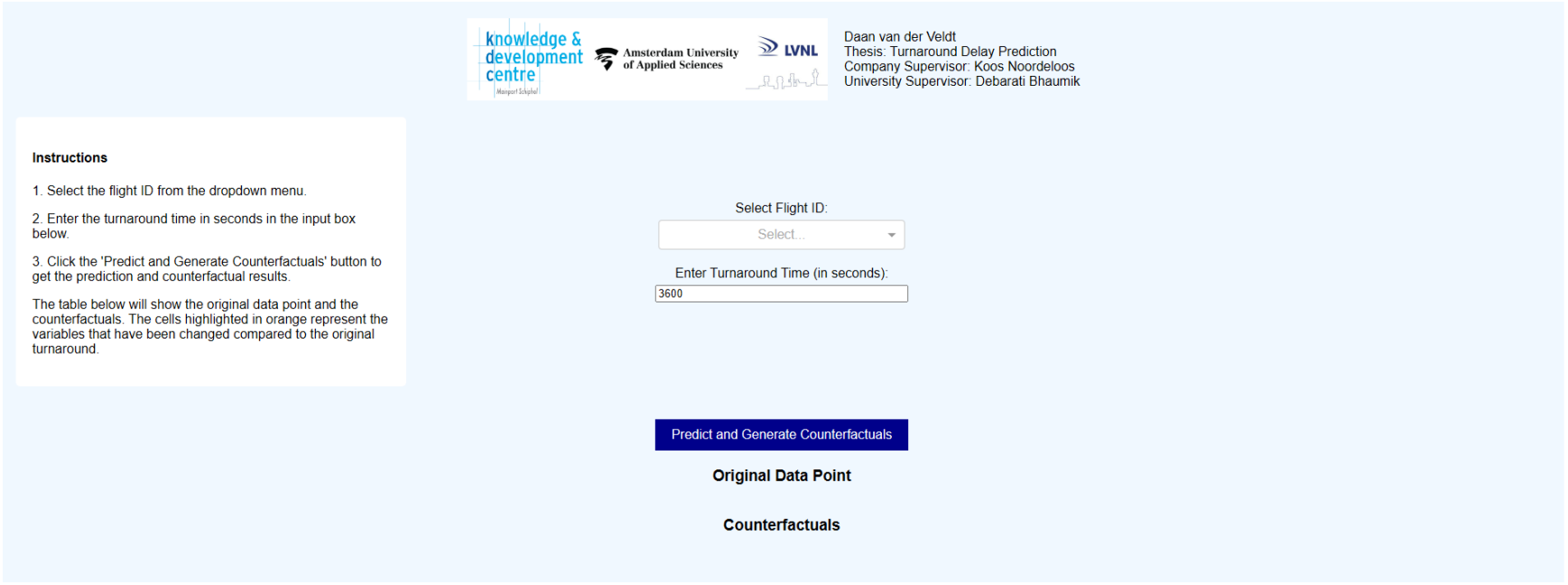
Bax Events	Catering Events	Line Maintenance Events	Water or Toilet Events	Pushback Events	Fuel Events	Pax Events	Capacity	Ramp	Month	Day	Hour
0.14	19.39	**77.21**	3.95	14.96	0.92	79.26	2	C15	February	Friday	19
0.14	19.39	84.42	3.95	14.96	0.92	79.26	0	**G06**	February	Friday	19
0.14	19.39	84.42	3.95	14.96	0.92	79.26	2	C15	February	Friday	19
0.14	19.39	84.42	3.95	14.96	0.92	79.26	2	C15	February	Friday	19
0.14	19.39	84.42	3.95	**4.10**	0.92	79.26	0	C15	February	Friday	19
0.14	19.39	84.42	3.95	**14.10**	0.92	79.26	2	C15	February	Friday	19
0.14	19.39	84.42	3.95	14.96	0.92	79.26	3	C15	February	Friday	19
0.14	19.39	84.42	3.95	14.96	0.92	79.26	3	C15	February	Friday	19
0.14	19.39	**25.69**	3.95	14.96	0.92	79.26	3	C15	February	Friday	19
0.14	19.39	84.42	3.95	14.96	0.92	79.26	0	C15	February	Friday	19

Bax Events	Catering Events	Line Maintenance Events	Water or Toilet Events	Pushback Events	Fuel Events	Pax Events	Capacity	Ramp	Month	Day	Hour
12.6	0	75.36	0	15.76	28.75	85.02	14	C06	December	Friday	19

100%| 1/1 [00:01<00:00, 1.30s/it]
Counterfactuals (highlighted changes):

Bax Events	Catering Events	Line Maintenance Events	Water or Toilet Events	Pushback Events	Fuel Events	Pax Events	Capacity	Ramp	Month	Day	Hour
12.60	0.00	75.36	0	15.76	**7.52**	85.02	14	C06	**March**	Friday	19
12.60	**0.10**	75.36	0	15.76	28.75	**69.90**	14	C06	December	Friday	19
12.60	0.00	75.36	0	15.76	28.75	**67.50**	11	C06	December	Friday	19
12.60	0.00	**56.28**	0	15.76	28.75	85.02	14	C06	December	Friday	19
12.60	0.00	**42.42**	0	15.76	**7.01**	85.02	14	C06	December	Friday	19
2.46	0.00	**10.27**	0	15.76	28.75	85.02	14	C06	December	Friday	19
12.60	0.00	75.36	0	15.76	28.75	**20.60**	14	C06	December	Friday	19
12.60	0.00	75.36	0	15.76	28.75	**29.70**	14	C06	December	Friday	19
12.60	0.00	**40.98**	0	15.76	28.75	**17.10**	14	C06	December	Friday	19
12.60	0.00	75.36	0	15.76	**4.70**	85.02	14	C06	December	Friday	19

Appendix 13: Dashboard



Daan van der Veldt
Thesis: Turnaround Delay Prediction
Company Supervisor: Koos Noordeloos
University Supervisor: Debarati Bhaumik

Instructions

1. Select the flight ID from the dropdown menu.

2. Enter the turnaround time in seconds in the input box below.

3. Click the 'Predict and Generate Counterfactuals' button to get the prediction and counterfactual results.

The table below will show the original data point and the counterfactuals. The cells highlighted in orange represent the variables that have been changed compared to the original turnaround.

Select Flight ID:

28

Enter Turnaround Time (in seconds):

3600

Predict and Generate Counterfactuals

The predicted difference between the TOBT and AOBT is more than 5 minutes

Original Data Point

Bax Events	Capacity	Catering Events	Day	Fuel Events	Hour	Line Maintenance Events	Month	Pax Events	Pushback Events	Ramp	Water or Toilet Events
149	14	179	Monday	180	20	3345	December	3102	186	C08	8

Counterfactuals

Bax Events	Capacity	Catering Events	Day	Fuel Events	Hour	Line Maintenance Events	Month	Pax Events	Pushback Events	Ramp	Water or Toilet Events
149	11	179	Monday	180	20	1852	December	3102	186	C08	8
149	14	179	Monday	180	20	1527	December	3102	186	C08	8
149	14	179	Monday	180	20	1945	December	3102	58	C08	8
149	14	179	Monday	180	20	367	December	3102	186	C08	8
149	14	179	Monday	180	20	2707	December	3102	186	C08	7
149	14	179	Monday	180	20	1966	December	3102	186	C08	8
149	14	179	Monday	180	20	3345	December	3102	29	C08	8
149	14	179	Monday	31	20	2018	December	3102	186	C08	8
149	9	4	Monday	180	20	3345	December	3102	186	C08	8
149	8	179	Monday	180	20	3345	December	3102	186	C08	11