MSc program in Business Administration and Data Science

Department of Digitalization

MODELING TRAFFIC DELAY SEVERITY CAUSED BY ACCIDENTS: A MACHINE LEARNING APPROACH

Machine Learning and Deep Learning (CDSCO2041C)

Examiner:
Somnath Mazumdar

Students:
Eduard Aguado (176199)
Luca Giovanni Gudi (175880)
Marco Sburlino (176186)
David Yoshio Uraji (175879)

Number of pages: 15
Character count: 31.156
Submission date: 16-05-2025

Abstract

This project investigates the use of machine learning to classify the severity of traffic delays caused by roadway accidents based on features available at the time of the incident. The problem addressed is the need for timely identification of high-impact events to support traffic management and routing decisions. The research question concerns how accident-related traffic delay severity can be predicted based on real-time features, with a focus on minimizing false negatives for high-severity cases. Concepts applied include supervised classification, class balancing, feature engineering, and model validation. The analysis is based on the US Accidents dataset containing over 7.7 million records, which was cleaned, binarized, balanced, and used to train four models. Histogram-Based Gradient Boosting achieved the highest recall at 0.79, outperforming Random Forest, Logistic Regression, and Multilayer Perceptron, which showed higher accuracy but lower sensitivity to severe cases. These results suggest that HGBoost is best suited for applications where the accurate identification of high-severity delays is prioritized. It is recommended as the preferred model when recall is the primary objective and training efficiency is also relevant.

Keywords: Traffic Delay Modelling; Binary Classification; Logistic Regression; Random Forest; HGBoost; Multilayer Perceptron

Contents

1 Introduction

Traffic congestion is a persistent challenge across the United States, particularly in large metropolitan areas. In cities with more than three million residents, the average annual delay per person reaches 84 hours, reflecting the widespread impact of traffic disruptions on daily life (Texas A&M Transportation Institute, 2021). This problem is closely linked to the country's high reliance on private vehicles, which account for 87 percent of all daily trips and are used by over 90 percent of commuters (U.S. Bureau of Transportation Statistics, 2017).

One of the most significant contributors to traffic congestion is roadway accidents. According to the Federal Highway Administration (2010), accidents are responsible for approximately 25 percent of all delays nationwide. These incidents not only slow traffic but also affect the broader efficiency of urban transportation systems and generate considerable economic costs (Weisbrod et al., 2003). Addressing their impact requires timely and accurate information about the likely severity of disruption.

This project explores the use of machine learning models to classify the expected severity of traffic delays based on conditions observed at the time of an accident. These include temporal patterns, environmental and weather conditions, and road infrastructure characteristics that may influence traffic impact. By identifying the likely severity early, such models can support more responsive navigation and traffic control systems. Services like Google Maps have already integrated predictive approaches to inform routing decisions and reduce travel time (Lau, 2020). This project contributes to such applications by training models specifically to reduce the risk of severe cases being overlooked. In this context, the cost of misclassifying a severe delay as non-severe is considered higher than the opposite. It is therefore preferable for users to receive occasional false warnings rather than face unexpected major disruptions.

2 Related Work

In response to the large burden of traffic congestion, several studies have explored how data-driven methods can improve transportation systems. A key area of focus has been the application of machine learning techniques to detect traffic patterns and forecast congestion. These efforts aim to support real-time decision-making, improve public safety, and enable transportation authorities to allocate resources more effectively.

A foundational contribution in this domain is the US Accidents dataset introduced by Moosavi, Samavatian, Parthasarathy, and Ramnath (2019) which includes over 7.7 million accident reports collected across the United States between 2016 and 2023. The dataset offers extensive contextual features such as weather conditions, road infrastructure, and proximity to points of interest from OpenStreetMap. Its scale and detail have made it a valuable resource for developing predictive models in traffic ana-

lytics. For example, Moosavi, Samavatian, Parthasarathy, Teodorescu, and Ramnath (2019) proposed the Deep Accident Prediction model built upon the dataset which significantly outperformed classical machine learning approaches when predicting rare accident events. In their paper, the authors also highlighted the benefit of deep neural networks in capturing complex feature interactions and long tail event patterns which traditional models often fail to detect.

While this work highlights the value of large-scale datasets for accident prediction, other research in the field has focused on a broader range of modeling challenges. For instance, Deekshetha et al. (2022) concentrated on congestion forecasting by preprocessing traffic data into hourly intervals and training regression models to estimate congestion levels at key junctions. Their work emphasized the importance of real-time interpretability through graphical output which aligns with the operational needs of traffic authorities.

Further research has centered on benchmarking the effectiveness of various classifiers. Deng (2025) compared a range of tree-based and probabilistic models including XGBoost, Random Forest, Extremely Randomized Trees, and Gaussian Naive Bayes. Although the Gaussian Naive Bayes model achieved nearly perfect accuracy under normal conditions, the use of default hyperparameters and the lack of robustness testing raise questions about its real-world reliability. This illustrates an important gap in the literature regarding the role of hyperparameter tuning and model validation in achieving generalizable performance.

The applicability of machine learning to real-world traffic environments has also been tested in region-specific contexts. Hammoumi et al. (2025) used approximately 10,000 Waze accident reports to analyze traffic congestion in Casablanca. Their study explored multiple classification techniques and found that Random Forest consistently performed best with an accuracy of 96% and an AUC of 0.997. The results highlight the potential of ensemble methods for modeling structured transportation data even in settings with relatively limited samples.

Although these studies have advanced the field in important ways, much of the existing literature has concentrated on predicting either the likelihood of traffic events or general levels of congestion. Few studies have addressed the question of how severely an incident will impact traffic flow. This project aims to fill that gap by classifying the severity of traffic delays using only the features available at the time of the incident. This approach allows for a more precise estimation of disruption levels and contributes to more targeted and timely decision making in traffic control and routing systems.

3 Methodology

The overall methodology of this study follows the structured approach shown in Figure 1. Starting from the raw US Accidents dataset, the process includes data preprocessing, exploratory analysis, and

preparation steps such as feature engineering and class balancing. This is followed by model configuration, including selection of algorithms, evaluation metrics, and tuning strategies, and concludes with training and performance evaluation. Each step was designed to address the specific characteristics of the dataset, avoid data leakage, and ensure robust model validation.

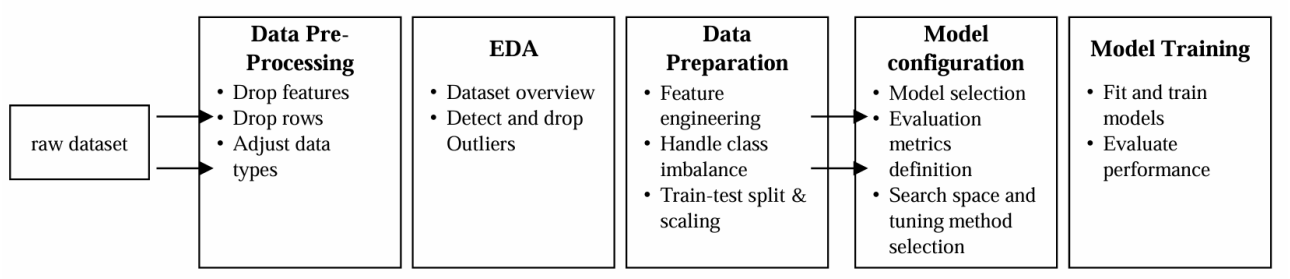

Figure 1: End-to-end modelling pipeline

3.1 Dataset Description

This US Accidents dataset by Moosavi, Samavatian, Parthasarathy, and Ramnath (2019), as already described in Section 1, is a large-scale compilation of traffic accident reports recorded across the United States between February 2016 and March 2023. The dataset was sourced from multiple channels, including traffic incident APIs, local and state transportation departments, and traffic sensor streams. It contains over seven million individual entries, each capturing a range of variables related to the time and location of the accident, weather conditions, and road infrastructure near the accident. The primary outcome variable, `Severity`, reflects the impact of an accident on traffic flow, measured on a four-point ordinal scale. A value of one indicates minimal disruption, while a value of four corresponds to significant delays.

3.2 Preprocessing and Exploratory Data Analysis (EDA)

The preprocessing of the dataset started with an initial inspection of the dataset and the removal of duplicate entries to ensure that each observation represented a distinct incident. Categorical variables with only one unique value or categories deemed irrelevant for predictive purposes, such as coordinates and the `Description` field, were excluded from further analysis. Additional columns, including `Weather_Condition`, were dropped due to redundancy with other variables. Rows containing isolated missing values in `City`, `Zipcode`, or `Civil_Twilight` could not be imputed with sufficient confidence and were therefore removed. In accordance with the dataset documentation, missing entries in `Precipitation(in)` and `Wind_Speed(mph)` were interpreted as an absence of that weather condition (either precipitation or wind) and replaced with zero. Lastly, the target variable `Severity` was converted to categorical format in preparation for the classification task.

After these initial preprocessing steps, the dataset contained 7,703,274 rows and 26 columns. Missing values were detected in the four features `Pressure(in)`, `Temperature(F)`, `Humidity(%)`, and `Visibility(mi)`, ranging from around 135,000 to around 173,000 absent entries. A distribution plot of the target variable `Severity` revealed a pronounced class imbalance, with most observations labeled as severity level two. This imbalance raised concerns about potential prediction bias toward the dominant class, highlighting the importance of addressing it. Possible strategies considered included synthetic oversampling techniques such as SMOTE, as well as a binary reformulation of `Severity` by grouping levels one and two as lower-impact cases and levels three and four as higher-impact cases. Further steps to manage this issue were taken later in the process and are discussed in Section 3.3.

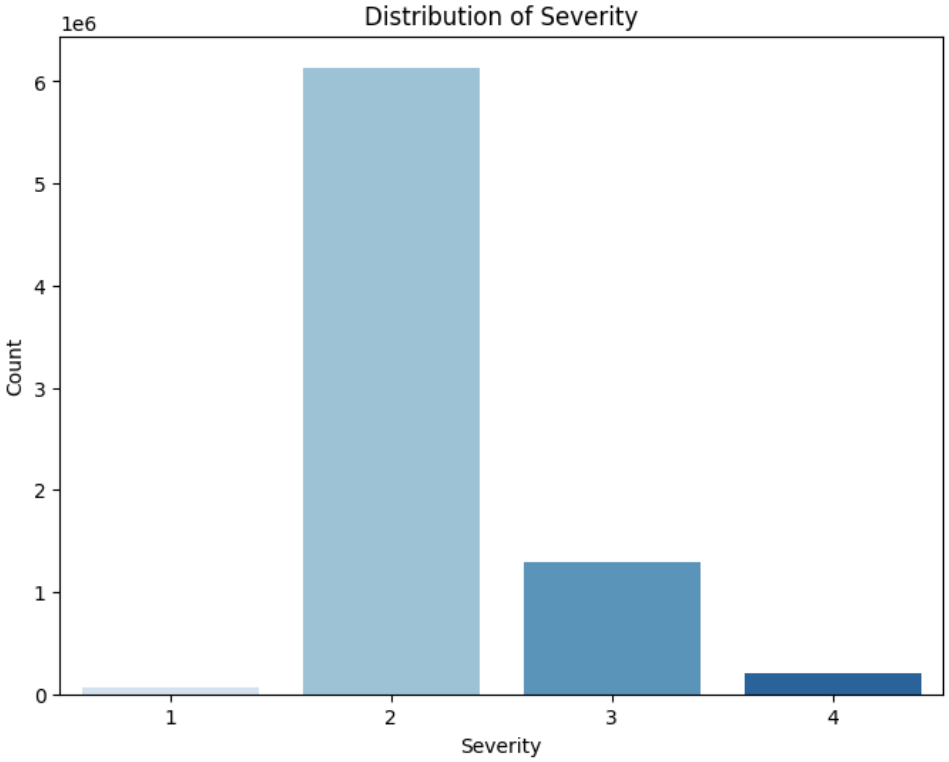

Figure 2: Initial Distribution of Severity Levels (1–4)

Histograms and boxplots revealed skewness, irregular patterns, and the presence of outliers. Features such as `Wind_Speed(mph)` and `Precipitation(in)` displayed highly skewed distributions, with most values concentrated at the lower end and a few extreme cases extending into the upper range. Several of these outliers appeared unrealistic or disproportionate and could introduce noise during model training. The 1.5 IQR method was applied, removing values outside one and a half times the interquartile range. The revised plots showed reduced skewness and tighter clustering, improving interpretability and minimizing the risk of model distortion from extreme values.

The correlation matrix revealed expected patterns among numerical variables, such as a negative correlation between Humidity(%) and both Visibility(mi) and Temperature(F). However, most variables exhibited only weak or negligible correlations with one another, suggesting that multicollinearity is unlikely to affect later modeling stages. Additionally, correlations between most features and Severity were minimal, raising concerns regarding the limited explanatory power of the features with the target variable. To explore this further, each numerical feature was plotted against Severity using boxplots, providing a more granular view of how values shifted across severity levels. While some trends were observed, such as slightly lower visibility and pressure in higher severity categories, the differences were modest, and distributions remained largely overlapping. No individual feature exhibited a clear or consistent pattern, reinforcing the assumption that outcomes are likely influenced by interactions among multiple variables. Supplementary visualizations, including the correlation matrix, histograms, and boxplots of all numerical features, are provided in Appendix A.

3.3 Data Preparation

After the initial data cleaning and exploratory analysis, further preparation was conducted, with particular attention given to addressing class imbalance in the dataset. As part of the feature engineering process, time-based features were extracted from the Start_Time variable, including Hour, DayOfWeek, Month, and Year. Month was then mapped to a new categorical variable, Season, which was one-hot encoded with Fall set as the base category and dropped to prevent multicollinearity. Similarly, the Civil_Twilight variable was transformed into a binary Daylight indicator (1 for day, 0 for night). Infrastructure-related features were explicitly cast as boolean to ensure consistent interpretation across models. Finally, the State variable was one-hot encoded to avoid introducing ordinal bias and to enable the model to treat each state as a distinct, non-ordered category contributing individually to the target variable.

To manage class imbalance, the target variable Severity was first binarized to simplify the classification task and mitigate the effect of sparsely populated edge classes. Values 1 and 2 were grouped as 0 (non-severe accidents), and values 3 and 4 as 1 (severe accidents), as levels 1 and 4 accounted for only 0.8 percent and 2.6 percent of the dataset, respectively. While this transformation reduced the imbalance, the distribution remained skewed, with 79.5 percent of observations labeled as 0 and 20.5 percent as 1.

To avoid data leakage and support generalizability, the dataset was split into training and test sets before any balancing. The Stratified Shuffle Split method was used to extract 630,000 observations, approximately 20 percent of the dataset, ensuring that class proportions were preserved in the test set. After the split, data processing pipelines were implemented separately for the training and test sets. Missing values were imputed using the mean for numeric variables and the mode for boolean ones,

and scaling was performed using statistics computed independently within each subset.

To address the remaining class imbalance in the training data, the majority class was undersampled using the `RandomUnderSampler` method, reducing the number of non-severe cases to 1,900,000 in proportion to the 1,089,708 severe cases. Given the dataset's size and the similarity among many observations, this approach was considered preferable to synthetic oversampling techniques such as SMOTE, as it reduced the risk of overfitting and helped maintain the integrity of the training data.

The final train-to-test split ratio was 82.6 percent to 17.4 percent, closely aligned with the commonly used 80 to 20 convention. As shown in Figure 3, the original class distribution in the test set was preserved, supporting generalizability and enabling reliable model evaluation.

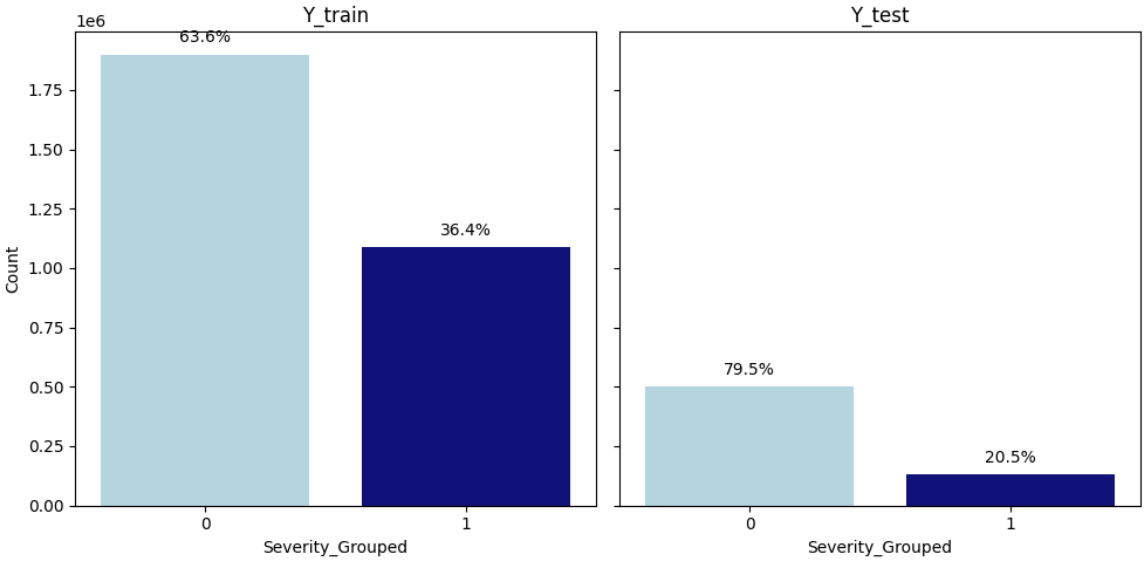

Figure 3: Distribution of Grouped Severity (Train and Test dataset)

3.4 Model Selection

With the data prepared and the target variable balanced, the next step involved training and evaluating a set of predictive models. Four models were selected to classify the severity of accidents in terms of traffic delay, each representing a distinct binary classification strategy. The first model was a logistic regression, which served as a baseline for comparative evaluation. Additionally, two ensemble-based models were implemented: a bagging-based method (Random Forest) and a boosting-based method (Histogram-Based Gradient Boosting). Finally, the fourth model, based on a multilayer perceptron architecture, was trained from scratch and shared the same hyperparameter optimization approach.

3.4.1 Logistic Regression

The logistic regression was chosen as the baseline method due to its simplicity and strong theoretical foundations. As a probabilistic classifier, it is well suited for binary outcome variables (Géron, 2023), fitting perfectly with our research question and objectives. After defining `Grouped_Severity` as the target variable and the rest of the features as explanatory, the model was trained. Given the ease of the model, the design choice only relied on choosing an approach to efficiently handle a large dataset. For this specific reason, the maximum number of iterations allowed for the solver to converge to an optimal solution was set to 1000 to reduce the risk of convergence warning.

3.4.2 Random Forest

The first ensemble-based classification method applied was Random Forest, which constructs multiple decision trees using randomly drawn subsets of input features. Each tree is trained independently, and final predictions are made through majority voting. This bagging approach reduces variance, mitigates overfitting, and performs well on high-dimensional or noisy data, making it suitable for our dataset. Additionally, the model is scalable and provides internal feature importance estimates that support interpretability (Breiman, 2001).

Before tuning, the hyperparameter space was defined. The number of trees was set between 100 and 150 to balance predictive stability and computation time. Maximum depth ranged from 10 to 18 to control tree complexity. The minimum samples to split a node and per leaf were set between 2–6 and 3-6, respectively, to enhance generalization on the results. Additionally, bootstrap sampling was toggled to test the impact of resampling. Finally, the number of features considered at each split was also varied to assess the effect of feature selection strategies, including the square root, logarithm base 2, and all available features.

The model's hyperparameters were optimized using the Randomized Search method, which requires significantly fewer computational resources than a standard grid search (Tibshirani, 1996). A stratified 2-fold cross-validation was applied to mitigate overfitting and improve the reliability of generalization estimates (Hastie, 2009). To further reduce runtime while maintaining data diversity, hyperparameter tuning was performed on a 30 percent subset of the training data. The search space included the number of estimators, maximum tree depth, minimum samples for splits and leaf nodes, bootstrap usage, and the feature selection strategy.

The randomized search tested 20 parameter combinations and optimized for the weighted F_1-score, which accounts for class imbalance. The best configuration included 289 trees, a maximum depth of 25, a minimum of 7 samples to split a node, and 4 samples per leaf. Bootstrap sampling was enabled, with no feature limit applied at each split.

3.4.3 Histogram-based Gradient Boosting

The Histogram-Based Gradient Boosting Classifier (HGBoost) was the second ensemble method used. It is a variant of traditional Gradient Boosting Decision Trees (GBDT) that improves memory efficiency and computational performance through feature binning (Géron, 2023).

Before hyperparameter tuning, class weights were assigned to address class imbalance, giving class 1 (the minority class) four times more importance than class 0. This weighting was essential, as misclassifying the minority class is more costly than misclassifying the majority class. The learning rate was tested across five equally spaced values between 0.05 and 0.1, balancing model performance and the number of boosting stages. The number of trees was set between 100 and 200, sufficient to avoid underfitting while limiting the risk of overfitting. To further optimize model complexity, `max_depth` was tested between 5 and 10 to balance expressiveness and generalization. The `min_samples_leaf` parameter ranged from 50 to 100 to ensure each leaf contained enough data to prevent overfitting. Lastly, the maximum number of leaf nodes per tree (`max_leaf_nodes`) was explored between 31 and 63 to manage tree complexity.

Hyperparameters were optimized using the Halving Random Search Cross-Validation (HRS-CV) method, which combines random sampling with successive halving. A reduction factor of 2 was used, retaining only half of the configurations at each iteration as more resources were allocated to the top-performing ones. A stratified 3-fold cross-validation was applied to mitigate overfitting and improve the reliability of generalization estimates (Hastie, 2009). To prevent memory overflow, a common issue with complex models on large datasets, parallel jobs were limited to two. Model performance was evaluated using the weighted F_1-score metric, which prioritizes recall while accounting for the precision–recall trade-off and class imbalance.

3.4.4 Multilayer Perceptron (MLP)

The final model implemented was a feedforward neural network based on the Multilayer Perceptron (MLP) architecture. MLPs are fully connected networks capable of capturing complex non-linear relationships and are well suited for classification tasks involving structured input data (Géron, 2023). For binary classification, a single output neuron with a logistic activation function outputs the probability of the positive class, making the model appropriate for predicting traffic delay severity.

The model was implemented using `MLPClassifier` from `scikit-learn`, with a maximum of 500 training iterations and a fixed random seed to ensure reproducibility. Since this implementation does not support class weighting, the model was trained on the balanced version of the dataset to ensure equal representation of both classes.

Before training, the hyperparameter space was defined to establish the neural network configuration. Particularly, the model is composed by two hidden layer, the first counts 100 neurons while the sec-

ond 50. Although, the MLP model can handle complex problems using one hidden layer, two hidden layer were choosen because deep networks are far more parameter-efficient than shallow ones (Géron, 2023). Furthermore, two activation functions were tested, `tanh` and `relu`, to assess non-linearity handling. In particular, both were included to make the optimizer pick the most effective one depending on the number of hidden layers, the data distribution, and the regularization. The solvers considered were `adam` and `sgd`, representing adaptive and gradient-based optimization strategies. Learning rate strategies included both constant and adaptive options. Regularization strength was controlled through the `alpha` parameter, sampled uniformly between 0.0001 and 0.01. Although log-uniform sampling is often preferable for regularization parameters (Bergstra & Bengio, 2012), uniform sampling was chosen for its practical performance and simplicity.

Hyperparameter optimization was carried out using halving randomized search to explore a wider hyper-parameter space (Géron, 2023). Specifically, each round keeps the top 1/3 of configurations and increases resources by 3. Each configuration was then evaluated with 3-fold cross-validation using the weighted F_1-score as the objective function, balancing precision and recall under class imbalance.

3.5 Evaluation Metrics

Aligned with the use case outlined in Section 1, evaluation metrics were selected to prioritize the accurate identification of severe accidents, meaning the minimization of severe cases incorrectly predicted as non-severe. As minimizing false negatives is critical, recall was chosen as the primary evaluation metric. Precision and recall involve a trade-off: increasing recall typically leads to more false positives, reducing precision, and vice versa. The F_1-score, which requires both metrics to be high, was used to monitor this trade-off and mitigate excessive false positives resulting from recall maximization. These metrics are defined as follows:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{1}$$

$$F_1 = \frac{2}{\text{Recall}^{-1} + \text{Precision}^{-1}} \tag{2}$$

ROC AUC was used as a secondary metric to evaluate the model's ability to rank severe accidents above non-severe ones across all thresholds, regardless of the final decision cutoff. Finally, accuracy was included to assess overall correctness of the model. These metrics are defined as follows:

$$\text{ROC AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d\text{FPR} \tag{3}$$

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}} \tag{4}$$

4 Results

4.1 Model Performance

Table 1 shows the performance of the different models across the 4 targeted metrics. The baseline model, Logistic Regression, achieves the highest F_1-score (0.71), indicating a balanced precision–recall trade-off, but at the cost of recall (0.69), making it less effective at detecting severe cases, our primary evaluation focus. It also reports the lowest accuracy and ROC AUC, reflecting weaker overall and ranking performance.

Model	**Recall**	**F_1-score**	**Accuracy**	**ROC AUC**
Logistic Regression (Baseline)	0.69	0.71	0.68	0.74
Random Forest	0.77	0.52	0.71	0.81
Histogram-Based GBM (HGBoost)	0.79	0.54	0.73	0.82
Multilayer Perceptron (MLP)	0.63	0.54	0.78	0.82

Table 1: Overall evaluation metrics by model

Both Random Forest and HGBoost improve upon the baseline's performance slightly in terms of accuracy, and more notably in recall, which is the primary metric for our objective. Random Forest achieves a recall of 0.77, while HGBoost slightly outperforms it with a recall of 0.79. Additionally, HGBoost reports a slightly higher F_1-score (0.54 vs. 0.52), suggesting a better balance between false positives and false negatives. This indicates that HGBoost reduces missed severe cases compared to both the baseline and Random Forest, without significantly increasing incorrect severe predictions.

Finally, the MLP is the model that correctly classifies the highest number of accidents, with an accuracy of 0.78. However, this performance is achieved at the expense of correctly identifying severe cases, indicating a tendency to favor non-severe classifications. This results in a higher number of false negatives, which is misaligned with the project's goal of prioritizing severe accident detection.

4.2 Discussion Best Model

The evaluation results demonstrate that the Histogram-based Gradient Boosting (HGB) model is the best performer for binary classification of delay severity due to accidents, achieving a recall 0.79 on the test set. Following HGB, the Random Forest model yields an recall of 0.77, the Logistic Regression model achieves 0.69, and the Multilayer Perceptron (MLP) records an recall of 0.63. As mentioned earlier, the project focused more on avoiding misclassification of severe delays. This is based on the assumption that if the model were deployed at scale across major platforms, users would prefer to receive an occasional false warning rather than miss a notification about a major delay. Furthermore, detailed precision, recall, and F_1-scores are presented in Table 2.

	Precision	**Recall**	**F1-score**	**Support**
Non-severe (0)	0.93	0.71	0.81	502401
Severe (1)	0.41	0.79	0.54	127599
Accuracy	-	-	0.74	630000
Macro avg	0.74	0.75	0.74	630000
Weighted avg	0.77	0.74	0.75	630000

Table 2: Classification report for the test dataset

The test set comprises 630,000 accident records, with 502,401 labeled as non-severe and 127,599 as high severity, which may contribute to slightly lower precision for high-severity cases across models due to this imbalance. Confusion matrices in Figure 5 illustrate the predictions against actual labels, revealing a moderate rate of false positives, where low-severity delays are occasionally misclassified as high severity. This is likely influenced by the dataset's imbalance. The HGB model's superior performance is further supported by the ROC, which shows a strong discriminative performance with an area under the curve (AUC) of 0.83. This relatively high value indicates good classification ability, significantly outperforming the random case, represented by the dashed line. In addition to this, the model performs particularly well in the region of the low false positive rate (0.0-0.2), in line with the desired output (optimizing the recall).

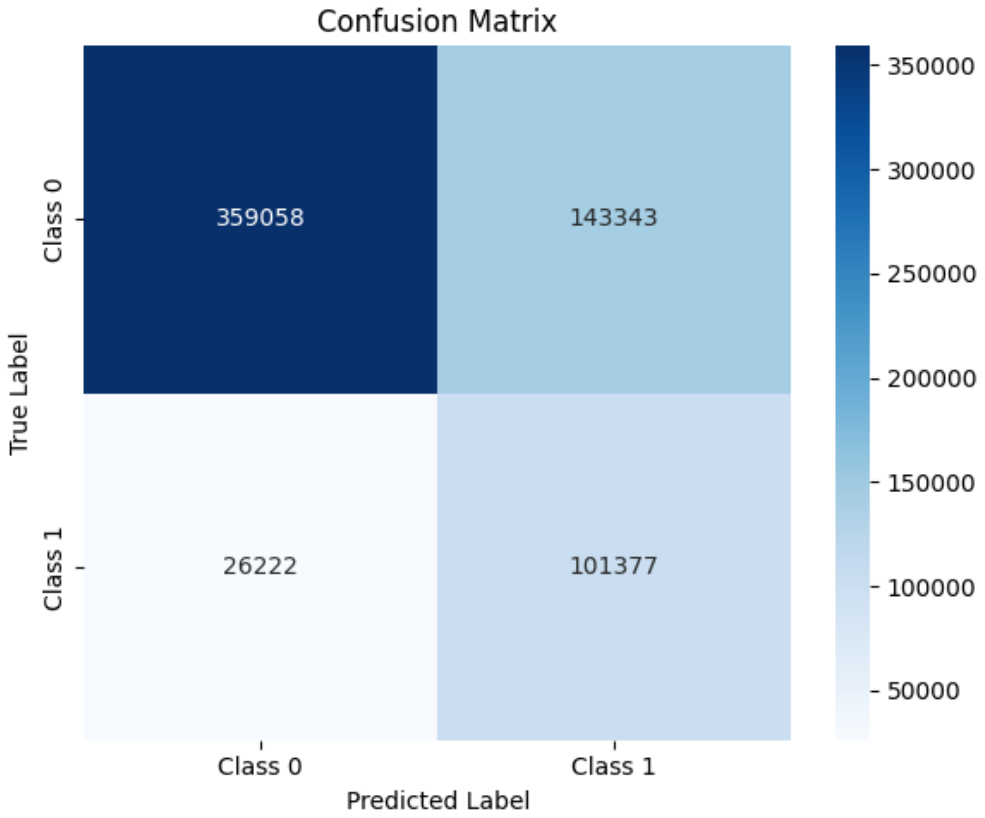

Figure 4: Confusion matrix for HGBoost model

4.3 Model Complexity Analysis

While evaluation metrics such as recall and F_1-score are central to model selection, model complexity and training efficiency are equally important, especially when considering real-world deployment. As shown in Figure 5, training time varied significantly across the models. Logistic Regression, the baseline, is the most efficient, completing training in just 144.3 seconds. Random Forest and MLP are more computationally demanding, requiring 2,652.1 and 3,186.8 seconds respectively. HGBoost offers a balanced trade-off, with a moderate training time of 582.6 seconds while achieving the highest recall (0.79) and a strong ROC AUC (0.82). Although MLP reaches the highest accuracy (0.78), its long training time and relatively low recall (0.63) make it less aligned with our objective of prioritizing severe accident detection. The comparison suggests that HGBoost offers the most practical balance between performance and complexity, outperforming MLP and Random Forest in efficiency without compromising the primary evaluation goal. All models were trained on a machine with 64 vCPUs and 384 GB of memory, without GPU acceleration.

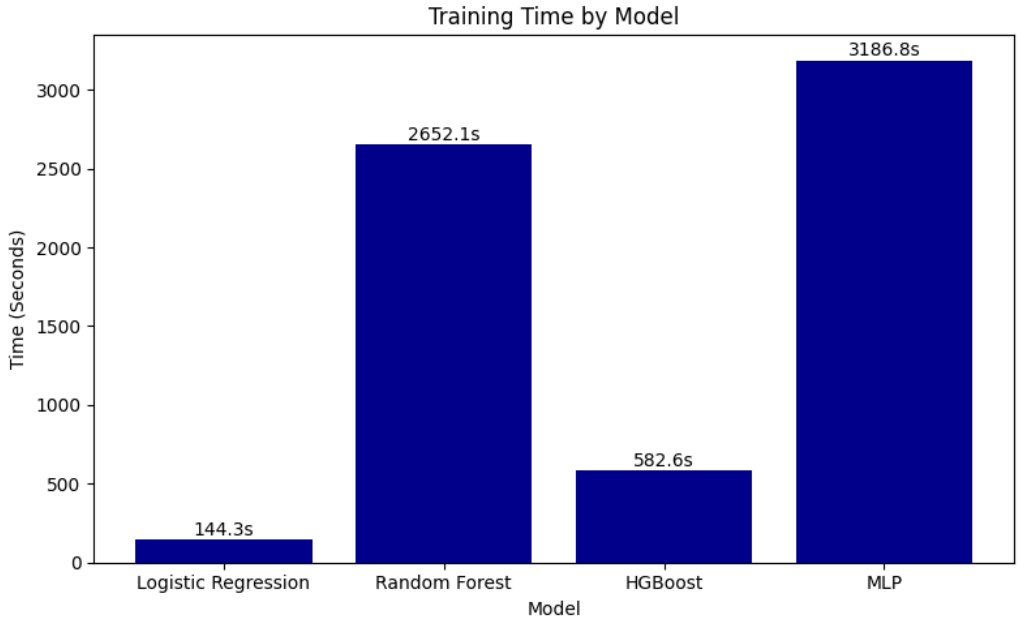

Figure 5: Training time by model (in seconds)

5 Discussion

Overfitting was a primary concern. For each model, it was addressed by comparing classification performance across training and test sets. Accuracy and recall showed no significant variance, indicating good generalization. Precision, however, was consistently lower on the test set due to class imbalance, one of the main challenges in the dataset. As outlined in Section 3.3, this was addressed by balancing the training set while keeping the test set imbalanced to preserve generalizability. This likely increased false positives in the test set, reducing precision and indirectly lowering the F_1-score. Finally, ROC AUC scores remained stable across datasets, suggesting effective generalization of probability rankings.

The project also faced several dataset limitations. As shown in Figure 10 in Appendix A, most features exhibited only weak or negligible correlations with `Severity`, limiting the model's predictive power. Additionally, the target variable is ambiguous, as it is not defined by any specific measurable criteria. These factors suggest that the explanatory power of the features in the dataset may be limited and that the current models may have reached their boundaries. This explanatory power could be expanded through the collection and inclusion of additional contextual features with greater predictive value, such as vehicle type, number of vehicles involved, or driver characteristics (e.g., novice driver).

Regarding the limitations faced, some ethical concerns arised. Including features such as the number of vehicles involved in the accident, driving experience (e.g., novice driver), and driver gender could enhance the model's explanatory power but may also raise privacy risks. As suggested by Mittelstadt

et al. (2016), transparency and accountability are essential for trust in algorithmic decision-making systems, particularly in high-stakes domains such as public transportation and emergency planning. Additionally, in line with the dataset limitations, the underrepresentation of certain states or environmental conditions in the training data may lead to systematically less accurate predictions for those cases. This can result in unequal treatment in real-world applications. As highlighted by Barocas et al. (2023), fairness in machine learning depends not only on overall performance but also on the distribution of errors across subgroups.

6 Conclusion and Future Work

In this study, four models for binary classification of accident-related delay severity were evaluated. Particularly, three machine learning models were deployed, namely: Random Forest, Histogram-based Gradient Boosting (HGB), and Multilayer Perceptron (MLP). All of them were then compared against a baseline Logistic Regression model. Although the benchmark model counts on a better F_1-score than the other models, the HGB is considered the best performing. Indeed, our results indicate that the HGB model outperformed the others, achieving the highest recall of 0.79, likely due to its robustness in handling imbalanced data. In addition to this, evaluating the training time the same model outperforms all others, similar to the logistic regression runtime. These findings suggest that ensemble methods, particularly HGB, are highly effective for this classification task.

The current dataset for traffic delay classification lacks visual data, a key future step is to integrate it with new datasets to provide richer contextual information. A first idea could be starting from the "StreetSurfaceVis" dataset (Kapp et al., 2025) to train a convolutional neural network (CNN) for image recognition to extract features like road conditions from these images. The CNN's outputs can then be merged with the existing dataset to enhance the performance of the current models. Additionally, advanced models such as graph neural networks could be explored to capture spatial road network relationships. In particular, those kind of models are already being used in Financial Fraud Detection since leverage the topological structure of financial transaction networks to find discrepancies.

References

Barocas, S., Hardt, M., & Narayanan, A. (2023). *Fairness and machine learning*. fairmlbook.org. https://fairmlbook.org/pdf/fairmlbook.pdf

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *The journal of machine learning research*, *13*(1), 281–305. https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

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

Deekshetha, H. R., Madhav, A. V. S., & Tyagi, A. K. (2022). Traffic prediction using machine learning. In V. Suma, X. Fernando, K. L. Du, & H. Wang (Eds.), *Evolutionary computing and mobile sustainable networks* (pp. 803–814, Vol. 116). Springer. https://doi.org/10.1007/978-981-16-9605-3_68

Deng, S. (2025). Research on traffic prediction based on machine learning. *Applied and Computational Engineering*, *135*, 195–203. https://doi.org/10.54254/2755-2721/2025.21218

Federal Highway Administration. (2010). Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation. https://ops.fhwa.dot.gov/publications/fhwahop10010/presentation.htm

Géron, A. (2023). *Hands-on machine learning with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.

Hammoumi, L., Farah, S., Benayad, M., Maanan, M., & Rhinane, H. (2025). Leveraging machine learning to predict traffic jams: Case study of casablanca, morocco. *Journal of Urban Management*. https://www.sciencedirect.com/science/article/pii/S2226585625000172

Hastie, T. (2009). The elements of statistical learning: Data mining, inference, and prediction. https://doi.org/10.1007/978-0-387-84858-7

Kapp, A., Hoffmann, E., Weigmann, E., et al. (2025). Streetsurfacevis: A dataset of crowdsourced street-level imagery annotated by road surface type and quality. *Scientific Data*, *12*, 92. https://doi.org/10.1038/s41597-024-04295-9

Lau, J. T. (2020). Google maps 101: How ai helps predict traffic and determine routes. https://blog.google/products/maps/google-maps-101-how-ai-helps-predict-traffic-and-determine-routes/

Mittelstadt, B. D., Allo, P., Taddeo, M., Wachter, S., & Floridi, L. (2016). The ethics of algorithms: Mapping the debate. *Big Data & Society*, *3*(2), 1–21. https://doi.org/10.1177/2053951716679679

Moosavi, S., Samavatian, M. H., Parthasarathy, S., & Ramnath, R. (2019). A countrywide traffic accident dataset. *arXiv preprint*. https://doi.org/10.48550/arXiv.1906.05409

Moosavi, S., Samavatian, M. H., Parthasarathy, S., Teodorescu, R., & Ramnath, R. (2019). Accident risk prediction based on heterogeneous sparse data: New dataset and insights. *arXiv preprint*. https://doi.org/10.48550/arXiv.1909.09638

Texas A&M Transportation Institute. (2021). 2021 urban mobility report. http://mobility.tamu.edu

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *58*(1), 267–288. https://academic.oup.com/jrsssb/article/58/1/267/7027929?login=false

U.S. Bureau of Transportation Statistics. (2017). National household travel survey: Daily travel quick facts. https://www.bts.gov/statistical-products/surveys/national-household-travel-survey-daily-travel-quick-facts

Weisbrod, G., Vary, D., & Treyz, G. (2003). Measuring economic costs of urban traffic congestion to business. *Transportation research record*, *1839*(1), 98–106. https://doi.org/10.3141/1839-10

Appendix

Appendix A: EDA Results

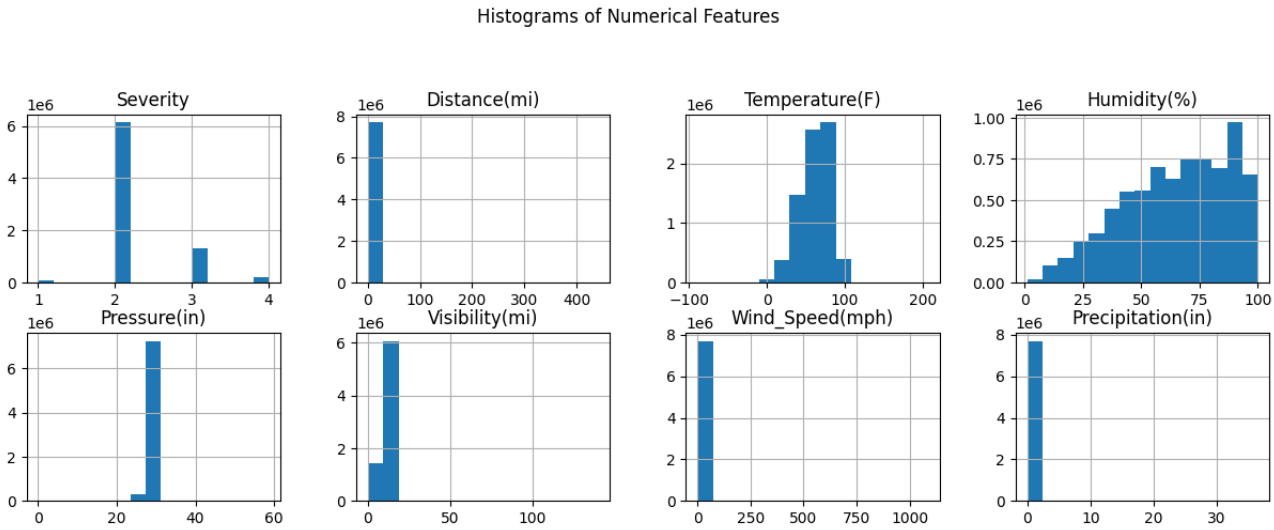

Figure 6: Histograms of Numerical Variables before Outlier Exclusion

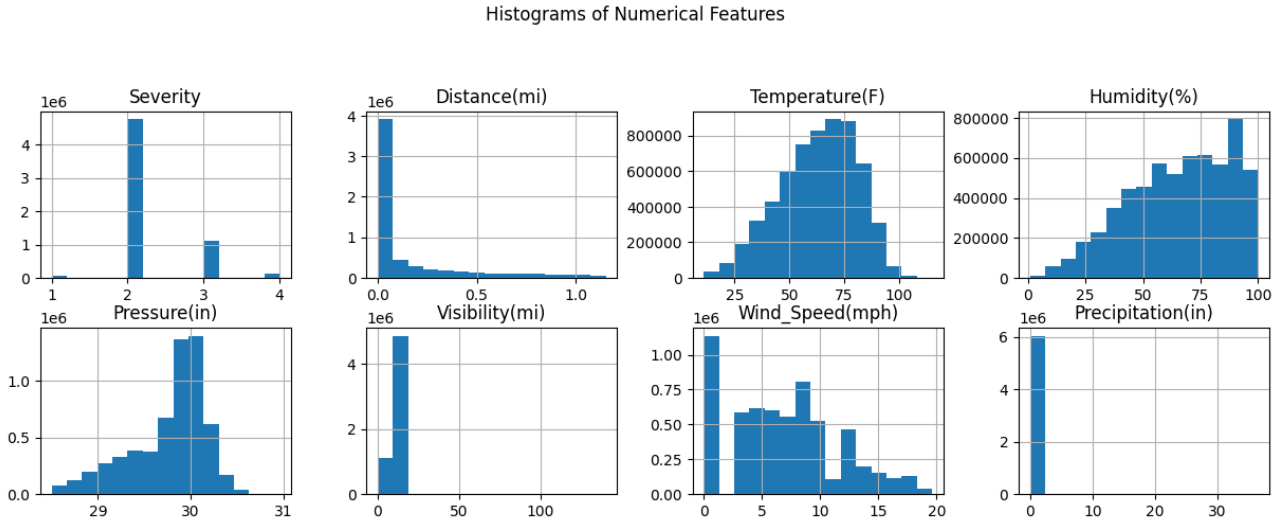

Figure 7: Histograms of Numerical Variables after Outlier Exclusion

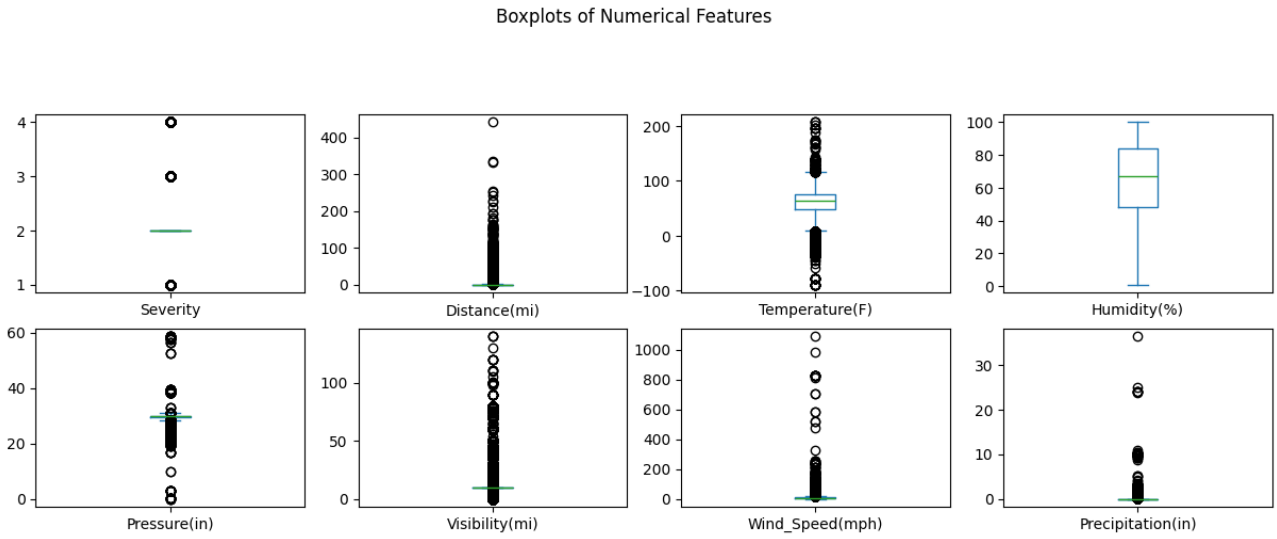

Figure 8: Boxplots of Numerical Variables before Outlier Exclusion

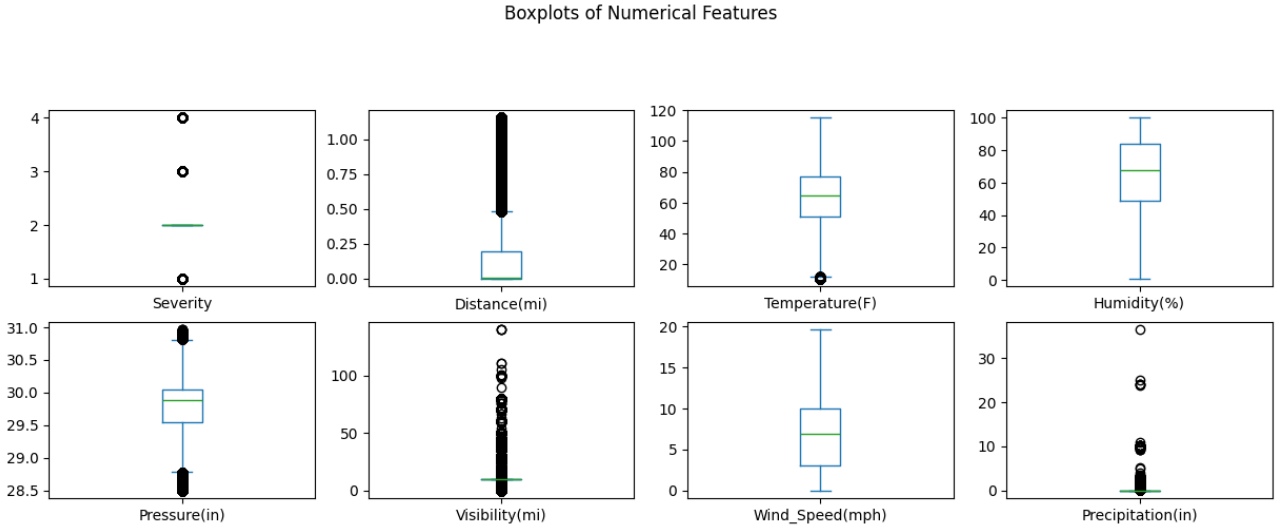

Figure 9: Histograms of Numerical Variables after Outlier Exclusion

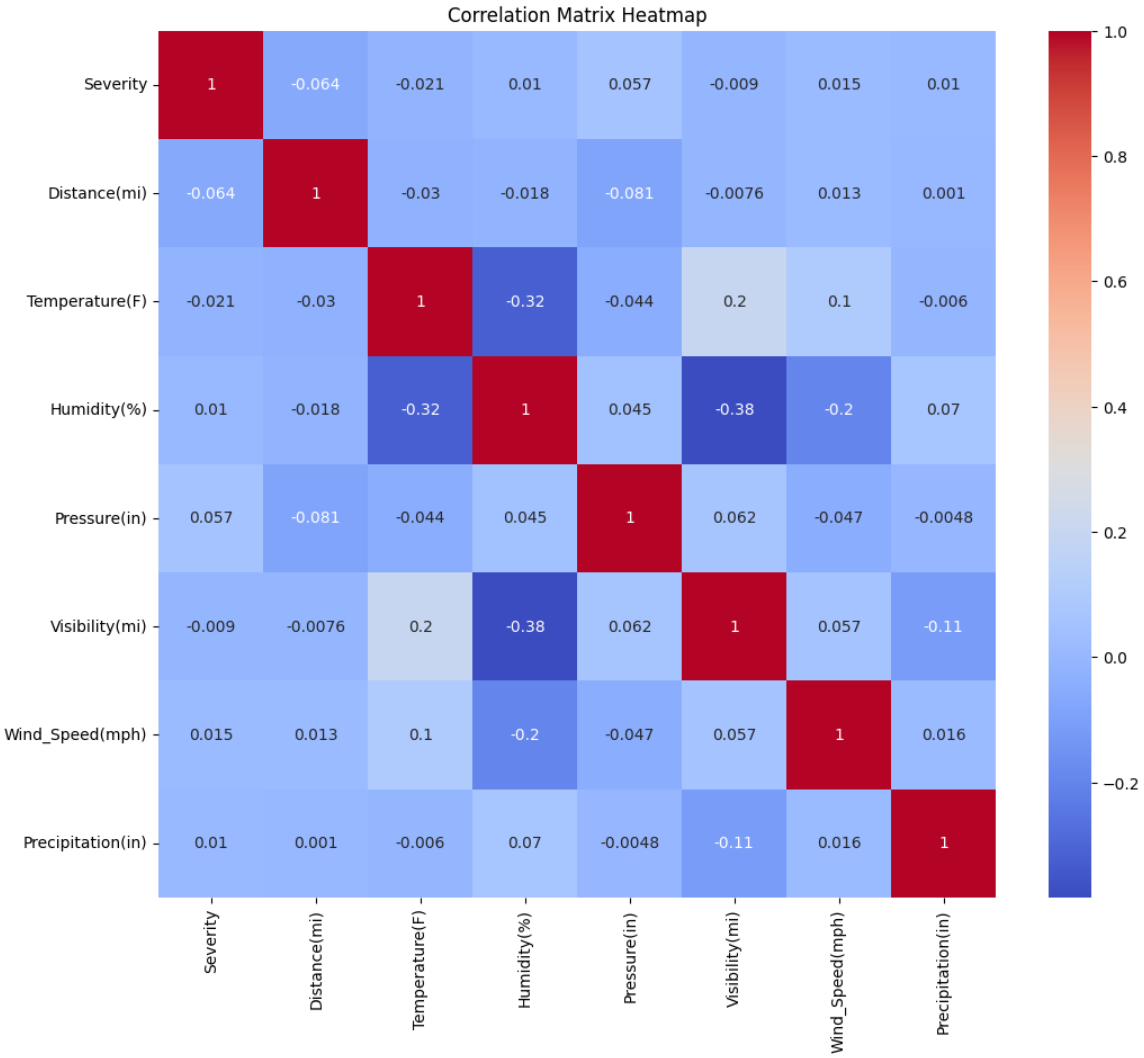

Figure 10: Correlation Matrix Heatmap of Numerical Features

Appendix B: Model Performance Results

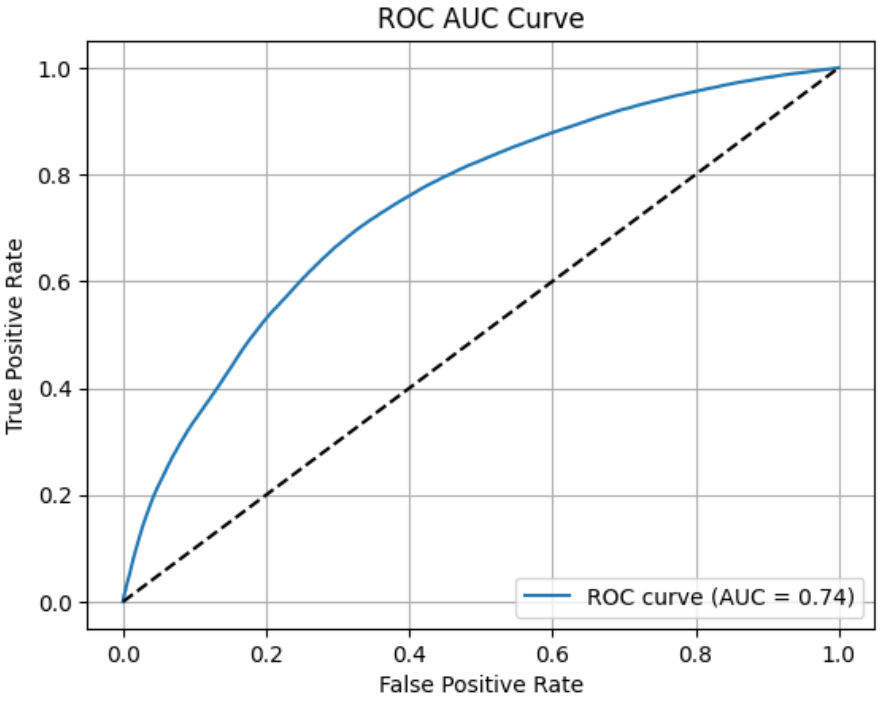

Figure 11: Logistic Regression ROC Curve

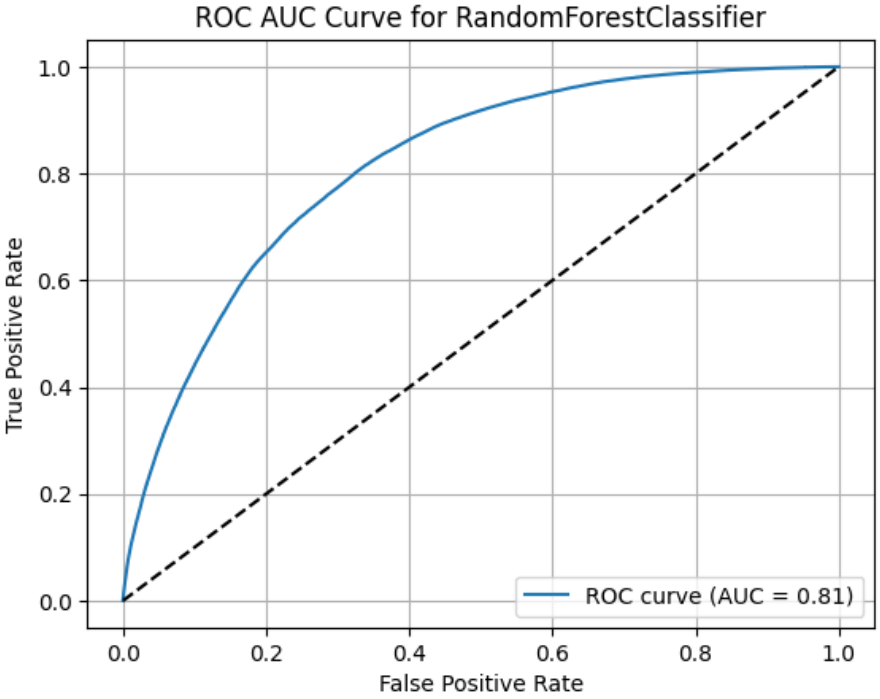

Figure 12: Random Forest ROC Curve

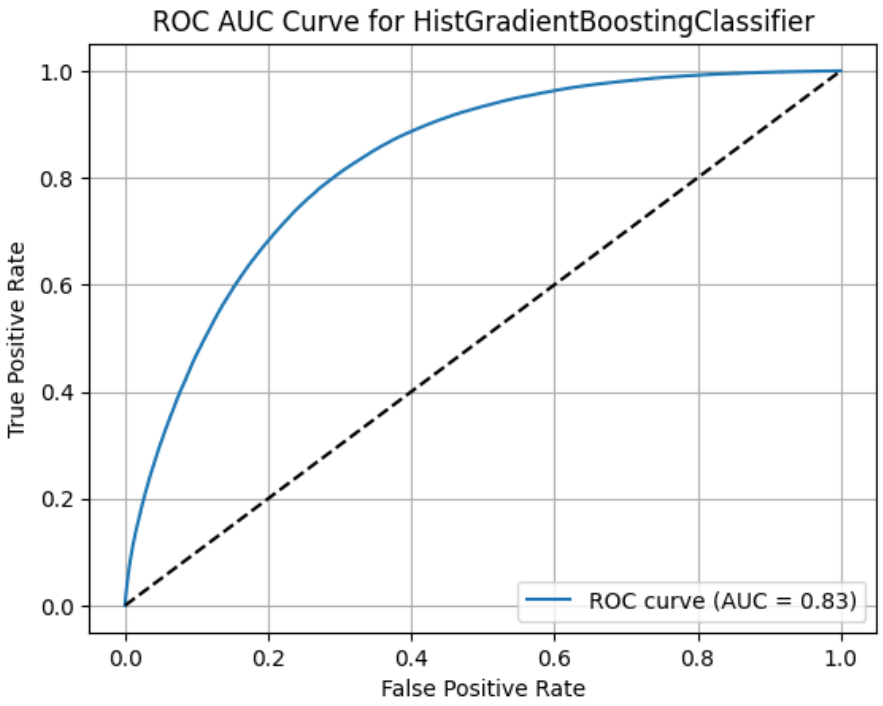

Figure 13: HGBoost ROC Curve

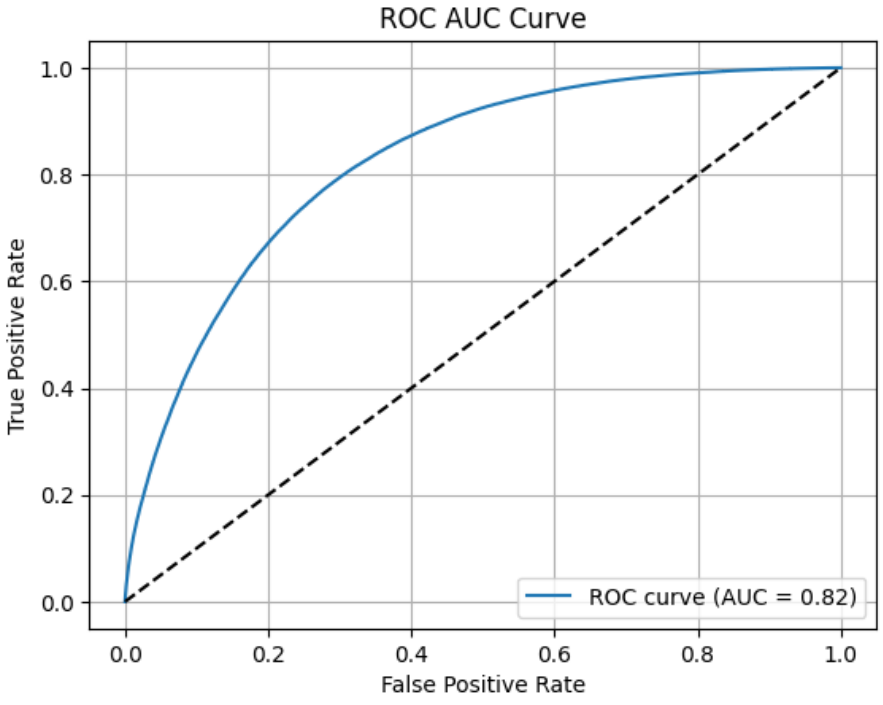

Figure 14: Multilayer Perceptron (MLP) ROC Curve