Module Code: CT046-3-M-AML

Module Name: Applied Machine Learning

Individual Assignment

Predicting Train Service Disruptions Using Supervised Machine Learning on UK Rail Data

Student Name: Muhammad Yousouf Ali Budullah
TP Number: TP086704
Intake Code: APDMF2501DSBA(BI)(PR)
Programme: MSc Data Science & Business Analytics
Module Lecturer: Assoc. Prof. Dr. Nirase Fathima Abubacker

Date of Submission: June 15, 2025

Abstract

Train delays and cancellations continue to challenge the reliability and efficiency of national railway systems, particularly in the UK. Previous studies have indicated that machine learning (ML) holds promise for predicting such disruptions, with ensemble models like Random Forest and XGBoost showing the most success. Unlike the existing literature that uses sensors or operational data, this study uses ML on a structured ticket-level mock dataset acquired from Maven Analytics. Incorporated into the dataset are features such as ticket type, booking lead time, and historical route disruption likelihood. In the data preparation we exclude all the post-event features. In addition SMOTE algorithm with class weighting is applied to address a strong class imbalance (13% disrupted journeys). Logistic Regression, Random Forest, and XGBoost were implemented and tuned within cross-validated pipelines. According to the evaluation based on macro-averaged F1-score and Matthews Correlation Coefficient, Random Forest offered the best overall performance, with XGBoost offering the highest recall for disruptions. The results of this study confirm the usefulness of ML for the classification of disruption using structured pre-journey data.

Keywords: machine learning, train delays, classification, UK rail, imbalance handling, XGBoost, Random Forest, predictive modelling

Contents

Abstract **1**

Part A: Literature Review / Related Works **3**

Title . . . 3
Introduction . . . 3
Aim . . . 4
Objectives . . . 4
Scope . . . 4
Related Works . . . 4
Theme 1: ML Classification for Transport Disruption . . . 4
Theme 2: Handling Class Imbalance in Classification . . . 6
Theme 3: Feature Engineering in Railway Delay Prediction . . . 6
Theme 4: Evaluation Metrics in Imbalanced Classification . . . 7
Summary of Related Works . . . 8

Part B: Methods and Experimental Design **11**

Methods . . . 11
Dataset Preparation . . . 12
Data Cleaning and Formatting . . . 12
Feature Engineering . . . 13
Exploratory Data Analysis (EDA) . . . 14
Model Implementation . . . 17
Model Selection Justification . . . 17
Feature Filtering . . . 18
Handling Class Imbalance . . . 18
Model Pipelines and Tuning . . . 18
Model Validation . . . 20
Analysis & Recommendations . . . 22
Conclusion . . . 23

Bibliography **26**

Part A

Title

Predicting Train Service Disruptions Using Supervised Machine Learning on UK Rail Ticket Data

Introduction

Train services are a critical part of the UK's public infrastructure, serving commuters, logistical suppliers, and travellers equally. However, frequent disruptions, like delays and cancellations, have considerable effects on the reliability of service, the satisfaction of passengers, and operational efficiency. These disruptions may lead to missed appointments, congested platforms, logistical bottlenecks, and financial losses for both passengers and operators.

This study uses supervised machine learning (ML) approaches to forecast train service disruptions. Disruptions are classified as either delays or cancellations. The intent is to identify each rail journey as either "disrupted" or "on time" based on information available prior to the trip, such as ticket class, route and scheduled departure time, and previous delay trends. Understanding the historical trends enables ML models to anticipate disruptions and make proactive decisions such as providing warnings, reallocating resources, or altering timetables.

While train delay prediction has been thoroughly explored using real-world data from national rail networks, there is no indication of machine learning being used on the mock dataset selected for this study. Although it is a fictional dataset, it is intended to imitate real-world rail operations and customer behaviour. Its usability and structured data make it suitable for modelling and testing classification algorithms. Furthermore, the study has real business significance by influencing refund management, passenger communication, and operational planning.

In order to address the problem, the machine learning pipeline was utilised. The dataset was pre-processed by cleaning, feature engineering, and transforming key variables including journey time and delay categories. The binary classification objective was constructed by categorizing delays and cancellations under the label "disrupted." Given the dataset's class imbalance, SMOTE (Synthetic Minority Over-sampling Technique) and class weighting were applied to ensure as fair learning as possible. To determine the most effective strategy, three models; Logistic Regression, Random Forest, and XGBoost were trained and evaluated using measures such as accuracy, F1-score, precision, and recall.

The data was obtained via Maven Analytics' Data Playground. It is a mock rail dataset that simulates UK National Rail rides from January to April 2024, comprising of 31,653 records and 18 fields. The dataset includes information about ticket types, departure and arrival stations, travel times, ticket prices, delays, and refund requests. Its structure also includes both cate-

gory and numerical data, making it appropriate for classification jobs while presenting genuine difficulties such as class imbalance.

Aim

To develop and evaluate supervised machine learning models for predicting train service disruptions using mock UK National Rail data.

Objectives

- To understand and prepare the dataset through cleaning and pre-processing
- To engineer features and define a binary disruption classification target
- To address class imbalance using SMOTE and class weighting techniques
- To implement and evaluate three supervised classification models
- To analyse model performance using appropriate metrics
- To interpret results and derive insights

Scope

This study limits itself to predicting whether or not a journey will be disrupted using historical and pre-journey data. It makes no attempt to predict the exact disruption duration or the specific cause of it. The study centres around supervised classification, with a single dataset and three distinct models. Further advances could include external factors time-series modelling, or unsupervised anomaly identification.

Related Works

Theme 1: ML Classification for Transport Disruption

Recently, machine learning (ML) algorithms have been successfully used in railway analytics for the identification and prediction of train disruptions. These models assist and improve decision-making in difficult, uncertain operating environments. This section investigates literature that utilised classification-based ML methods to predict delays. Random Forest (RF), XGBoost and Logistic Regression were examined as they are the models that will be implemented on the mock dataset.

In a study by Li et al. (2021) near-term train delays were predicted using a Random Forest model based on operational data from the Dutch railway system. The authors highlighted the model's ability to capture small and long delays by splitting their dataset to two delay severity categories ($\leq$3 min and ¿3 min). The Random Forest technique achieved better accuracy when

compared to XGBoost, artificial neural networks (ANN), and the Gradient-boosted decision tree (GBDT) algorithms in particular, for three-minute delays.

Chen et al. (2024) conducted a detailed methodology review on incident delay prediction in the urban railway system. According to their studies evaluation of regression and classification algorithms, XGBoost was especially successful in real-time prediction. The study showed that when faced with operational noise and nonlinearity the research revealed that, classification models, especially tree-based ones, consistently performed better than linear baselines such as Logistic Regression.

Liu et al. (2023) took an alternative approach, utilizing natural language processing with with XGBoost by natural language processing. Word2Vec vector representations of delay related texts were given to an XGBoost classifier. The hybrid model did better than alternative techniques under consideration, indicating that the combination of both structured and unstructured data can improve performance.

Sarhani and Voß (2024) conducted another industry-wide study and discovered that Random Forest and Gradient Boosting are ensemble methods that are commonly used in modern transport systems. Their study of rail delay prediction research found that, when real-time feeds are available, ensemble classifiers are more effective than traditional methods in detecting abnormalities and delays. Even though logistic regression is still a useful baseline model it has been shown to struggle capturing complicated feature interactions without extensive preprocessing (Sarhani & Voß, 2024).

Laifa et al. (2022) contribute to the discussion by presenting a two-level hybrid model that classification-regression model that employs LightGBM. Their classification process divided delay intervals into specific groups (0-5min, 6-10min etc) corresponds to our aim to classify the disrupted vs non-disrupted. Their approach did better than single-stage models, suggesting the potential of disruption-focused classification algorithms (Laifa et al., 2022).

While none of this literature used the same mock UK railway dataset as the current study, each of them worked in similar domains and address similar predictive targets. Most rely on operational logs or sensor data, whilst this study uses structured data from Maven Analytics, which includes features such as ticket type, refund eligibility, travel class, and scheduled arrival time vs. actual arrival. This distinction adds a new layer of business and passenger context to the disruption classification task. Nonetheless, the constant performance of RF, XGBoost, and Logistic Regression in previous train classification research establishes an appropriate methodological basis for their use here.

Theme 2: Handling Class Imbalance in Classification

The presence of class imbalance is a significant challenge to binary classifiers. As only a small proportion of journeys are disrupted in the mock dataset used for this study this classification problem is highly skewed. If not addressed, this imbalance can create models that greatly favour the majority class (on time journeys). As a result, the models achieve a high accuracy that is misleading and does not translate well to the real-world application.

Zhang et al. (2019) proposed WOTBoost, a new ensemble algorithm of weighted oversampling and boosting to address this problem. They can detect hard-to-learn minority class samples in each boosting iteration and applies targeted synthetic oversampling using an adaptive weight distribution . WOTBoost outperformed other methods SMOTE + Decision Tree, ADASYN, and SMOTEBoost in AUC metrics on 7 of 18 public datasets and the best G-mean metrics was achieved on 6 datasets. The method can enhance recall and sensitivity for the minority class while maintaining specificity for the majority class. The current project aims to preserve the data integrity of the two classes. Additionally, it seeks to enhance the identification of the minority class using a dual approach of SMOTE with model-based weighting (Zhang et al., 2019).

Abdelhamid and Desai (2024) also examined various methods, such as SMOTE, class weighting, and decision threshold calibration, to address class imbalance through a comparative analysis of 30 real-world imbalanced datasets. As per the findings of the study, overall F1-score for threshold calibration gives the best results at 0.617, while SMOTE and class weighting follow at 0.605 and 0.594, respectively, but with different results depending on the context used. While SMOTE improved the recall of the minority class, it did so at the cost of probability calibration, but class weighting was more consistent across datasets. According to the findings, combining both approaches utilising SMOTE for producing synthetic minority samples and class weighting to influence classifiers' sensitivity could be beneficial in high cost or low frequency scenarios, such as for the classification of transport disruption (Abdelhamid & Desai, 2024).

As such, the current study employs a combined SMOTE and class weighting approach on all the 3 models where LR, RF and XGBOOST. Giving prominent attention to the significant class imbalance observed in the dataset, the literature review informed this dual-method approach. While the two studies analyzed were inconsistent with railway data, it is reasonable to generalise the works to any structured, skewed dataset. So, these justifications support the use of a hybrid imbalance handling pipeline in transport disruption prediction.

Theme 3: Feature Engineering in Railway Delay Prediction

Feature engineering is necessary for enhancing the performance of ML model in the railway disruption classification task. It turns raw data into useful variables that reveal patterns and

relationships for prediction success. In transport contexts delay flags, and route identifiers transformations are often more informative than the original domain variables. This section showcases previous works that used these techniques and informed this study.

As mentioned previously, Laifa et al. (2022) used LightGBM to create a two-level machine learning model for classification then prediction of train delays on the Tunisian rail network. Extensive feature engineering via cyclic encoding and target encoding and one-hot encoding as per type of variable were crucial. To retain smoothness in delay outcome across both time-based (e.g. time of day, day of week) and categorical features (e.g. train IDs, motif), were encoded accordingly. For instance, weekdays and seasons were encoded cyclically (via sine and cosine pairs) to retain their temporal continuity. On the other hand, categorical fields such as train IDs and delay reasons ("motifs") were target-encoded based on mean delay outcome. The delay intervals also created additional variables in the two-level model; at the model's first level, these were learned, and at the second level, they were sent as features to the regression predictor. This model's performance improved due to multiple stages of feature creation.

Tiong (2024) elaborated on four feature engineering methods which are feature creation, discretization, categorical encoding and normalization. Tiong (2024) applied this to a data set that used railway operation data. He highlighted the importance of delay variables at the current station for feature creation. For discretization, he mentioned domain-based binning of environmental variables. Temperature was one variable for which empirical thresholds that affect punctuality were presented. For instance, cold, normal, extremely cold. These are sent for one-hot encoding after binning to create a binary form of these. It was further emphasized that normalization is needed for gradient-based optimization models.

The feature engineering applied in the current study were largely guided by the literature. We used the timestamp and location data to come up with the derived variables like estimated journey time, a delay difference, and a route id. These changes are in line with the practices mentioned in the literature, aimed at exposing temporal, operational, and ticket-related patterns that impact disruption likelihood.

Theme 4: Evaluation Metrics in Imbalanced Classification

In tasks involving imbalanced binary classifications, such as predicting train disruptions, you can not just rely on model accuracy alone for evaluation . A model that simply predicts the majority class (not disrupted) can achieve high accuracy, but this means it does not identify any disrupted journeys. This can lead to wrong assessments and poor decisions.

To overcome this, existing literature assessed the classifiers using precision, recall, and F1-score, computed per class along with macro and weighted averages. These measures give a better knowledge of how well the model performs for both classes. F1-score, in particular, gets the best of both worlds between precision and recall, giving a more accurate picture of

performance in skewed datasets (Gao et al., 2025).

The Matthews correlation coefficient (MCC) is also commonly viewed as a strong and balanced metric for imbalanced binary classification. The MCC takes into account all factors of the confusion matrix and remains fairly stable in case of significant skewing of class distributions. Studies have also shown MCC to outperform or remain consistent compared to F1 and accuracy across various imbalance levels (Boughorbel et al., 2017).

This result further justifies the choice to prioritize both the F1-score and class-wise evaluations in this project. MCC is still a strong candidate for future iterations or cases where a single balanced performance score is required. In the end, these evaluation methods ensure that performance assessment is done fairly for disrupted and non-disrupted outcomes.

Summary of Related Works

Table 1: Table of References

Study	Li et al. (2021)
Dataset Used	Multiple data sources
Focus/Method	Delay Prediction
Models Used	RF, XGBoost, ANN
Key Findings	RF outperformed other algorithms
Relevance to This Work	Validates using tree-based models for transport disruption
Study	Chen et al. (2024)
Dataset Used	Urban Railway, Hong Kong
Focus/Method	Methodology review
Models Used	Multiple incl. XGBoost
Key Findings	XGBoost tree models outperform other models
Relevance to This Work	Supports model choice and classification framing
Study	Liu et al. (2023)
Dataset Used	Wuhan–Guangzhou High-Speed Railway
Focus/Method	Delay prediction with NLP
Models Used	XGBoost + Word2Vec
Key Findings	Achieved highest accuracy with combined features
Relevance to This Work	Affirms hybrid feature importance
Study	Sarhani and Voß (2024)
Dataset Used	Zurich Railways

Focus/Method	Delay Prediction
Models Used	RF, LR, GR, XGB
Key Findings	Ensemble models consistently outperformed LR
Relevance to This Work	Justifies model selection direction
Study	Laifa et al. (2022)
Dataset Used	Tunisian Railways
Focus/Method	2-level LightGBM delay model
Models Used	LightGBM
Key Findings	Feature engineering boosted performance
Relevance to This Work	Inspired use of cyclic & categorical encoding
Study	Tiong (2024)
Dataset Used	Multiple datasets
Focus/Method	Delay prediction
Models Used	RF, LightGBM
Key Findings	Derived delay and route features improved accuracy
Relevance to This Work	Supports feature design choices
Study	Zhang et al. (2019)
Dataset Used	18 imbalanced UCI datasets
Focus/Method	Oversampling + Boosting
Models Used	WOTBoost, SMOTE, ADASYN
Key Findings	WOTBoost improved minority recall and G-mean
Relevance to This Work	Informed combined use of SMOTE + weights
Study	Abdelhamid and Desai (2024)
Dataset Used	30 datasets from various domains
Focus/Method	Comparison of imbalance methods
Models Used	SMOTE, class weights, calibration
Key Findings	No one-size-fits-all; hybrid methods perform best
Relevance to This Work	Supports our use of SMOTE + weight balancing
Study	Gao et al. (2025)
Dataset Used	Multiple datasets
Focus/Method	Metric evaluation for imbalance
Models Used	F1, AUC, MCC
Key Findings	Accuracy unreliable; F1 & MCC preferred

Relevance to This Work	Validates chosen metrics in our imbalanced task
Study	Boughorbel et al. (2017)
Dataset Used	64 datasets
Focus/Method	MCC classifier vs others
Models Used	MCC-optimal, SVM
Key Findings	MCC robust across imbalance
Relevance to This Work	Supports MCC as a balanced metric (discussed, not used)

The table above displays the literature which have strong empirical and methodological foundations for the current study. Research by various authors (Chen et al., 2024; Li et al., 2021; Sarhani & Voß, 2024) validated the use of classification models for rail network disruption prediction, namely Random Forest and XGBoost. While an exact use of the Maven UK ticketing dataset is not found in any of the studies, they all share the same domain of delay analysis of transport. Allowing for an indirect benchmarking is possible.

From the data processing perspective, Laifa et al. (2022) and Tiong (2024) guided the project regarding feature engineering, particularly cyclic encoding for time variables, delay duration, and route ID as derived feature engineering. Furthermore, Zhang et al. (2019) and Abdelhamid & Desai (2024) provided insights on how to handle class imbalance by combining SMOTE and class weighting. Finally, Gao et al. (2025) and Boughorbel et al. (2017) reinforced the use of F1-score and macro-averaged metrics over plain accuracy in imbalanced classification contexts.

With respect to previous studies, the current study provides a different application of supervised machine learning to a UK ticketing data set, encompassing not just operational delays but also customer-facing variables, including refund requests. While other datasets used leverage sensor or dispatch level data, the Maven UK Rail dataset is most suited for schedule based disruption modelling. While the current project draws upon the literature, it is differentiated by methodological, domain and other choices.

Part B

Methods

This study aims to build and evaluate machine learning models to classify whether a UK rail journey will be disrupted (delayed or cancelled). The methodology includes data preparation, model selection, and performance evaluation, all of which are briefly described below.

Data Description.

The dataset used for this project was sourced from the *Maven Analytics Open Data Playground*. It contains synthetic yet realistic records of rail journeys across the UK. The full dataset consisted of **31,653 samples**, each representing a single ticketed journey. As well as **18 features** including.

- **Date and Time attributes:** e.g., date of purchase, departure/arrival times, purchase hour, and booking lead time.
- **Journey details:** such as departure and arrival stations, ticket type, ticket class, route, and delay information.
- **Post-trip outcomes:** journey status (on time, delayed, cancelled), refund requests, and reason for delay.

The target variable is the binary label `Journey Status Binary`, where 1 conveys a disrupted journey (delayed or cancelled) and 0 conveys on-time.

Preprocessing and Feature Engineering.

To ensure model reliability and prevent data leakage, all post-journey features such as actual arrival time, delay reason, and refund request were removed. Only features that would be available prior to a journey's departure were retained. New features were engineered such as.

- **Cyclic Encodings:** for departure hour and day-of-week.
- **Route Encoding:** based on historical disruption rates.
- **Booking Lead Time:** days between purchase and journey.

Categorical variables were one-hot encoded, and continuous features were scaled where required.

Learning Techniques and Tools.

Three supervised learning classifiers were implemented, each chosen based on their success in related literature in delay prediction tasks and suitability for tabular, structured datasets:

- **Logistic Regression:** used as a linear baseline with L1/L2 regularisation.
- **Random Forest:** an ensemble of decision trees offering robustness and non-linearity handling.
- **XGBoost:** a gradient-boosted decision tree model known for its performance on structured data.

Each model was implemented using a pipeline that incorporated `SMOTE` (Synthetic Minority Over-sampling Technique) and class weighting to handle the large class imbalance in the dataset. Hyperparameter tuning was conducted using 5-fold cross-validation with `GridSearchCV`.

Figure 1: General ML Pipeline for Disruption Classification (SMOTE + Scaling + Weighted Classifiers)

The implementation used the following libraries: `pandas`, `numpy`, `scikit-learn`, `xgboost`, `seaborn`, and `imbalanced-learn`.

Evaluation Metrics.

Given the class imbalance (only 13% of journeys were disrupted), accuracy itself was not a reliable measure of performance. Instead, the following metrics were used as well:

- **Precision, Recall, F1-score:** reported per class as well as macro and weighted averages.
- **Matthews Correlation Coefficient (MCC):** a balanced metric that considers all parts of the confusion matrix.

These metrics were chosen based on insights from related literature (Boughorbel et al., 2017; Gao et al., 2025), which emphasize their reliability in evaluating imbalanced classification tasks.

Dataset Preparation

The dataset used in this study was taken from Maven Analytics and contains **31,653 records** of UK rail journeys between January and April 2024. Each row includes information on ticket purchase, travel time, railcard usage, and journey outcome (On Time, Delayed, or Cancelled). The dataset shows a significant class imbalance, with around **87%** of journeys being on time.

Data Cleaning and Formatting

The following preprocessing steps were applied to ensure data quality and structure:

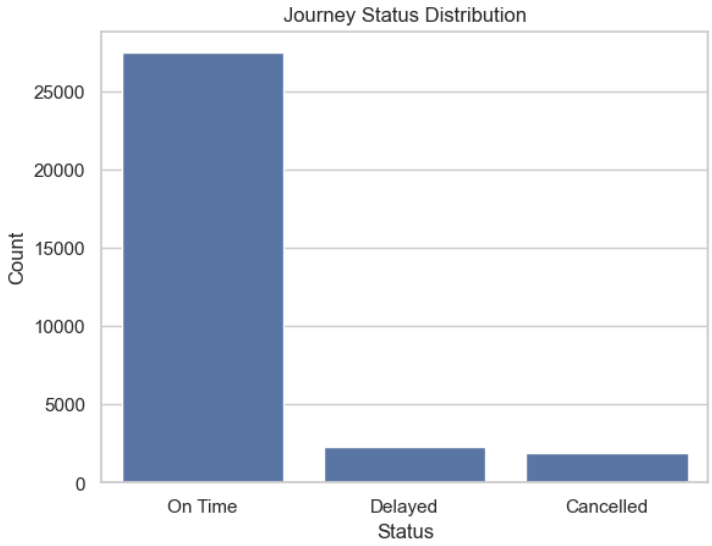

Figure 2: Distribution of Journey Status

- **Removed** `Transaction ID` column (held no predictive value).
- **Dropped** rows where journeys were marked as completed but had missing `Actual Arrival Time`.
- **Imputed** missing values:
  - `Railcard` filled with "None"
  - `Reason for Delay` filled with "Not Applicable"
- **Converted** columns to appropriate data types:
  - `Price` to numeric
  - `Date of Purchase`, `Date of Journey` to datetime
  - Time columns (e.g., `Departure Time`) to `datetime.time`

Feature Engineering

Several features were engineered to enhance the dataset:

- **Route**: Concatenation of departure and arrival stations
- **Route Encoded**: Target encoding using historical disruption rate
- **Temporal features**:
  - `Purchase Hour`, `Departure Hour`, `Day of Week`
  - **Cyclic encodings**: `Day_sin`, `Day_cos`, `DepHour_sin`, `DepHour_cos`
- **Booking Lead Time**: Days between purchase and journey

- **Delay Time (min)**: Difference between predicted and actual arrival times
- **Delay Category**: Grouped into On Time, Slight, Moderate, Severe, and Cancelled
- **Target Variable**: `Journey Status Binary` (0 = On Time, 1 = Disrupted)

Exploratory Data Analysis (EDA)

The following visual analyses were performed to explore variable distributions and identify meaningful patterns in the dataset:

Delay Category Distribution. The 'Delay Category' feature was created to group delays into On Time, Slight, Moderate, Heavy, Severe, and Cancelled. As illustrated in Figure 3, the majority of journeys were on time, followed by a much smaller number of cancelled or heavily delayed trips. This highlights the class imbalance present in the dataset and the need for imbalance handling in later modelling stages.

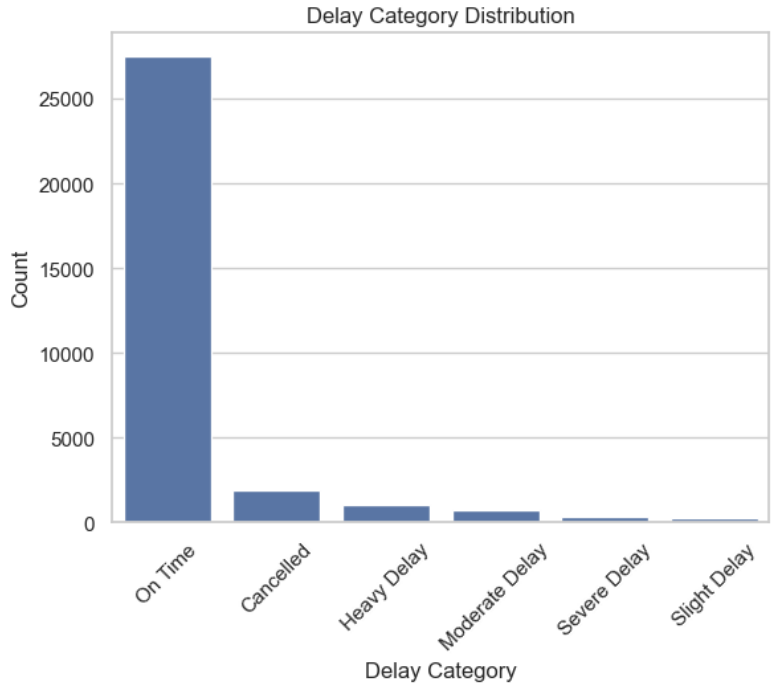

Figure 3: Delay Category Distribution

Delay Duration. Figure 4 shows a right-skewed distribution of delay times. Most journeys experience minimal delays, but a number of journeys have extreme delay values exceeding 100 minutes, which justifies treating this as a classification task rather than regression.

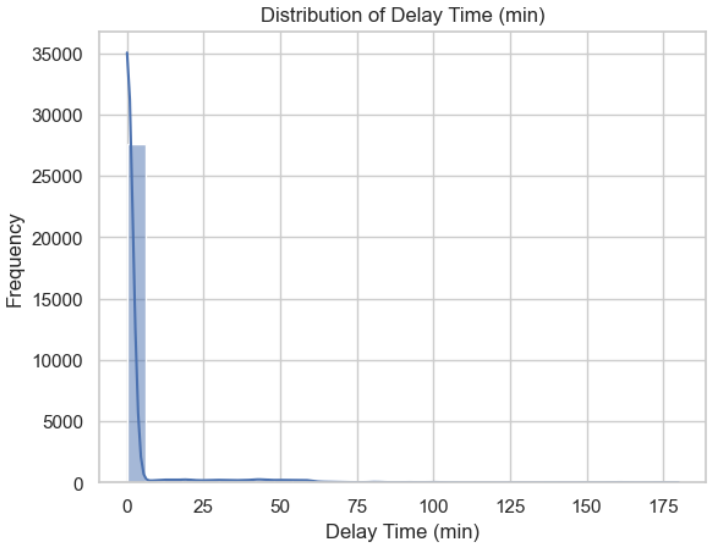

Figure 4: Distribution of Delay Time (min)

Time-of-Day Trends. Delay severity appears to fluctuate throughout the day. Figure 5 illustrates that delays tend to peak during early morning and late afternoon departures, supporting the inclusion of temporal features.

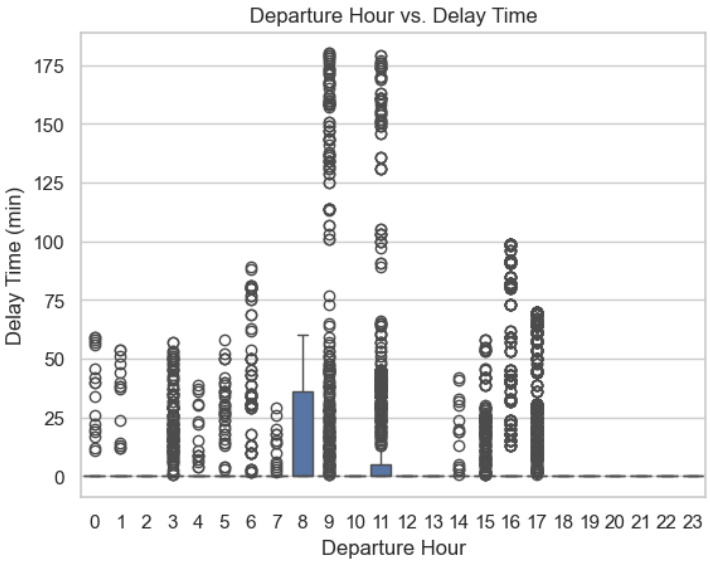

Figure 5: Departure Hour vs. Delay Time

Route-Level Disruption. Each route was assigned a disruption score based on historical data. Figure 6 presents the 10 routes with the highest disruption frequencies, validating the decision to encode route-level disruption likelihood as a feature.

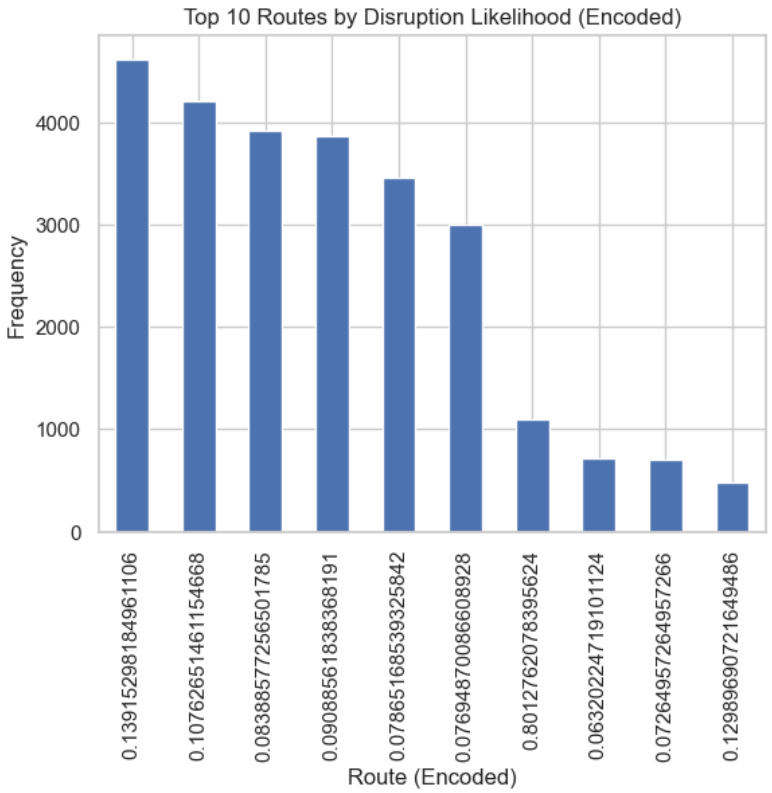

Figure 6: Top 10 Routes by Disruption Likelihood (Encoded)

Feature Correlations. A correlation matrix was generated to identify linear relationships amongst the preprocessed features. As shown in Figure 7, there were no excessively strong correlations that warranted immediate removal. Engineered features such as `Route_encoded`, `Booking Lead Time`, and cyclic time encodings (`Day_sin`, `DepHour_cos`, etc.) did not show concerning redundancy, supporting their inclusion in the final model.

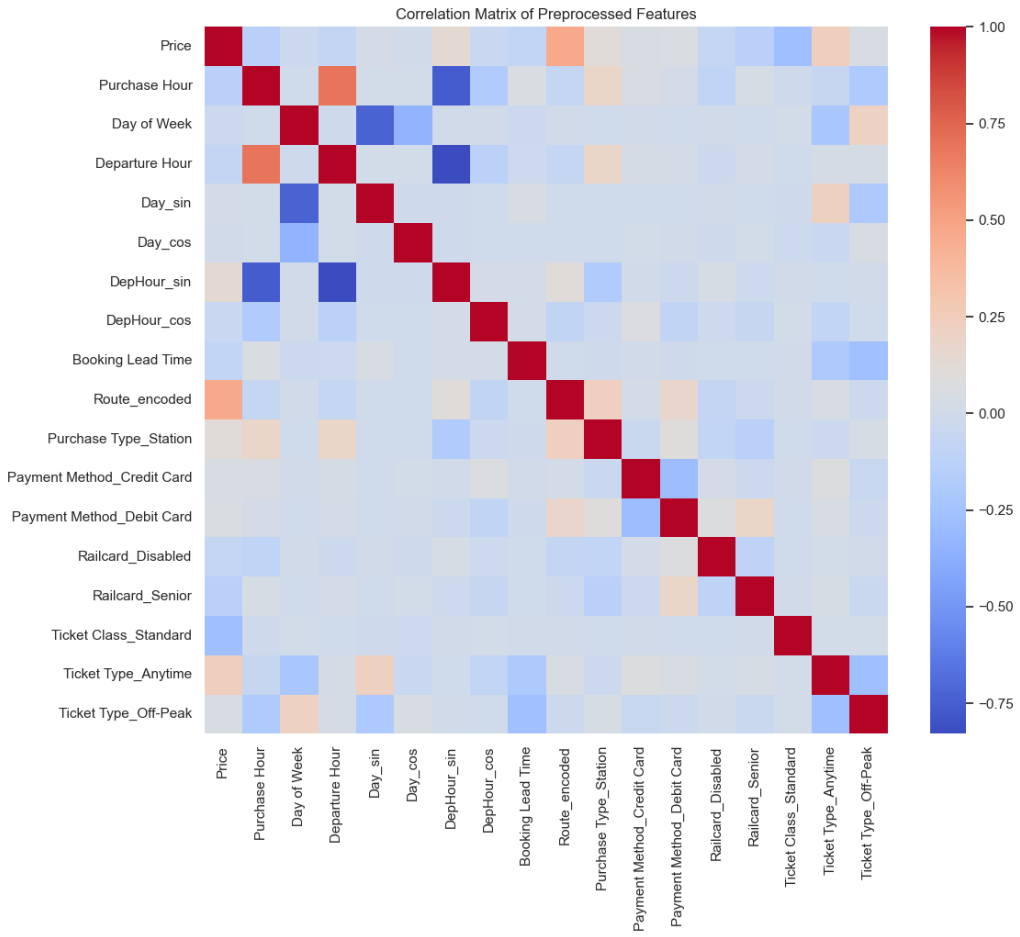

Figure 7: Correlation Matrix of Preprocessed Features Including Engineered Variables

Model Implementation

To address the binary classification problem of predicting UK rail journey disruptions, three supervised learning models were built; **Logistic Regression**, **Random Forest**, and **XGBoost**. These models were chosen based on both their success in the transport disruption literature and their suitability for structured tabular data like the Maven Analytics mock dataset.

Model Selection Justification

- **Logistic Regression (LR)** was selected as a baseline model. It is interpretable and efficient, offering a benchmark for how linear decision boundaries perform in this context. However, the literature highlights its limitations in capturing complex feature interactions (Sarhani & Voß, 2024).
- **Random Forest (RF)** is a tree-based ensemble model that constructs multiple decision trees and aggregates their results. It has displayed solid performance in delay prediction, particularly in distinguishing between minor and major delays (Li et al., 2021).

- **XGBoost (Extreme Gradient Boosting)** is another ensemble method that uses boosting instead of bagging. It builds the trees sequentially, correcting errors from previous iterations. XGBoost has been widely recognized for its success in transportation systems, especially in noisy and nonlinear conditions (Chen et al., 2024; Liu et al., 2023).

These three models provide a robust comparison spectrum; from interpretable linear classifiers (LR) to flexible ensemble models (RF and XGB).

Feature Filtering

To prevent information leakage, only features available *before* the journey were used. Post-event fields like `Actual Arrival Time`, `Refund Request`, and `Delay Category` were excluded. High-cardinality text features like `Route` were replaced with target-encoded features such as `Route_encoded`, reflecting historical disruption likelihood.

Categorical features (e.g., ticket class, railcard type, payment method) were one-hot encoded, while numerical features were kept or standardised as required.

Handling Class Imbalance

The dataset showed significant class imbalance; The majority of journeys were "On Time" while disruptions were relatively rare. To address this, the following dual-method approach was applied:

- **SMOTE (Synthetic Minority Over-sampling Technique)** was used to synthetically generate new examples of the minority class.
- **Class weighting** (or `scale_pos_weight` for XGBoost) adjusted model sensitivity to misclassification of the minority class.

This technique was informed by literature such as Zhang et al. (2019) and Abdelhamid and Desai (2024), who found that combining techniques like SMOTE with model-based weighting yielded better recall and F1-scores in imbalanced scenarios.

Model Pipelines and Tuning

Each model was implemented using an `ImbPipeline` to integrate SMOTE oversampling, preprocessing, and classification. GridSearchCV was used for hyperparameter tuning with 5-fold cross-validation.

Logistic Regression (SMOTE + Class Weights)

Pipeline: SMOTE → StandardScaler → LogisticRegression (with `class_weight='balanced'`)

Hyperparameters Tuned:

- Regularization strength $C \in \{0.01, 0.1, 1, 10\}$
- Penalty $\in \{\ell_1, \ell_2\}$

Best parameters: $C = 0.01,$ penalty $= \ell_1$

Using L1 regularization (also known as Lasso), the model eliminated irrelevant features and performed well on the imbalanced data with minimal overfitting.

Random Forest (SMOTE + Class Weights)

Pipeline: SMOTE $\rightarrow$ RandomForestClassifier (with `class_weight='balanced'`)

Hyperparameters Tuned:

- $n\_estimators \in \{100, 200\}$
- $max\_depth \in \{None, 10, 20\}$
- $min\_samples\_split \in \{2, 5\}$

Best parameters: $n\_estimators = 100,$ $max\_depth =$ None, $min\_samples\_split = 5$

The model's best configuration allowed full-depth trees, implying that the structured feature data set benefited from more granular decision paths. This aligns with the findings of Li et al. (2021) who reported strong RF performance in classifying delay severity thresholds.

XGBoost (SMOTE + Scale Pos Weight)

Pipeline: SMOTE $\rightarrow$ XGBClassifier (with `scale_pos_weight = 5`)

Hyperparameters Tuned:

- $n\_estimators \in \{100, 200\}$
- $max\_depth \in \{3, 5, 7\}$
- $learning\_rate \in \{0.01, 0.1, 0.2\}$

Best parameters: $n\_estimators = 200,$ $max\_depth = 7,$ $learning\_rate = 0.2$

XGBoost required a higher tree depth and learning rate, demonstrating the model's ability to learn complex interactions. This is consistent with Chen et al. (2024), who noted the robustness of XGBoost in operational settings with noise and irregularities.

Model Validation

After building and tuning the selected classification models, their performance was evaluated. This validation step was important to confirm whether the models generalised well to unseen data and whether their predictions were statistically reliable. Especially given the imbalanced nature of the dataset, where only a small percentage of journeys were actually disrupted.

To ensure a balanced evaluation across both classes, the following metrics were considered:

- **Accuracy** – Overall correctness of predictions
- **Precision** – Correctly predicted disruptions out of all predicted disruptions
- **Recall** – Proportion of actual disruptions correctly identified
- **F1-score** – Combined value of both precision and recall
- **Matthews Correlation Coefficient (MCC)** – A robust metric for imbalanced classification

These metrics were reported for both classes (On Time / Disrupted), with macro-averaged scores and MCC used to summarize overall performance.

Table 2: Comparison of Classification Models Across Evaluation Metrics

Metric	Class	Logistic Regression	Random Forest	XGBoost
Precision	On Time	0.92	0.94	**0.96**
	Disrupted	0.33	**0.55**	0.40
Recall	On Time	0.83	**0.92**	0.83
	Disrupted	0.54	0.64	**0.76**
F1-score	On Time	0.87	**0.93**	0.89
	Disrupted	0.41	**0.59**	0.52
Accuracy	—	0.79	**0.88**	0.82
F1-score (Macro Avg)	—	0.64	**0.76**	0.70
Matthews Corr. Coefficient	—	0.30	**0.53**	0.46

Logistic Regression (LR) served as a baseline model. It performed decent on the majority class (Not Disrupted) but struggled with disrupted journeys, achieving the lowest F1-score (0.41) for that class. Its overall MCC 0.30 illustrates limited balance in its prediction.

Random Forest (RF) displayed the most balanced performance. It achieved the highest macro-average F1-score (0.76) and MCC (0.53), implying a strong ability to distinguish both classes. This supports Li et al. (2021), who identified RF as more accurate in predicting delay severity.

XGBoost (XGB) demonstrated the highest recall for disrupted journeys (0.76), capturing more

disruptions than the other models. However, this also resulted in a lower precision (0.40) for that class, indicating more false positives. This aligns with the findings of Chen et al. (2024) and Liu et al. (2023) who acknowledged XGB's performance in noisy, complex domains.

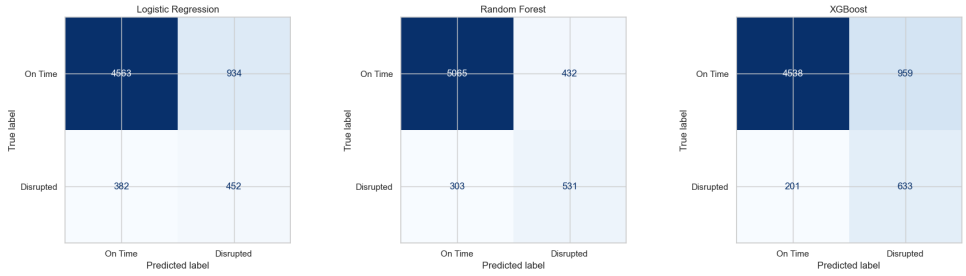

Figure 8: Confusion Matrices for LR, RF, and XGBoost (Disrupted = Class 1)

Figure 8 visualizes how each model distributed predictions. RF showed balanced performance across both classes. XGB achieved more true positives for disruptions, while LR missed the most.

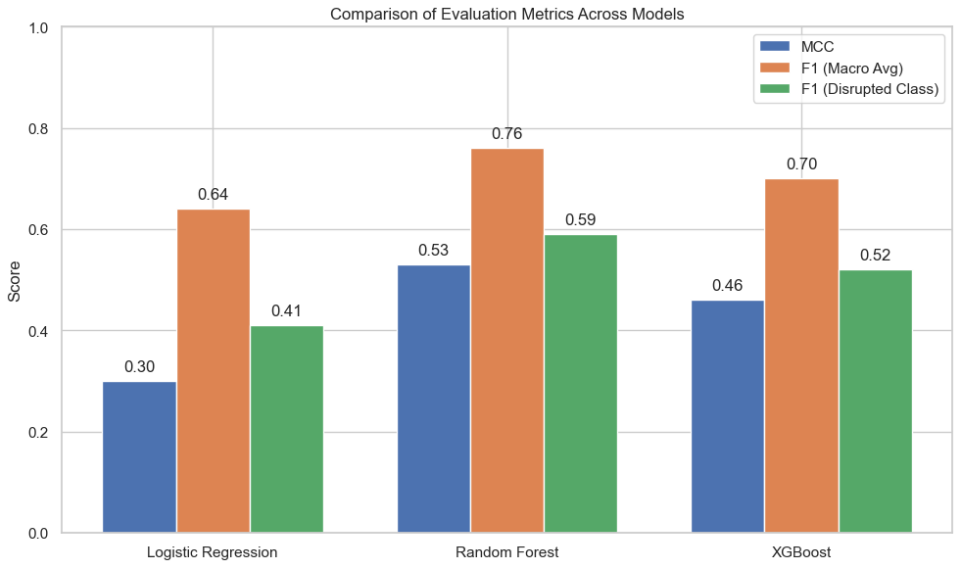

Figure 9: Comparison of Macro F1-score and MCC Across Models

Figure 9 confirms that RF maintained strong and consistent performance across both evaluation metrics. While XGB performed well, RF offered the best balance of precision and recall for real-world deployment.

Conclusion: Random Forest is the most suitable model of the three tested, having the best trade-off between detecting disruptions and avoiding false positives. It also balances precision and recall across classes and scores highest on both F1-macro and MCC.

Analysis & Recommendations

Expected Model Behaviour

As expected, ensemble models outperformed the baseline Logistic Regression (LR), which struggled particularly with disrupted journeys. Table 2 clearly illustrates that Random Forest (RF) achieved the highest F1-macro score (0.76) and Matthews Correlation Coefficient (MCC) of 0.53, outperforming both LR and XGBoost (XGB). These metrics prove RF is capable of maintaining balance between precision and recall, and is consistent with what was researched by Li et al. (2021) and Sarhani and Voß (2024).

Insights from Class-Level Metrics:

- **Logistic Regression:** While LR achieved high precision (0.92) on the majority class (on-time journeys), it underperformed on the minority class (f1-score = 0.41), with an MCC of only 0.304. Demonstrating its limited ability to generalise disruption patterns under imbalance.
- **Random Forest:** RF provided more balanced and interpretable results, achieving better recall (0.64) and F1-score (0.59) for disruptions while keeping a high accuracy. This balance is important in transport applications where false negatives (missed disruptions) can have significant operational consequences.
- **XGBoost:** XGB performed the best in recall for disruptions (0.76), which means it correctly flagged more disrupted journeys. However, its lower precision (0.40) led to more false positives, slightly reducing reliability in its disruption alerts. The MCC of 0.456 places it between LR and RF.

Unexpected Observations

There was not any anomalies with encountered while modelling , but a key insight emerged during feature filtering where post-event attributes such as `Actual Arrival Time` and `Refund Request` initially led to unrealistically high model accuracy (near 100%). This demonstrated the importance of excluding features that reflect outcomes rather than predictive signals. Once removed, model performance dropped to more realistic levels.

Limitations and Interpretation

Despite the application of SMOTE and class weighting, disrupted journeys remained harder to classify. Precision-recall trade-offs were evident; XGB leaned towards higher recall, LR towards higher precision, and RF achieved the best balance. This further proves the challenge of predicting disruptions in imbalanced datasets. Nevertheless, the use of a hybrid imbalance-

handling pipeline was overall effective especially for ensemble methods when classifying journeys.

Benchmark Alignment

While the dataset used in this study is mock data and does not have any prior published benchmarks, the observed model performance are consistent with findings in related works. No prior literature have used this specific UK rail journey dataset for disruption prediction. However, ensemble classifiers such as Random Forest and XGBoost consistently outperformed simpler models like Logistic Regression across transport analytics research. In particular, Li et al. (2021) and Sarhani and Voß (2024) reported Random Forest's strength in capturing delay patterns, while Chen et al. (2024) and Liu et al. (2023) found XGBoost to be successful in handling nonlinear, noisy, or hybrid (structured and text-based) data environments. Although direct result comparison is not feasible, this indirect comparison with prior research supports the validity of the modelling choices and results observed in this project.

Recommendations:

- Consider incorporating real-time contextual features such as weather conditions, signal failures, or track disruptions to enhance the model's ability to detect and predict delays more accurately.
- Explore advanced modelling approaches like model stacking or anomaly detection to better capture rare disruption events that are often missed by standard classifiers.
- Implement decision threshold tuning or probability-based output interpretation to fine-tune the balance between false positives and false negatives, particularly in real-world operational contexts.
- Maintain strict exclusion of post-journey attributes (e.g., actual arrival time, refund status) during feature engineering to avoid information leakage and ensure fair model evaluation.
- Extend route encoding strategies by incorporating historical reliability metrics, route clustering, or service frequency data to improve the estimation of disruption chances across different routes.

Conclusion

This study aimed to build machine learning models that could succesfully classify train journey disruptions in the UK rail network using a mock dataset from Maven Analytics. The objectives were met successfully, with a proper ML pipeline developed and validated using multiple classifiers. Everything from cleaning and engineering features to training models and tuning

hyperparameters was completed. All three models; Logistic Regression, Random Forest, and XGBoost were implemented with SMOTE and class imbalance handling, and performed reliably, with Random Forest achieving the best overall results.

What Went Well.

The data cleaning and preparation phase was smooth. The data transformation and formatting did not have any major issues. Furthermore, Through exploratory data analysis (EDA) utilising Seaborn for visualisation, helpful information on patterns such as delay categories and route-level disruption were discovered. This helped in effectively interpreting the data and its trends.

What Was Challenging.

The implementation phase was especially challenging. Trying to balance the class distribution using SMOTE and class weighting took the most time and consideration. Moreover, understanding how to properly structure the pipeline and apply tuning inside the cross-validation loop was also quite a learning curve, but in the end a rewarding experience.

What I Learned.

Through this study, I've became very familiar with Python libraries like `scikit-learn`, `seaborn`, and `imblearn`. I learned how to apply SMOTE, how to build robust pipelines, and how to choose and evaluate appropriate metrics like F1-score and MCC for imbalanced binary classification tasks.

Limitations.

One major limitation encountered was the absence of research using the same mock dataset. While this made benchmarking difficult, I was able to rely on related transport delay prediction literature to justify and guide model selection. Additionally, because the dataset was originally built more for business intelligence, extra care had to be taken in engineering features to make them suitable for machine learning.

Project Success.

I believe the goal was achieved. The selected models were well-documented and widely used, and there was plenty of community and academic guidance online to support their implementation. The final Random Forest and XGBoost models performed in line with expectations.

Future Work.

There is still more that could be explored. For example, alternative strategies for handling class imbalance (e.g., cost-sensitive learning or threshold tuning) could be tested. Additional mod-

els such as LightGBM or stacking ensembles could be applied for further performance gains. Lastly, incorporating real-time features such as weather and signalling data could improve predictive accuracy for real-world deployment.

Bibliography

Abdelhamid, M., & Desai, A. (2024). Balancing the scales: A comprehensive study on tackling class imbalance in binary classification. http://arxiv.org/abs/2409.19751

Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PLoS ONE*, *12*. https://doi.org/10.1371/journal.pone.0177678

Chen, X., Ma, Z., & Sun, W. (2024). Incident delay prediction in urban railway systems: Methodology review and exploratory comparative analysis. *Transportation Research Record*. https://doi.org/10.1177/03611981241252831

Gao, X., Xie, D., Zhang, Y., Wang, Z., He, C., Yin, H., & Zhang, W. (2025). A comprehensive survey on imbalanced data learning. http://arxiv.org/abs/2502.08960

Laifa, H., Khcherif, R., & Ghezala, H. B. (2022). Predicting trains delays using a two-level machine learning approach. *International Conference on Agents and Artificial Intelligence*, *3*, 737–744. https://doi.org/10.5220/0010898300003116

Li, Z. C., Wen, C., Hu, R., Xu, C., Huang, P., & Jiang, X. (2021). Near-term train delay prediction in the dutch railways network. *International Journal of Rail Transportation*, *9*, 520–539. https://doi.org/10.1080/23248378.2020.1843194

Liu, Q., Wang, S., Li, Z., Li, L., Zhang, J., & Wen, C. (2023). Prediction of high-speed train delay propagation based on causal text information. *Railway Engineering Science*, *31*, 89–106. https://doi.org/10.1007/s40534-022-00286-x

Sarhani, M., & Voß, S. (2024). Prediction of rail transit delays with machine learning: How to exploit open data sources. *Multimodal Transportation*, *3*. https://doi.org/10.1016/j.multra.2024.100120

Tiong, K. Y. (2024). *Data-driven train delay prediction* [Doctoral Thesis (compilation)]. Lund University Faculty of Engineering, Technology, Society, Transport, and Roads [Faculty of Engineering, LTH].

Zhang, W., Ramezani, R., & Naeim, A. (2019). Wotboost: Weighted oversampling technique in boosting for imbalanced learning. http://arxiv.org/abs/1910.07892