Predicting road accidents in smart cities: machine learning approach for enhanced safety

Anastasiya Doroshenko[*,†] and Dmytro Savchuk[†]

Lviv Polytechnic National University, Stepan Bandera 12 79013 Lviv, Ukraine

Abstract

The rapid development of smart cities opens up new opportunities for improving road safety using predictive technologies. This article focuses on predicting road accidents in smart cities using big data, artificial intelligence (AI), and machine learning models. The paper analyzes a dataset of 45 features and about 8 million incidents, including factors such as time of the event, coordinates, distance of the road incident, city, region, zip code, time zone, temperature, airport, wind, humidity, pressure, visibility, precipitation, weather conditions, amenity, bump, junction, crossing, railway, roundabout, station, stop, period of day, and others. Different machine learning models, including Random Forest, Extreme Gradient Boosting, Gradient Boosting, Logistic Regression, Extra Tree, Decision Tree, MLP Classifier, and others, were evaluated for their prediction accuracy. The most effective model was Gradient Boosting, which achieved 85% accuracy while offering better interpretability.

The study highlights the potential of AI and machine learning in traffic accident prediction, with Gradient Boosting offering the most effective solution due to its balance of accuracy and clarity. The research helps integrate predictive analytics into smart city infrastructure, improve road safety, and minimize the social and economic costs associated with road accidents. Future research should focus on incorporating real-time data streams from IoT-based systems and extending models that can be adapted to different cities, thereby improving the accuracy of predictions and extending the generalizability of results to the broader urban environment. This work contributes to developing safer and more efficient transportation systems as part of the evolving concept of smart cities.

Keywords

Road accident, data, dataset, classification, prediction, feature, Random Forest, SMV, Logistic Regression, Extra Trees, Gradient Boosting, MLP Classifier, model, importance.

1. Introduction

With the development of technology, smart cities are becoming a reality, providing new opportunities to improve road safety. One of the important tasks that can be solved with the help of intelligent systems is the prediction of road traffic accidents (RTAs). Using big data, artificial intelligence (AI), and analytical tools, we can create models that predict possible accidents, allowing preventive measures to be taken in advance.

Such solutions have the potential to significantly reduce the number of road accidents, minimize medical costs, and improve the overall efficiency of the urban transportation system. This thesis discusses modern approaches to traffic accident forecasting, including machine learning methods, data analytics, and factors that influence the occurrence of accidents.

The object of research is the intellectual transportation systems of smart cities, particularly their ability to analyze and predict road traffic accidents (RTAs). In today's context of growing urbanization and the increasing number of vehicles, road safety is becoming increasingly important, making it necessary to find new solutions to reduce the number of road accidents.

CIAW-2024: Computational Intelligence Application Workshop, October 10-12, 2024, Lviv, Ukraine

[*] Corresponding author.

[†] These authors contributed equally.

anaastasiia.v.doroshenko@lpnu.ua (A. Doroshenko); dmytro.y.savchuk@lpnu.ua (D. Savchuk)

0000-0002-7214-5108 (A. Doroshenko); 0009-0006-0937-6161 (D. Savchuk)

The subject of the study is methods and technologies for predicting road accidents in smart cities using big data, data mining, artificial intelligence (AI), machine learning, and other analytical approaches. The study of factors that affect the likelihood of accidents, as well as tools for their prediction, is a key aspect of this research.

The purpose of the research is to develop an effective intellectual model for predicting accidents in smart cities, which will reduce the number of accidents by preventing risks in advance. To achieve this goal, it is planned to apply modern methods of data analysis, use integrated traffic monitoring systems, and identify key factors affecting road safety.

A dataset consisting of 45 features and about 8 million incidents was used [1] to predict traffic accidents. These characteristics include a time of the event, coordinates, distance of the road incident, city, region, zip code, time zone, temperature, airport, wind, humidity, pressure, visibility, precipitation, weather conditions, amenity, bump, junction, crossing, railway, roundabout, station, stop, period of day, and others [7].

In this work, was solved several diverse tasks that covered all stages of working with the dataset. First, a detailed analysis of the dataset was conducted, including a review of its content, identification of the main quantitative and qualitative characteristics, and study of possible types of these characteristics. After that, statistical information about the data was collected, formatted it, and checked for zero values. In cases where the amount of missing data exceeded 40%, a mechanism for generating missing data was implemented [2].

Particular attention was paid to studying the number of different types of incidents depending on several factors, such as region of the country, city, year, month, day of the week, and weather conditions. The distribution of incidents by time of day, weather conditions, and duration of events was analyzed in detail. In addition, a correlation matrix was constructed to examine the relationships between the quantitative and qualitative characteristics of the dataset. Qualitative characteristics were converted into quantitative ones by encoding them.

The next step was to split the dataset into training and test samples in the ratio of 70% to 30%, respectively. Next, various machine learning models were researched and developed that could be used to predict the probability of traffic incidents. The models considered included Random Forest, Extreme Gradient Boosting, Gradient Boosting, Logistic Regression, Extra Tree, Decision Tree, MLP Classifier, and others [3].

2. Data preparation

2.1. Source dataset

The first step involves examining the original dataset to understand its structure, including the number of observations, the characteristics it contains, and the target variable for prediction [4].

In our case, the dataset includes a wide range of characteristics, such as time of incident, geographic coordinates, distance to the incident, city, region, zip code, time zone, temperature, airport proximity, wind, humidity, atmospheric pressure, visibility, precipitation, and various road and weather conditions. It also includes attributes such as intersections, roundabouts, stations, and time of day. The dataset consists of 45 characteristics and approximately 8 million recorded incidents, which are used to predict traffic accidents [6].

Now let’s take a closer look at the first 23 independent features of our dataset in more detail Table 1.

Table 1
Source Dataset

N	Feature name	Description	Type
X_1	Time	The local time of the accident.	Interval
X_2	Coordinate (Latitude, Longitude)	GPS coordinates of the accident location.	Interval

N	Feature name	Description	Type
X_3	Distance	The length of the road traffic accident.	Ratio
X_4	Code of the airport	Airport station which is the closest to the accident.	Nominal
X_5	Weather time	Local time-stamp of weather observation record.	Interval
X_6	Temperature	Temperature in Fahrenheit.	Interval
X_7	Wind Chill	Wind chill in Fahrenheit.	Interval
X_8	Humidity	Humidity percentage.	Interval
X_9	Pressure	Air pressure in inches.	Interval
X_{10}	Visibility	Visibility in miles.	Interval
X_{11}	Wind Direction	Wind direction.	Nominal
X_{12}	Wind Speed	wind speed in miles per hour.	Ratio
X_{13}	Precipitation	Precipitation amount in inches.	Ratio
X_{14}	Weather Condition	Weather condition, can be rain, snow, thunderstorm, fog, etc.	Nominal
X_{15}	Amenity sign	Availability of amenity sign in a nearby location.	Nominal
X_{16}	Bump sign	Availability of speed bump or hump signs in a nearby location.	Nominal
X_{17}	Crossing sign	Availability of crossing sign in a nearby location.	Nominal
X_{18}	Give Way sign	Availability of give way sign in a nearby location.	Nominal
X_{19}	Junction sign	Availability of junction sign in a nearby location.	Nominal
X_{20}	No Exit sign	Availability of no exit sign in a nearby location.	Nominal
X_{21}	Railway sign	Availability of railway sign in a nearby location.	Nominal
X_{22}	Station sign	Availability of station sign in a nearby location.	Nominal
X_{23}	Stop sign	Availability of stop sign in a nearby location.	Nominal
X_{nn}	…	…	…

As for the description of the target class labels, it is given below in Table 2.

Table 2
Target Class

N	Severity	
	Details	Value
Y_1	Least impact on traffic	1
Y_2	Small impact on traffic	2
Y_3	Moderate impact on traffic	3
Y_4	Significant impact on traffic	4

2.1. Missing values check

Now, we need to check for blank values of the dataset features, and for this purpose, we can build bar chart Figure 1.

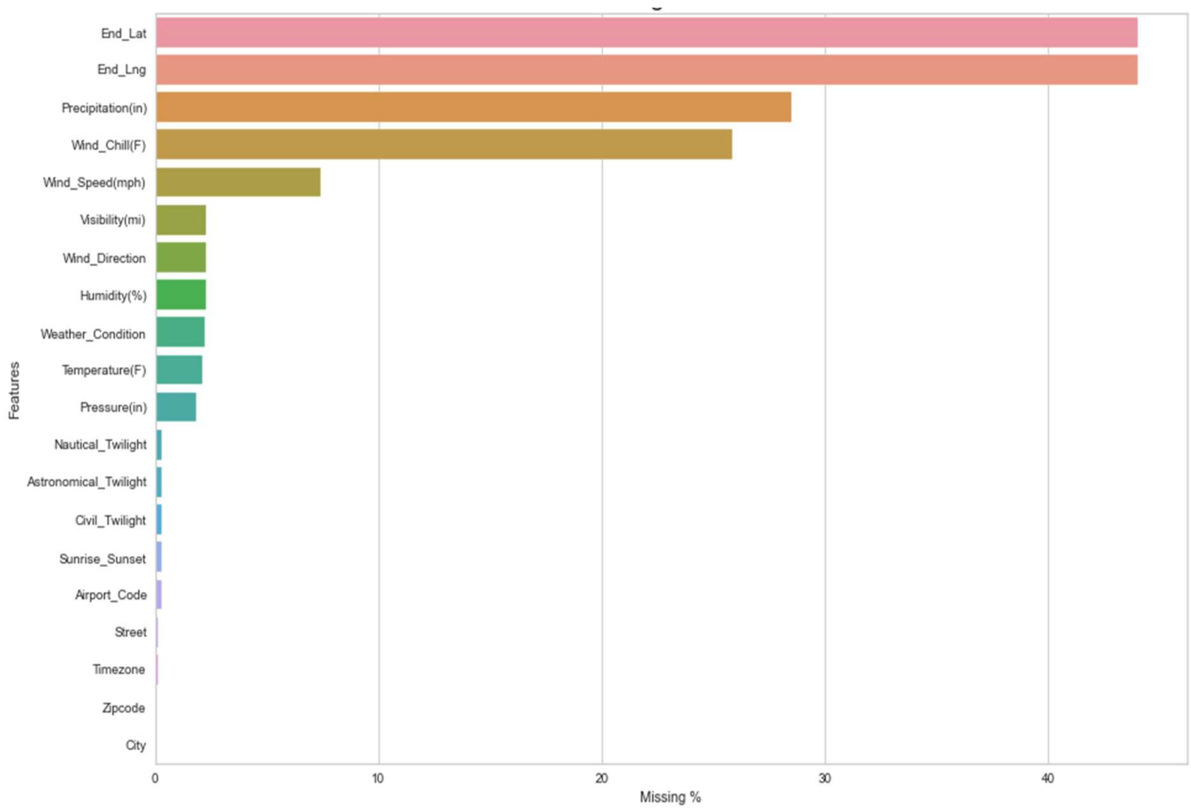


Figure 1: Percentage of missing values for the dataset.

As we can see from the graph, for the End Latitude and End Longitude features, the percentage of missed values is more than 40. Therefore, these missing values need to be filled with new ones using one of the techniques, in our case, filling using the mean value.

2.2. Investigation of the Target Class (Severity)

To investigate target class, it is better to draw a graph of the distribution of the number of traffic events by severity (Figure 2). The graph above shows the general distribution of incidents by severity. It can be seen that there are 4 types of severity levels in total:

1 - Least impact;
2 - Small impact;
3 - Moderate impact;
4 - Significant impact;

It can also be seen (Figure 5) that the total number of incidents by severity is as follows:

- Small impact equals 79.4% of all accidents in the dataset, which is 6129159 values.
- Moderate impact is 16.8%, which is 1295336 records.
- Significant impact belongs to 2.8% of accidents which is only 203120 of all.
- Only 67066 accidents have the least impact, which is almost 1%.

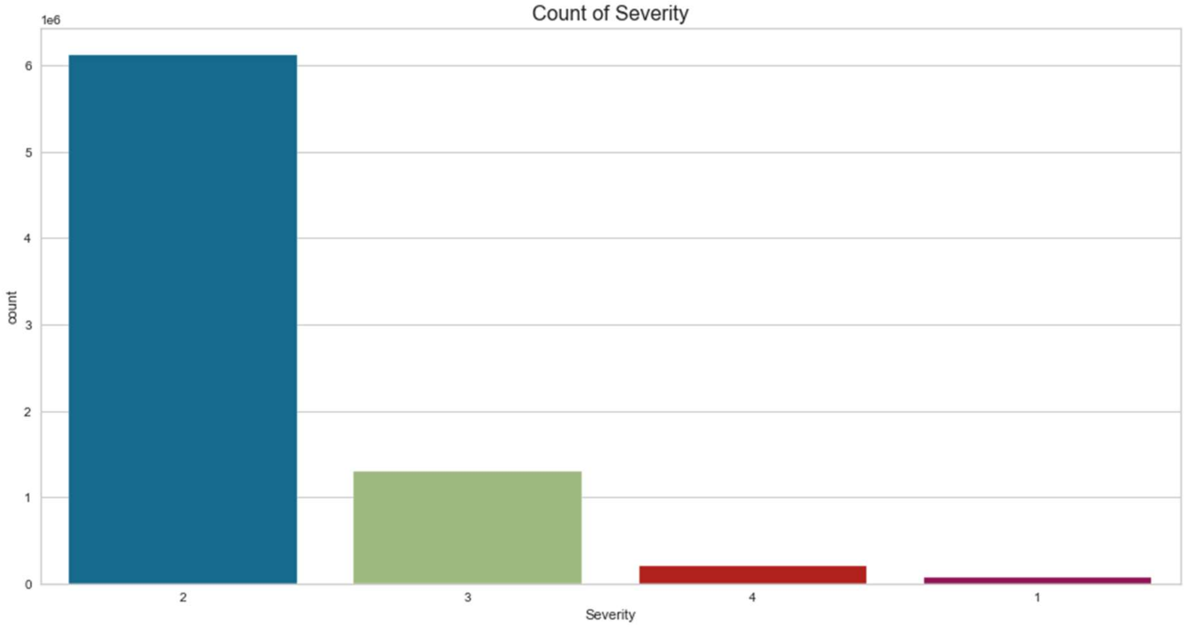


Figure 2: Accident Severity distribution.

2.3. Data Analysis

Let's build a country figure that will represent the top 10 states with the highest number of traffic incidents.

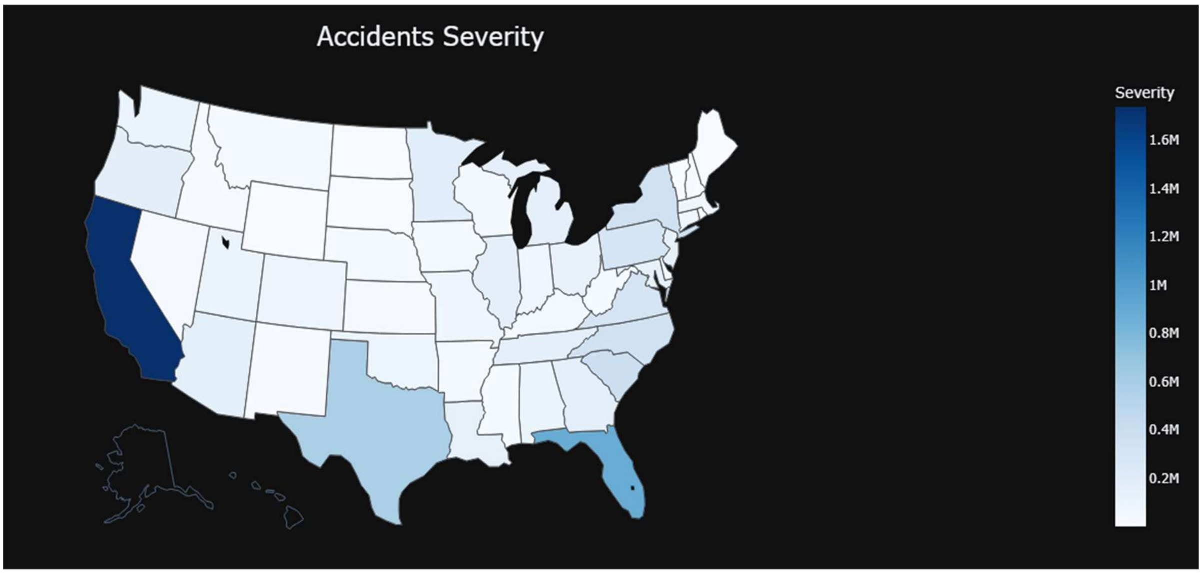


Figure 3: Severity distribution in the USA.

Figure 3 shows that the states with the highest number of incidents are shown in blue, and the states with the lowest number of incidents are shown in white. So, the top 3 states with the highest number of incidents are:

1) California;
2) Florida;
3) Texas.

The graph above (Figure 4) shows the top 10 states by incidents in the form of a bar chart. It shows that incidents occurred most frequently in the following states: California, Florida, Texas, New York City, South Carolina, New York, Oregon, Virginia, Pennsylvania, and Illinois.

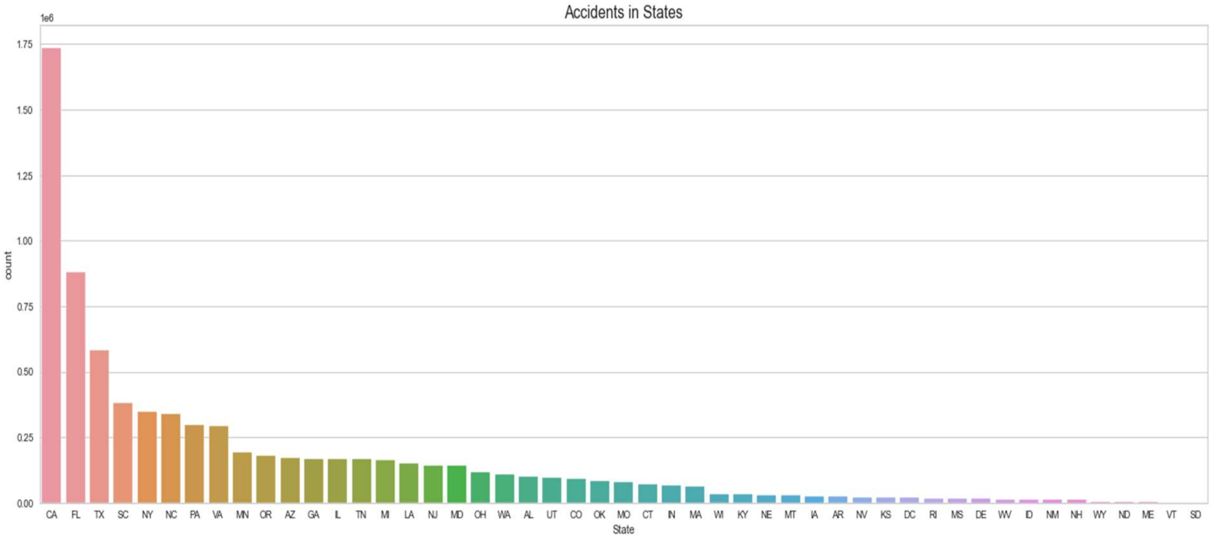


Figure 4: Severity distribution by State.

The next step is to build a graph of incidents in the states depending on their severity. To do this, let’s divide the total number of incidents in the state into four parts. Namely, incidents with the least (blue), small (green), moderate (red), and significant (purple) impact on traffic.

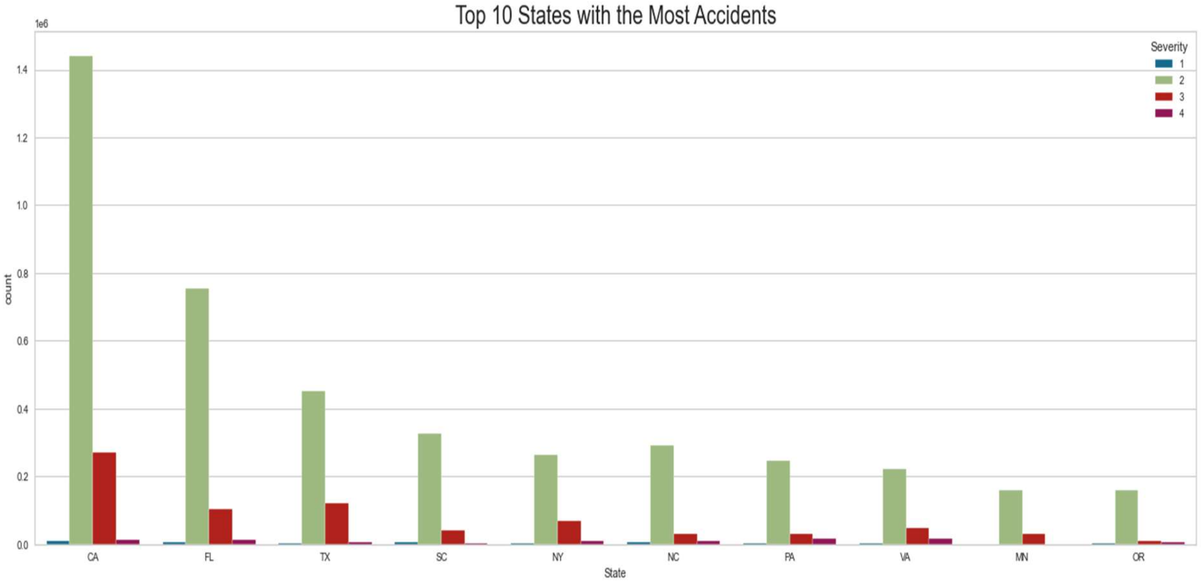


Figure 5: Top States by Severity.

As can be seen from the Figure 5 above, the distribution of incident severity across the states is uneven, with the number of small-impact incidents being much higher than the other types.

However, the distribution is the same for all ten states, with the highest number of:

- small-impact incidents, followed by
- moderate-impact,
- significant-impact, and

- least-impact.

Now let's look at the mean severity of incidents depending on weather conditions. To do this, we need to group the data by two characteristics: weather conditions and incident severity. And after that, let's draw Figure 6. This graph shows that:

- The worst severity of an incident, namely an incident with significant impact, is typical for a weather condition such as Light Blowing Snow.
- For moderate-impact incidents, the weather conditions are usually as follows: Patches of Fog / Windy, Light Fog, Partial Fog / Windy, Heavy Freezing Rain / Windy.
- The lightest severity of the incident is typical for such weather conditions as: Low Drifting Snow, and Heavy Rain Showers.

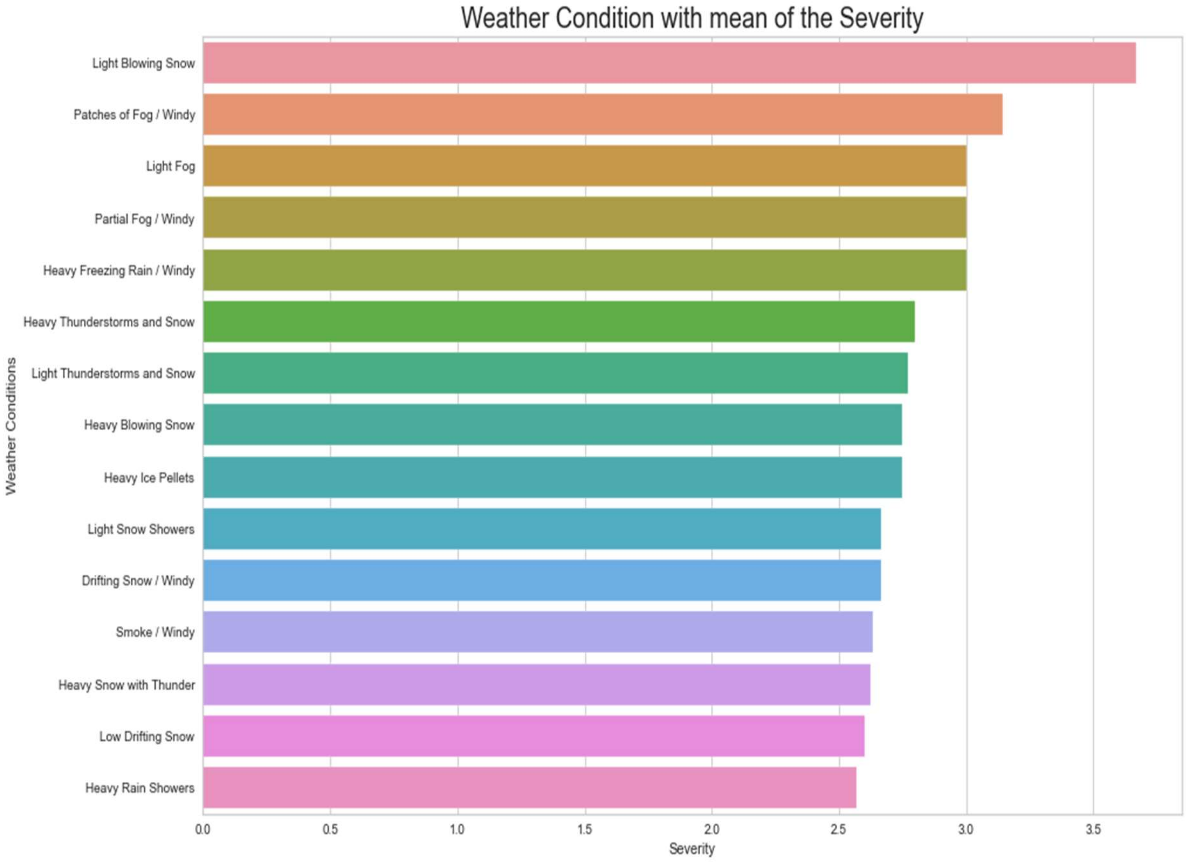


Figure 6: Mean severity distribution by Weather Condition.

The graphic above (Figure 7) shows the total duration of each accident depending on its severity. It shows that the more complex the incident, the longer it takes to resolve. For example, it takes only about 0.6 hours to resolve a least impact accident, and about 1 day to resolve a significant accident.

2.4. Correlation Matrix

In addition, it is important to create a correlation matrix to get a clearer understanding of how different characteristics are related to each other and influence each other. This type of chart allows you to observe the relationships between various variables. The resulting correlation matrixes that visually represent these relationships are shown in Figure 8, and Figure 9 below.

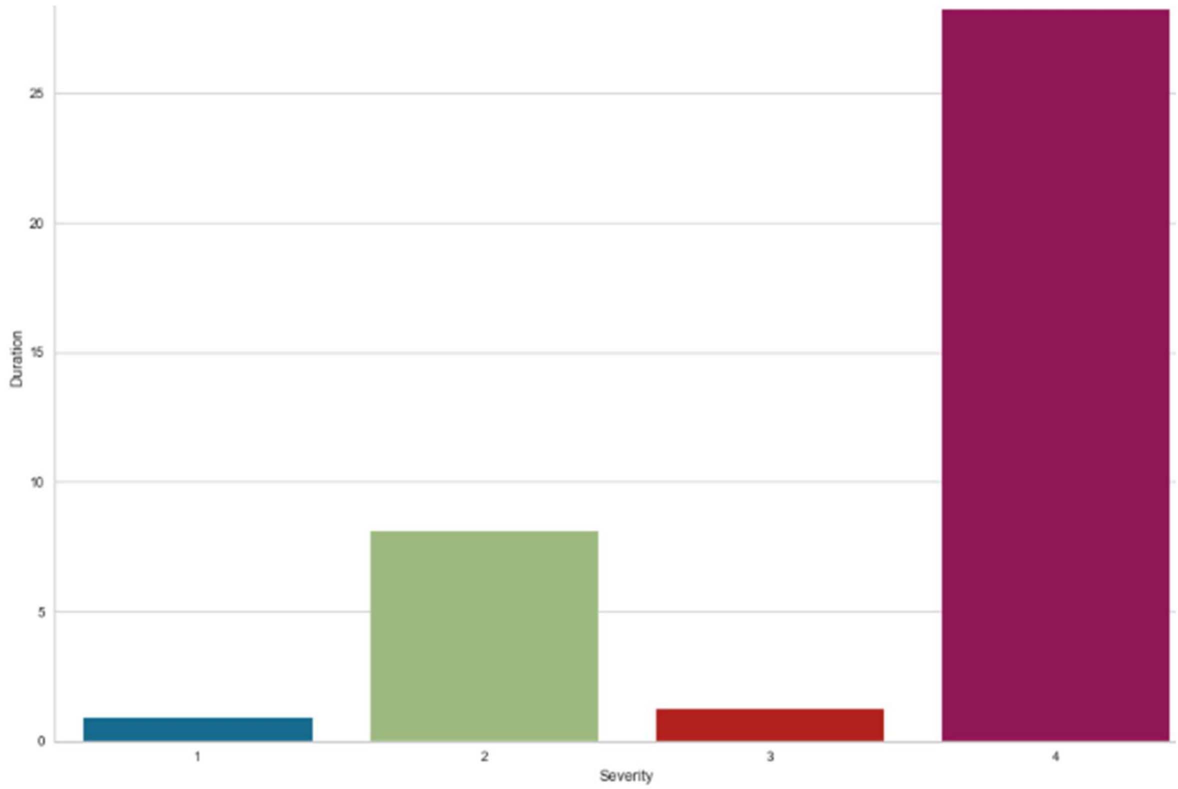
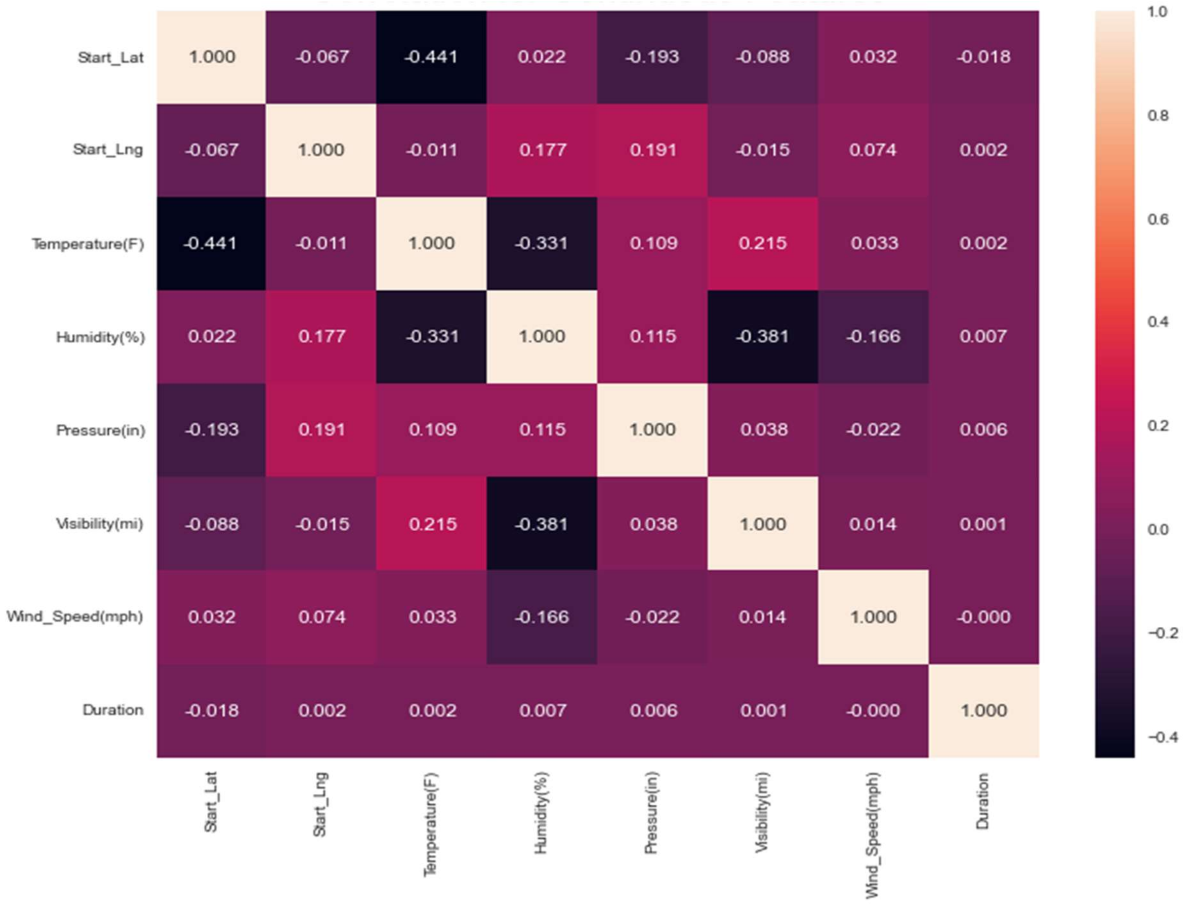


Figure 7: Mean duration of each accident by severity.



	Start_Lat	Start_Lng	Temperature(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	Duration
Start_Lat	1.000	-0.067	-0.441	0.022	-0.193	-0.088	0.032	-0.018
Start_Lng	-0.067	1.000	-0.011	0.177	0.191	-0.015	0.074	0.002
Temperature(F)	-0.441	-0.011	1.000	-0.331	0.109	0.215	0.033	0.002
Humidity(%)	0.022	0.177	-0.331	1.000	0.115	-0.381	-0.166	0.007
Pressure(in)	-0.193	0.191	0.109	0.115	1.000	0.038	-0.022	0.006
Visibility(mi)	-0.088	-0.015	0.215	-0.381	0.038	1.000	0.014	0.001
Wind_Speed(mph)	0.032	0.074	0.033	-0.166	-0.022	0.014	1.000	-0.000
Duration	-0.018	0.002	0.002	0.007	0.006	0.001	-0.000	1.000

Figure 8: Correlation matrix for continuous features.

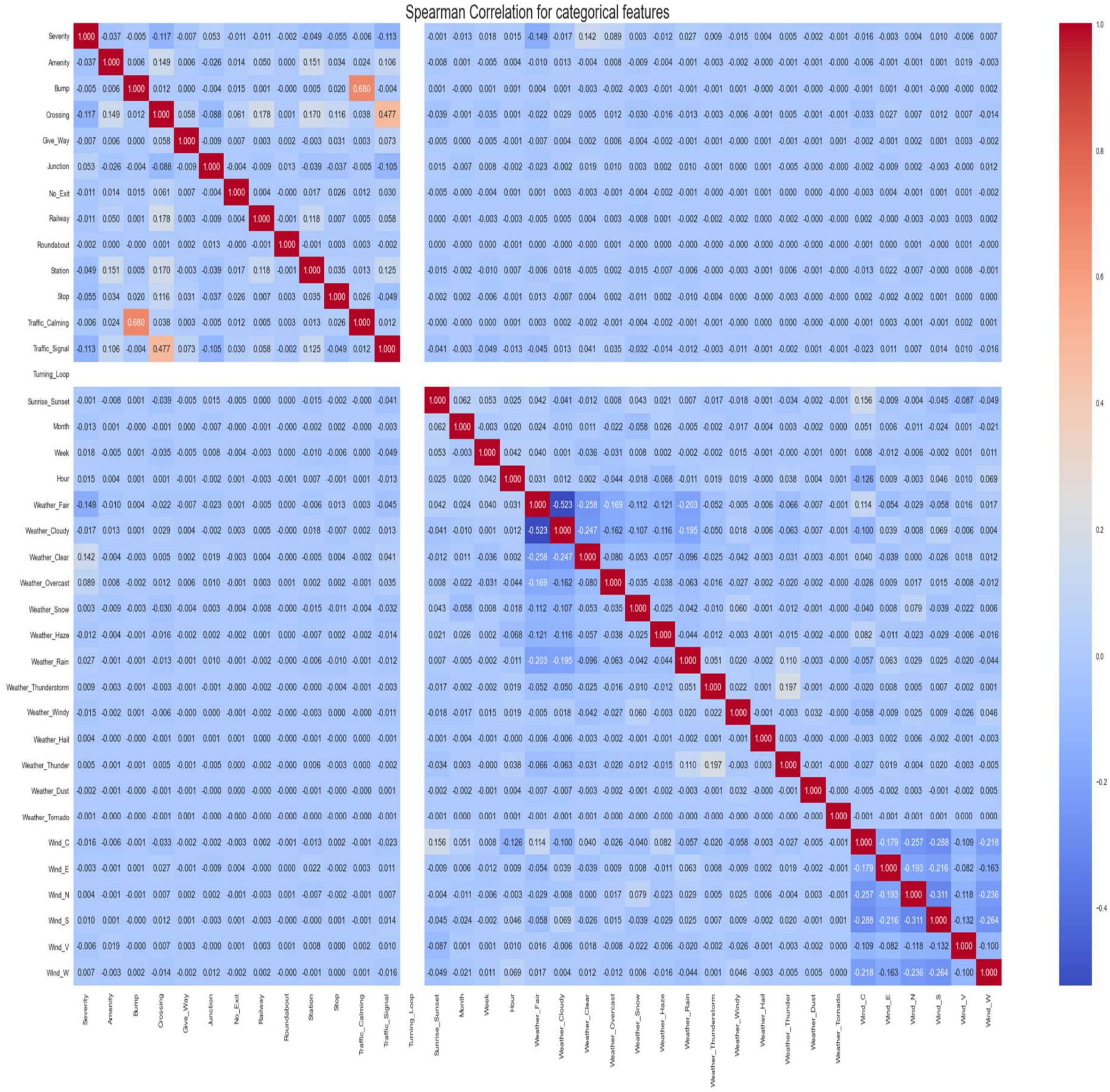


Figure 9: Correlation matrix for categorical features.

From the figure above, it can be seen that the weakest relationship is with the following characteristics:

- **Temperature (F)**, and **Start Latitude**;
- **Humidity (%)**, and **Temperature (F)**;
- **Visibility (mi)**, and **Humidity (%)**.

3. Classification

3.1. Classifiers Types

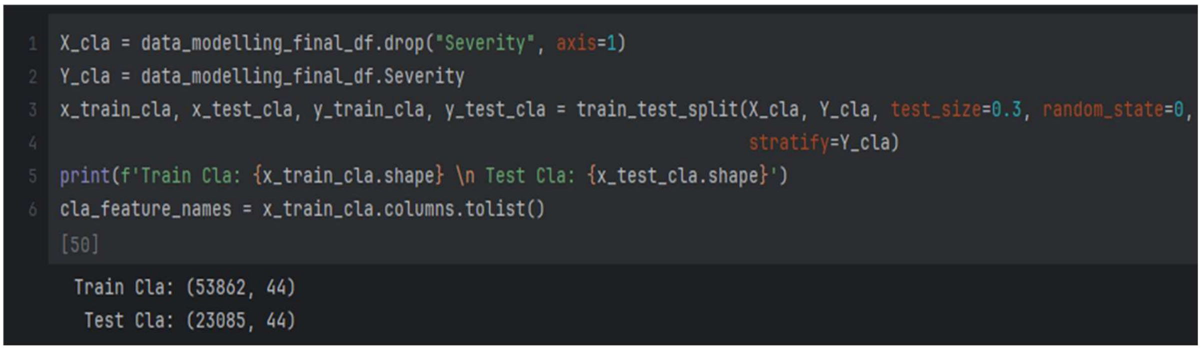
A variety of machine learning algorithms and models were used to predict the probability of road accidents. These algorithms were selected based on their unique capabilities and strengths in performing the classification task. These classifiers are as follows: extreme gradient boosting (xgboost), light gradient boosting machine (lightgbm), gradient boosting classifier (gbc), random forest (rf), extra trees (et), logistic regression (lr), ridge classifier, dummy classifier, adaboost (Ida), k - nearest neighbors (knn), decision tree (dt), and SVM with a linear kernel (Table 3).

Table 3
List of Classifiers

N	Abbreviation	Reference	Classifier
1	xgboost		Extreme Gradient Boosting
2	lightgbm	[7]	Light Gradient Boosting Machine
3	gbc		Gradient Boosting Classifier
4	rf	[9]	Random Forest Classifier
5	et	[10]	Extra Trees Classifier
6	ada	[11]	Ada Boost Classifier
7	Ir	[12]	Logistic Regression
8	ridge	[13]	Ridge Classifier
9	dummy	[14]	Dummy Classifier
10	Ida	[15]	Linear Discriminant Analysis
11	svm	[16]	SVM (Linear Kernel)
12	knn	[17]	K Neighbors Classifier
13	dt	[18]	Decision Tree Classifier
14	NB	[19]	Naive Bayes

3.2. Model Creation

Before starting the model building process, it is important to divide the dataset into two separate parts: one for training and one for testing. This separation ensures that the performance of the model can be properly evaluated. Given that the dataset contains a large number of records, it was decided to select only 1% of the total data to prevent the risk of overfitting. Once the dataset was reduced, the next step was to split the data for modeling into training and test sets, as shown in Figure 10.



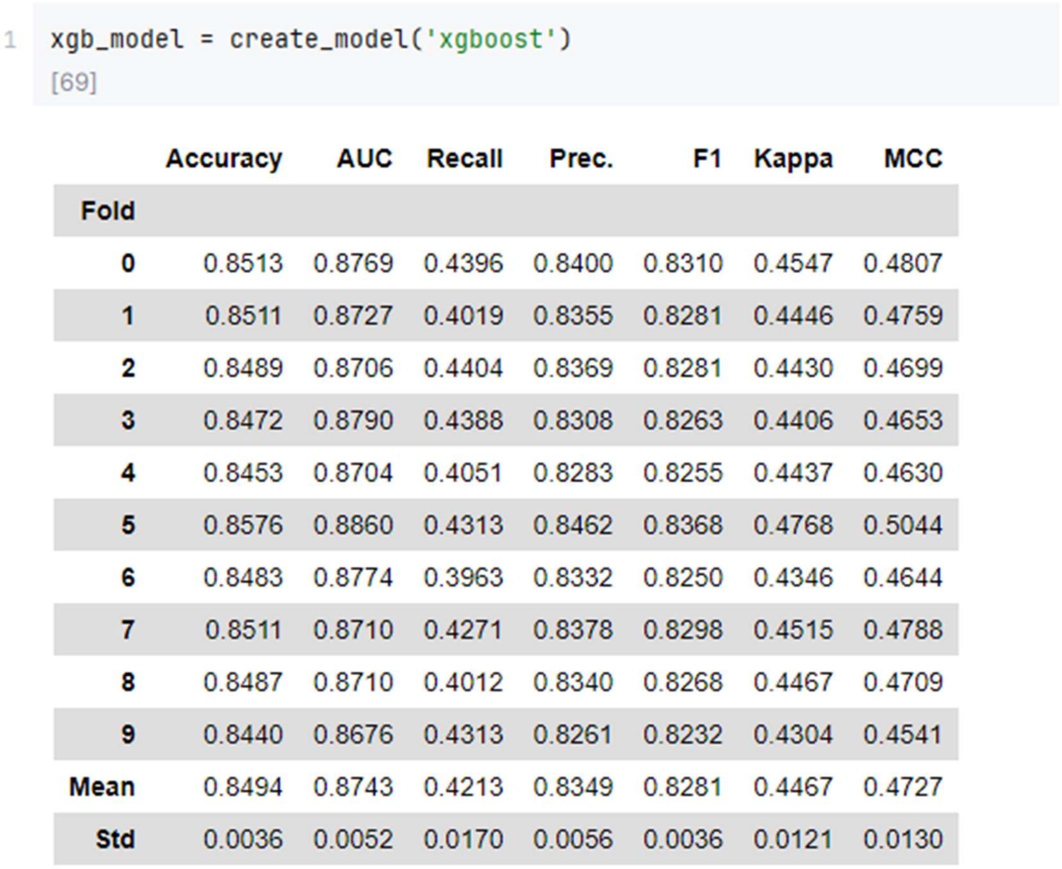
```
X_cla = data_modelling_final_df.drop("Severity", axis=1)
Y_cla = data_modelling_final_df.Severity
x_train_cla, x_test_cla, y_train_cla, y_test_cla = train_test_split(X_cla, Y_cla, test_size=0.3, random_state=0,
                                                                    stratify=Y_cla)
print(f'Train Cla: {x_train_cla.shape} \n Test Cla: {x_test_cla.shape}')
cla_feature_names = x_train_cla.columns.tolist()
[50]

 Train Cla: (53862, 44)
  Test Cla: (23085, 44)
```

Figure 10: Splitting a Dataset into parts.

Specifically, 70% of the data was allocated for training the model, allowing it to learn on a significant portion of the dataset, while the remaining 30% was reserved for testing. This split ensures that the model can be tested on data it has not seen before, allowing for a more accurate assessment of its predictive capabilities.

3.3. Standard Classifier

Figure 11 below shows a typical model building and training process using standard classification algorithms. The diagram illustrates the steps involved in building and training a model and emphasizes the iterative nature of the process, where the classifier was run ten times. After completing these ten runs, the average accuracy achieved by the model is approximately 85%, which is a good indicator of its performance.



```
xgb_model = create_model('xgboost')
[69]
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
Fold							
0	0.8513	0.8769	0.4396	0.8400	0.8310	0.4547	0.4807
1	0.8511	0.8727	0.4019	0.8355	0.8281	0.4446	0.4759
2	0.8489	0.8706	0.4404	0.8369	0.8281	0.4430	0.4699
3	0.8472	0.8790	0.4388	0.8308	0.8263	0.4406	0.4653
4	0.8453	0.8704	0.4051	0.8283	0.8255	0.4437	0.4630
5	0.8576	0.8860	0.4313	0.8462	0.8368	0.4768	0.5044
6	0.8483	0.8774	0.3963	0.8332	0.8250	0.4346	0.4644
7	0.8511	0.8710	0.4271	0.8378	0.8298	0.4515	0.4788
8	0.8487	0.8710	0.4012	0.8340	0.8268	0.4467	0.4709
9	0.8440	0.8676	0.4313	0.8261	0.8232	0.4304	0.4541
Mean	0.8494	0.8743	0.4213	0.8349	0.8281	0.4467	0.4727
Std	0.0036	0.0052	0.0170	0.0056	0.0036	0.0121	0.0130

Figure 11: Extreme Gradient Boosting Classifier.

The classification report generated as part of the evaluation includes several important metrics that provide a detailed understanding of the model's performance. These metrics are as follows:

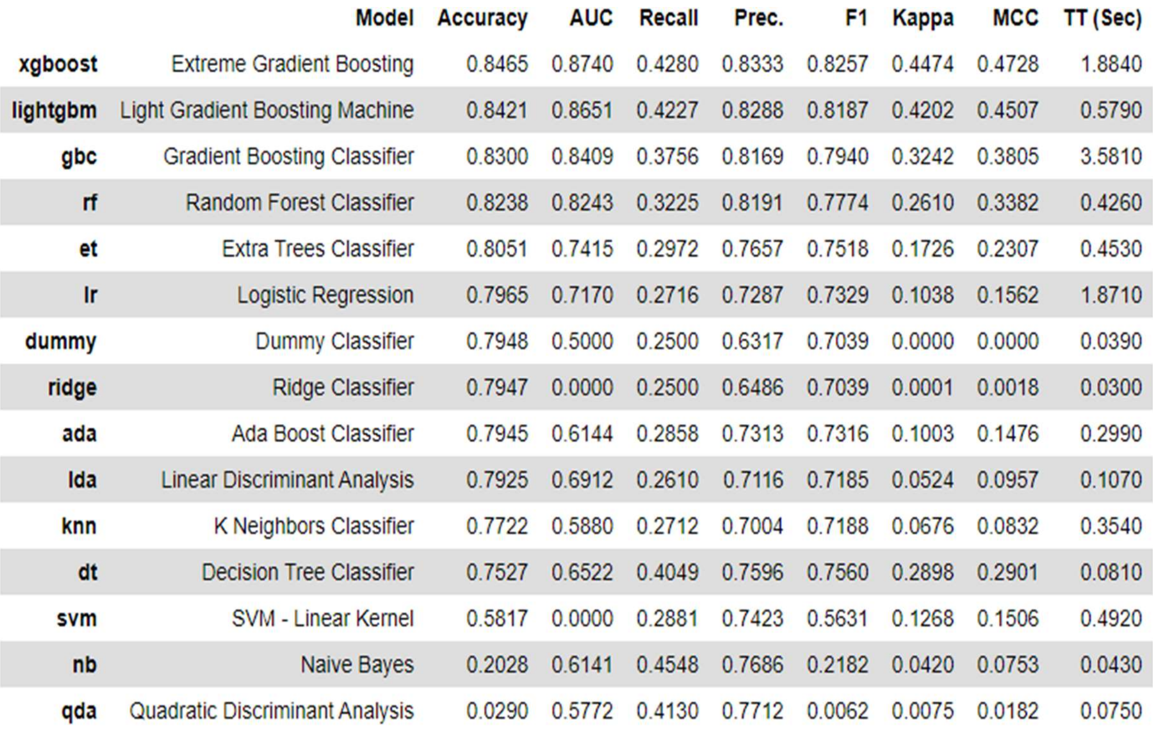
- **Fold** - refers to the breakdown of data during cross-validation.
- The main attribute of a classification report is **Accuracy**. It is measured as the percentage of correctly predicted cases out of the total number of prognoses.
- The next indicator is the receiver operating characteristic curve, in other words, **ROC curve**. It is measured by the area under the curve, which is estimated as the model's ability to recognize the existing classes.
- Sensitivity or **Recall** is a measure of the rate of actual positive cases that were correctly recognized.
- The relation of correctly predicted positive observations to the number of predicted positive observations is often referred to as **Precision.**
- The mean value of accuracy and recall, which provides a balance between these two indicators, is **F1 score**.
- **Cohen's kappa** – statistic that measures the consistency between annotators, taking into account the possibility of random agreement [21].
- **Matthew's correlation coefficient (MCC)** a balanced metric that considers true and false positive and negative responses, providing a comprehensive assessment of binary classifications [22].

Together, these metrics provide a comprehensive view of the model's performance in various aspects, providing a comprehensive evaluation.

4. Model Comparison and Result Analysis

In reference to Figure 12, it can be seen that the Extreme Gradient Boosting classification model performed best in this analysis, achieving an 85% accuracy rate. It is followed by Light Gradient Boosting with 84% accuracy and Gradient Boosting, which showed a good 83% of accuracy. Conversely, the model that performed the worst in this particular task was Support Vector Machine (SVM), which recorded a relatively low prediction performance of only 51%. In addition, Naïve Bayes performed even worse, achieving only 20% accuracy, and the Quadratic Discriminant Model performed terribly, showing only 2% accuracy.

Moreover, other classification quality metrics confirmed this assessment and produced consistent results similar to those illustrated by the ROC curve, Precision, Recall, F1, Cohen's kappa, and MCC. In the end, it is clear that the best model in this analysis is the Extreme Gradient Boosting classifier (xgboost), which provided a robust classification accuracy of 85%, as shown in Figure 11.



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
xgboost	Extreme Gradient Boosting	0.8465	0.8740	0.4280	0.8333	0.8257	0.4474	0.4728	1.8840
lightgbm	Light Gradient Boosting Machine	0.8421	0.8651	0.4227	0.8288	0.8187	0.4202	0.4507	0.5790
gbc	Gradient Boosting Classifier	0.8300	0.8409	0.3756	0.8169	0.7940	0.3242	0.3805	3.5810
rf	Random Forest Classifier	0.8238	0.8243	0.3225	0.8191	0.7774	0.2610	0.3382	0.4260
et	Extra Trees Classifier	0.8051	0.7415	0.2972	0.7657	0.7518	0.1726	0.2307	0.4530
lr	Logistic Regression	0.7965	0.7170	0.2716	0.7287	0.7329	0.1038	0.1562	1.8710
dummy	Dummy Classifier	0.7948	0.5000	0.2500	0.6317	0.7039	0.0000	0.0000	0.0390
ridge	Ridge Classifier	0.7947	0.0000	0.2500	0.6486	0.7039	0.0001	0.0018	0.0300
ada	Ada Boost Classifier	0.7945	0.6144	0.2858	0.7313	0.7316	0.1003	0.1476	0.2990
lda	Linear Discriminant Analysis	0.7925	0.6912	0.2610	0.7116	0.7185	0.0524	0.0957	0.1070
knn	K Neighbors Classifier	0.7722	0.5880	0.2712	0.7004	0.7188	0.0676	0.0832	0.3540
dt	Decision Tree Classifier	0.7527	0.6522	0.4049	0.7596	0.7560	0.2898	0.2901	0.0810
svm	SVM - Linear Kernel	0.5817	0.0000	0.2881	0.7423	0.5631	0.1268	0.1506	0.4920
nb	Naive Bayes	0.2028	0.6141	0.4548	0.7686	0.2182	0.0420	0.0753	0.0430
qda	Quadratic Discriminant Analysis	0.0290	0.5772	0.4130	0.7712	0.0062	0.0075	0.0182	0.0750

Figure 12: Classification result comparison.

4.1. Permutation Feature Importances

The last step of the research is to carefully study the importance of the features in the dataset. Here, we need to focus on the features and how they affect the performance of different classifiers and the overall severity of the accidents. To do this, we can plot the importance of the feature permutation. This graph is a good tool to understand the behavior of the model in machine learning.

The Feature permutation importance [23] gives a general idea of how the model makes decisions, namely, it allows to evaluate the contribution of individual features to the model's classification efficiency. This graph allows you to effectively group the importance of each feature and assess its impact on the model's classification efficiency. By studying these relationships, we can better understand which features are the most impactful and how they can be optimized to improve classification accuracy. This step is important to increase the reliability of the model and ensure that it accurately reflects the factors that trigger road accidents.

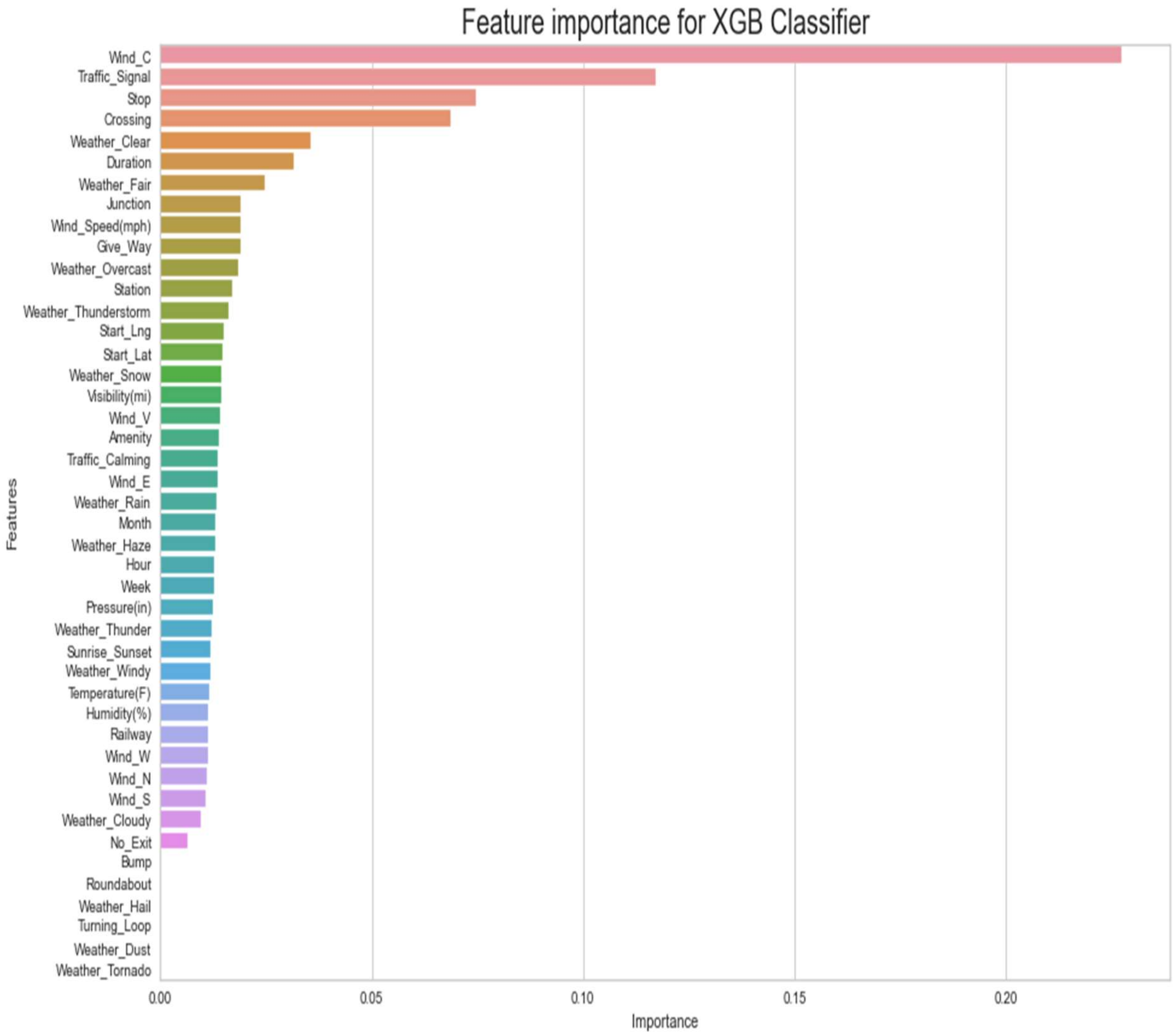


Figure 13: Feature Importance for Extreme Gradient Boosting.

The Figure 13 above shows a histogram that displays the importance of different features for an Extreme gradient boosting model (xgboost). The importance of each feature is represented by the length of the corresponding column, and the features are sorted in descending order of importance.

The most important features, according to this chart, are:

1. Wind Cloud (highest importance);
2. Traffic Signal;
3. Stop;
4. Crossing;
5. Weather Clear;
6. Duration;
7. Weather Fair;
8. Junction;
9. Wind Speed (mph);
10. Give Way.

On the other hand, features such as **“Weather Tornado”**, **“Weather Dust”**, and **“Weather Hail”** have very low importance because they have minimal impact on the model’s prediction.

5. Conclusions

This work demonstrated the potential for predicting traffic accidents in smart cities using big data models and machine learning, with Gradient Boosting proving to be the most effective approach due to its high accuracy (85%) and interpretability. By analyzing a dataset of 45 characteristics and approximately 8 million incidents, the study identified key factors that influence road accidents, such as time, place, and weather conditions, and emphasized the importance of data preprocessing to ensure reliable results.

The study showed that predictive models can significantly improve road safety in cities by enabling proactive measures such as adjusting traffic signals and warning of high-risk conditions. Although the study showed promising results, future work should focus on improving the models with real-time data and incorporating additional sources, such as IoT-based traffic monitoring systems, to improve the accuracy of prediction and generalization across different smart cities. Overall, the research contributes to the development of safer and more efficient urban transportation systems.

Acknowledgements

This work was realized within the framework of the Erasmus+ Jean Monnet Chair 2022 «101085612 – DataProEU – Data Protection in the EU».

References

[1] US Accidents (2016 - 2023). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents

[2] Doroshenko, Anastasiya. (2019). Application of global optimization methods to increase the accuracy of classification in the data mining tasks. Computer Modeling and Intelligent Systems, 2353, 98–109. https://doi.org/10.32782/cmis/2353-8

[3] Savchuk, D., Doroshenko, A. (2021). Investigation of Machine Learning Classification Methods Effectiveness. 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT). https://doi.org/10.1109/csit52700.2021.9648582

[4] A. Batyuk and V. Voityshyn, "Streaming Process Discovery Method for Semi-Structured Business Processes," 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, 2020, pp. 444-448, doi: 10.1109/DSMP47368.2020.9204201.

[5] Alshamrani, F. H., Syed, H. F., & Elhussein, M. A. (2019). Machine learning based model for traffic prediction in Smart Cities. 2nd Smart Cities Symposium (SCS 2019). https://doi.org/10.1049/cp.2019.0195

[6] Nagy, A. M., & Simon, V. (2018). Survey on traffic prediction in Smart Cities. Pervasive and Mobile Computing, 50, 148–163. https://doi.org/10.1016/j.pmcj.2018.07.004

[7] Obelovska, K., Snaichuk, Y., Selecky, J., Liskevych, R., Valkova, T. An Approach Toward Packet Routing in the OSPF-based Network with a Distrustful Router WSEAS Transactions on Information Science and Applications, 2023, 20, pp. 432–443.

[8] Gradient Boosting. WallStreetMojo. URL: https://www.wallstreetmojo.com/gradient-boosting/.

[9] Introduction to Random Forest in Machine Learning. Engineering Education (EngEd) Program | Section. URL: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/.

[10] How to Develop an Extra Trees Ensemble with Python - MachineLearningMastery.com. MachineLearningMastery.com. URL: https://machinelearningmastery.com/extra-trees-ensemble-with-python/.

[11] Yazhi Gao, W. Rong, Y. Shen and Z. Xiong, "Convolutional Neural Network based sentiment analysis using Adaboost combination," 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 2016, pp. 1333-1338, doi: 10.1109/IJCNN.2016.7727352.

[12] Thorn J. Logistic Regression Explained. Medium. URL: https://towardsdatascience.com/logistic-regression-explained-9ee73cede081.

[13] H. Luo and Y. Liu, "A prediction method based on improved ridge regression," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2017, pp. 596-599, doi: 10.1109/ICSESS.2017.8342986.

[14] Tezcan B. Why Using a Dummy Classifier is a Smart Move. Medium. URL: https://towardsdatascience.com/why-using-a-dummy-classifier-is-a-smart-move-4a55080e3549.

[15] J. Ghosh and S. B. Shuvo, "Improving Classification Model's Performance Using Linear Discriminant Analysis on Linear Data," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-5, doi: 10.1109/ICCCNT45670.2019.8944632.

[16] José Luis Rojo-Álvarez; Manel Martínez-Ramón; Jordi Muñoz-Marí; Gustau Camps-Valls, "Support Vector Machine and Kernel Classification Algorithms," in Digital Signal Processing with Kernel Methods, IEEE, 2018, pp.433-502, doi: 10.1002/9781118705810.ch10.

[17] Christopher A. K-Nearest Neighbor. Medium. URL: https://medium.com/swlh/k-nearest-neighbor-ca2593d7a3c4.

[18] What is a Decision Tree | IBM. IBM - Deutschland | IBM. URL: https://www.ibm.com/topics/decision-trees.

[19] Yıldırım S. Naive Bayes Classifier–Explained. Medium. URL: https://towardsdatascience.com/naive-bayes-classifier-explained-50f9723571ed (date of access: 02.09.2024).

[20] E. Pękalska and B. Haasdonk, "Kernel Discriminant Analysis for Positive Definite and Indefinite Kernels," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 6, pp. 1017-1032, June 2009, doi: 10.1109/TPAMI.2008.290.

[21] Performance Measures: Cohen's Kappa statistic. The Data Scientist. URL: https://thedatascientist.com/performance-measures-cohens-kappa-statistic/ (date of access: 02.09.2024).

[22] sklearn.metrics.matthews_corrcoef. scikit-learn. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.matthews_corrcoef.html (date of access: 02.09.2024).

[23] Feature Permutation Importance Explanations – ADS 1.0.0 documentation. Moved. URL: https://docs.oracle.com/en-us/iaas/tools/ads-sdk/latest/user_guide/mlx/permutation_importance.html (date of access: 02.02.2024).