Traffic Accident Severity Prediction Based on Data Cleaning and Machine Learning (Random Forest / Xgboost)

Wenjing Zhou *

California School of Management and Leadership, Alliant International University, San Diego, United States

* Corresponding Author Email: wzhou1@alliant.edu

Abstract. Traffic accidents have increasingly become a global concern, significantly affecting lives and economic sustainability. The US Accidents dataset from 2016 to 2023 provides an extensive record of accidents across the United States, containing detailed data and environmental data of these accident. This study aims to harness the potential of this rich database to predict severity level of accidents. Our research predominantly revolved around meticulous data cleaning, ensuring that the dataset's integrity was uncompromised. After preprocessing, the cleaned data was subjected to sophisticated Machine Learning techniques, primarily focusing on the Random Forest and XGBoost algorithms. These models were chosen due to their renowned capability in handling complex datasets and rendering accurate predictions, especially in scenarios laden with multiple variables. Upon application, the models demonstrated impressive efficacy. To validate the reliability and performance of our models, we employed the confusion matrix. This tool provided a clear visualization of the models' accuracy, revealing true positives, false negatives, and other crucial metrics. Furthermore, to enhance prediction outcomes, the Voting Classifier was implemented, combining the strengths of our primary models and consequently elevating the overall accuracy. The Random Forest algorithm exhibited substantial precision, while XGBoost further enhanced prediction accuracy. These findings underline the significant role of advanced data analytics and Machine Learning in comprehending traffic accident dynamics. In conclusion, our study emphasizes that leveraging state-of-the-art Machine Learning techniques on well-curated datasets can substantially improve our understanding and prediction of traffic accident severity. Such insights pave the way for the development of more effective preventive measures and safety protocols, aiming for a safer traffic environment in the future.

Keywords: US Accident; Random Forest; XGBoost.

1. Introduction

Traffic accidents are a global safety concern, affecting millions of lives each year. As per the World Health Organization, road traffic accidents lead to roughly 1.35 million fatalities each year, establishing it as the primary cause of death among individuals aged 5 to 29 years [1,2]. Several factors, including rapid urbanization, increased motorization, and sometimes lax enforcement of traffic laws, contribute to this grim statistic. High-income countries have seen a decrease in fatal accidents due to stringent laws, enhanced vehicle standards, and better road infrastructure. In contrast, low and middle-income countries continue to bear the brunt, accounting for over 90% of road traffic deaths. The socio-economic cost associated with these accidents is immense, with many families losing their primary breadwinners, which further perpetuates the cycle of poverty.

The United States, with its vast road network and high vehicle ownership, has a significant number of traffic accidents each year. The US Accidents dataset from 2016 to 2023 provides a comprehensive snapshot of the traffic accident landscape in the country. It covers various aspects such as location, time, weather conditions, and severity, among others. This rich dataset not only allows for a granular analysis of the factors contributing to accidents but also helps in benchmarking safety measures across states and identifying high-risk zones. The data spans over seven years, making it one of the most extensive accident datasets available, capturing both short-term anomalies and long-term trends.

The principal objective of this machine learning study [3,4] is to achieve a more profound comprehension of traffic accidents in the United States and discern patterns that can guide the implementation of preventive measures. The objectives include:

- Analyzing the dataset to identify high-risk zones, times, and conditions leading to accidents.
- Employing machine learning algorithms to make the prediction of accident severity level based on various features and gaining insights into the factors contributing most significantly to high-severity accidents.
- Providing recommendations based on the analysis to help policymakers, urban planners, and traffic authorities in implementing effective safety measures.

2. Methods

2.1. Dataset Description

From February 2016 to March 2023, this comprehensive traffic accident database, available on Kaggle.com, covers all 49 states of the United States [5].

The dataset we use, titled 'US_Accidents_March23', comprises 46 columns and 7728394 rows. Each column in this dataset describes various condition when the accident happens, such as weather, road conditions and the time of occurrence. Table 1 showcases a part of the first five and the last five entries form the 'US_Accidents_March23' dataset.

Table 1. Part of first and last five entries

ID	Source	Severity	Start_Time	Distance (mi)	Temperature (F)	Roundabout	Station
A-1	Source2	3	2/8/16 5:46	0.01	36.9	FALSE	FALSE
A-2	Source2	2	2/8/16 6:07	0.01	37.9	FALSE	FALSE
A-3	Source2	2	2/8/16 6:49	0.01	36	FALSE	FALSE
A-4	Source2	3	2/8/16 7:23	0.01	35.1	FALSE	FALSE
A-5	Source2	2	2/8/16 7:39	0.01	36	FALSE	FALSE
...							
A-7777757	Source1	2	8/23/19 18:03	0.543	86	FALSE	FALSE
A-7777758	Source1	2	8/23/19 19:11	0.338	70	FALSE	FALSE
A-7777759	Source1	2	8/23/19 19:00	0.561	73	FALSE	FALSE
A-7777760	Source1	2	8/23/19 19:00	0.772	71	FALSE	FALSE
A-7777761	Source1	2	8/23/19 18:52	0.537	79	FALSE	FALSE

2.1.1 Target Variable

Severity' serves as the target variable for this study. Figure 1 presents a bar chart of 'Severity'.

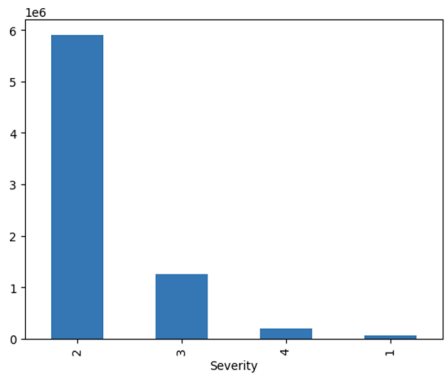

Fig. 1 The bar chart of 'Severity'

Referring to the bar chart presented, it depicts the distribution of the 'severity' feature, which is our target variable for prediction. The higher the value of 'Severity', the more severe the accident. As illustrated in the graph, the 'Severity' column consists of four distinct values: 1, 2, 3, and 4. The data

distribution for this column is imbalanced, with the number of incidents labeled as 2 considerably outnumbering the other three categories.

2.1.2 Features

Theoretically, all values excluding 'Severity' can be considered as features. However, after subsequent data processing, the final features used in our study are as shown in Table 2.

Table 2. Table of features used in final.

#	Column	Data Type
0	Severity	int64
1	Start_Time	object
2	End_Time	object
3	Start_Lat	float64
4	Start_Lng	float64
5	End_Lat	float64
6	End_Lng	float64
7	Distance(mi)	float64
8	Description	float64
9	Street	object
10	City	object
11	County	object
12	State	object
13	Zipcode	object
14	Timezone	object
15	Airport_Code	object
16	Weather_Timestamp	object
17	Temperature(F)	float64
18	Wind_Chill(F)	float64
19	Humidity (%)	float64
20	Pressure(in)	float64
21	Visibility(mi)	float64
22	Wind_Direction	object
23	Wind_Speed(mph)	float64
24	Precipitation(in)	float64
25	Weather_Condition	float64
26	Amenity	bool
27	Crossing	bool
28	Give_Way	bool
29	Junction	bool
30	Railway	bool
31	Station	bool
32	Stop	bool
33	Traffic_Signal	bool
34	Sunrise_Sunset	object
35	Civil_Twilight	object
36	Nautical_Twilight	object
37	Astronomical_Twilight	object

2.2. Data Cleaning and Preprocessing

2.2.1 Feature Selection

Table 3 displays all the features of this dataset along with their data types.

Table 3. All original features

#	Column	Data Type
0	ID	object
1	Source	object
2	Severity	int64
3	Start_Time	object
4	End_Time	object
5	Start_Lat	float64
6	Start_Lng	float64
7	End_Lat	float64
8	End_Lng	float64
9	Distance(mi)	float64
10	Description	object
11	Street	object
12	City	object
13	County	object
14	State	object
15	Zipcode	object
16	Country	object
17	Timezone	object
18	Airport_Code	object
19	Weather_Timestamp	object
20	Temperature(F)	float64
21	Wind_Chill(F)	float64
22	Humidity (%)	float64
23	Pressure(in)	float64
24	Visibility(mi)	float64
25	Wind_Direction	object
26	Wind_Speed(mph)	float64
27	Precipitation(in)	float64
28	Weather_Condition	object
29	Amenity	bool
30	Bump	bool
31	Crossing	bool
32	Give_Way	bool
33	Junction	bool
34	No_Exit	bool
35	Railway	bool
36	Roundabout	bool
37	Station	bool
38	Stop	bool
39	Traffic_Calming	bool
40	Traffic_Signal	bool
41	Turning_Loop	bool
42	Sunrise_Sunset	object
43	Civil_Twilight	object
44	Nautical_Twilight	object
45	Astronomical_Twilight	object

Columns like 'ID' and 'Source', which are primarily for sorting and categorizing, can be directly excluded. For the 'Country' column, since all the entries are 'US', this column doesn't provide any additional information or variance that could be useful for our analysis. Consequently, we have decided to drop this column from our dataset.

The subsequent step was to inspect the dataset for missing values. The results displaying the count of missing values are presented in Table 4.

Table 4. Missing values for each column

Column	Null Number
ID	0
Source	0
Severity	0
Start_Time	0
End_Time	0
Start_Lat	0
Start_Lng	0
End_Lat	3402762
End_Lng	3402762
Distance(mi)	0
Description	5
Street	10869
City	253
County	0
State	0
Zipcode	1915
Country	0
Timezone	7808
Airport_Code	22635
Weather_Timestamp	120228
Temperature(F)	163853
Wind_Chill(F)	1999019
Humidity(%)	174144
Pressure(in)	140679
Visibility(mi)	177098
Wind_Direction	175206
Wind_Speed(mph)	571233
Precipitation(in)	2203586
Weather_Condition	173459
Amenity	0
Bump	0
Crossing	0
Give_Way	0
Junction	0
No_Exit	0
Railway	0
Roundabout	0
Station	0
Stop	0
Traffic_Calming	0
Traffic_Signal	0
Turning_Loop	0
Sunrise_Sunset	23246
Civil_Twilight	23246
Nautical_Twilight	23246
Astronomical_Twilight	23246

As shown in Table 4, the columns 'End_Lat' and 'End_Lng', having excessive missing values and being temporal data, should be removed, as imputation is impractical due to their high degree of absence. Details on how to handle the missing values for the columns that do not need deletion will be elaborated in the following section.

For the rest of the columns, based on their data types, they can be categorized into three groups: boolean, float, and object types, each of which will be processed accordingly.

Since there are no missing values in the boolean columns of this dataset, we directly employed pie charts in Figure 2 to analyze and observe these data.

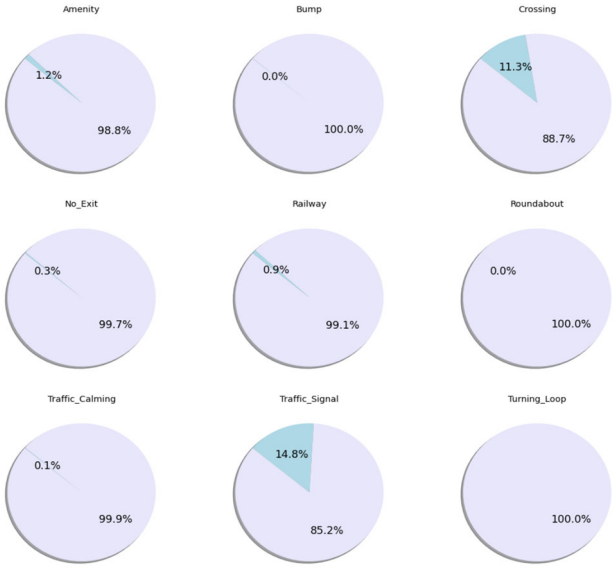

Fig. 2 Pie-chart for boolean data

As evident from the Figure 2, columns 'Bump', 'Roundabout', and 'Turning_Loop' have 100% 'False' values. Additionally, the 'No_Exit' and 'Traffic_Calming' columns also have 'False' values nearing 100%. Therefore, they offer no significant predictive value in the context of the vast dataset and would only increase computational costs unnecessarily. As a result, we decided to exclude these columns from the final set of features."

2.2.2 Handling Missing Values

For the numerical data within the dataset, we initially visualized them using histograms in Figure 3 to get an intuitive understanding of their distribution.

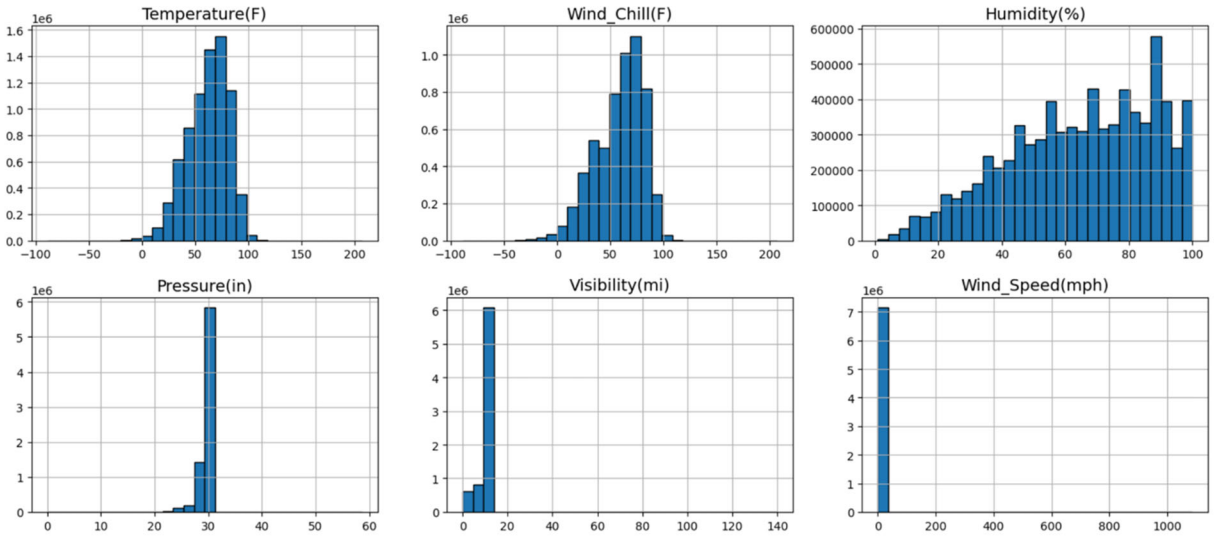

Fig. 3 Histograms for numerical data

Based on the insights from the histograms, we adopted the quartile imputation method for handling missing values, ensuring that the original characteristics of the data are retained to the greatest extent.

For the time-related numerical data in the database, such as 'Start-Time', 'End_Time', and 'Weather_Timestamp', a forward-fill strategy was adopted to address the missing values.

For the text-based columns in the dataset, such as 'Description', 'Street', 'City', and 'Zipcode', the most appropriate way to handle missing values is to assign them as 'unknown'.

However, there are four columns that need a slightly different treatment: 'Sunrise_Sunset', 'Civil_Twilight', 'Nautical_Twilight', and 'Astronomical_Twilight'. As illustrated in pie charts in Figure 4, these columns contain only two distinct values: 'Day' and 'Night'. Hence, they can be approximated as Boolean distributions. After converting them to a 0-1 encoding, we can conveniently impute missing values using the mode.

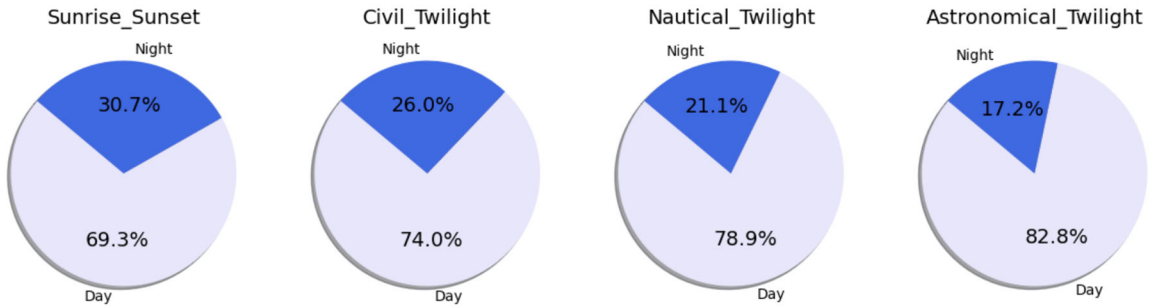

Fig. 4 Four pie charts for boolean-like columns

2.2.3 Encoding

In order to feed certain categorical data into machine learning algorithms, I applied one-hot encoding to the 'Timezone' and 'Wind_Direction' columns. Notably, the 'Wind_Direction' column presented some peculiarities.

As shown in unique values for the 'Wind_Direction' column, there's a significant variation in how wind directions are represented. Expressions of wind direction in this data varied in case, with no strict convention observed. For instance, 'W' and 'West' have the same meaning, as do 'CALM' and 'Calm'.

For such data, I first converted it all to uppercase. Then, I categorized it into seven major categories: "CALM", "NORTH", "EAST", "SOUTH", "WEST", "NE", "NW", "SE" and "VARIABLE". The category method is shown in Table 5.

Table 5. Category method

Category	Original Value
CLAM	CALM
NORTH	N, NNE, NNW
EAST	E, ENE, ESE
SOUTH	S, SSE, SSW
WEST	W,WSW, WNW
NE	NE
NW	NW
SE	SE
SW	SW
VARIABLE	VAR, VARIABLE

The bar chart after categorization is as shown in Figure 5.

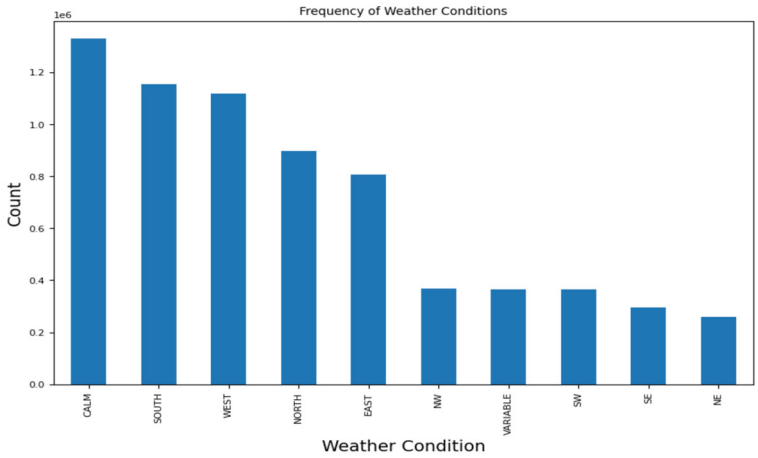

Fig. 5 The bar chart for 'Weather_Condition' after categorization

Subsequently, one-hot encoding can be carried out as usual on 'Wind_Direction' column.

For the remaining boolean columns like 'Sunrise_Sunset', 'Civil_Twilight', 'Nautical_Twilight', and 'Astronomical_Twilight', we implemented a 0-1 encoding process to transform them from boolean to integer types. This conversion facilitates easier computations in the later modeling phase.

For columns such as 'Street', 'City', 'State', and 'Zipcode', which describe locations and have high repetitiveness, frequency encoding is applied directly.

For irregular time-related data columns like 'Weather_Timestamp' and 'Start_Time', we employed the ISO8601 format to standardize them into a consistent datetime format. Subsequently, these datetime values were converted into floating-point numbers.

For the column 'Description', which describes the detailed circumstances of the accident in natural language, a natural language processing model was used for analysis. Here are some examples from the 'Description' column as shown in Table 6.

Table 1. Examples from the 'Description' column

#	Description
111	Accident on OH-48 Main St at Melford Ave.
112	Accident on Dixie Dr at Bartley Rd.
113	Accident on Brown St at Oakwood Ave.
114	Accident on Smithville Rd at Speice Ave.
115	Accident on Auburn Ave at Salem Ave.
...	
650000	HOV lane blocked due to crash on I-20 Westboun...
650001	One lane blocked and left-hand shoulder blocke...
650002	Lane blocked due to crash on US-53 Northbound ...
650003	Slow lane blocked due to crash on I-710 Eastbo...
650004	Right hand shoulder blocked due to crash on I-...

In this study, Sentiment Analysis was employed to analyze the text of this column, converting it into floating-point results.

2.3. Machine Learning Techniques

In the machine learning segment, I opted for two models to carry out predictions: Random Forest [6] and XGBoost. For both algorithms, we used a dataset split ratio of 2:8. Specifically, 80% of the data was utilized as the training set for model training, while the remaining 20% was reserved as the test set to assess the model's performance.

2.3.1 Random Forest

Random Forest is founded on the concept that a collective of weak learners can combine to create a strong learner. Random Forests are an ensemble of basic decision tree models. In Random Forests, multiple tree predictors are combined in such a way that each tree's decision depends on the values of a random vector, which is sampled independently and follows the same distribution for all trees within the forest [7]. The foundation of Random Forest lies in the concept that 'a collection of individual weak learners can collectively create a robust learner.' Each individual tree might not be the best model for the data, but when many trees vote on the output, they collectively produce a more accurate prediction. Each individual tree might not be the best model for the data, but when many trees vote on the output, they collectively produce a more accurate prediction. Along with bootstrapped samples, Random Forest also selects a random subset of features for each tree's split decision. This ensures that the trees are diverse and not overly reliant on any single feature. Due to its randomness, Random Forest is less likely to overfit compared to a single decision tree. It can generalize well to unseen data. Random Forest has built-in methods to handle missing values. Since each tree is independent of others, the model can be parallelized, making it faster for large datasets.

2.3.2 XGBoost Algorithms

XGBoost, abbreviated as "Extreme Gradient Boosting" [8], is a widely recognized open-source machine learning library celebrated for its efficient and scalable implementation of gradient boosting. It finds its application in supervised learning scenarios, where we leverage training data (comprising multiple features denoted as x) to make predictions about a target variable, y [9]. Fundamentally, XGBoost is a realization of the gradient boosting algorithm, which is an ensemble technique that sequentially introduces new models to rectify errors made by prior models [10]. What sets XGBoost apart from other gradient boosting algorithms is its built-in L1 (Lasso Regression) and L2 (Ridge Regression) regularization, effectively guarding against overfitting. XGBoost has been designed to be highly efficient. It can utilize the power of parallel processing (on a single machine) and can also be run on distributed systems, making it feasible to train on very large datasets. XGBoost incorporates a built-in mechanism to manage missing values. During the process of selecting splits, it intelligently determines whether missing values should be assigned to the left or right child node, guided by the potential gain. This sets it apart from other Gradient Boosting Machine (GBM) algorithms that rely on greedy methods and halt node splitting upon encountering a negative loss. XGBoost grows the tree to its maximum depth and then prunes backward. The library features a highly efficient cross-validation implementation at every stage of the boosting process, facilitating the straightforward selection of the optimal number of boosting rounds for achieving peak performance. By controlling the depth of the decision trees in the boosting process, XGBoost can be made more robust to overfitting.

2.4. Model Validation – Classification Matrix

For classification tasks, assessing a model's performance requires the use of appropriate metrics. Various metrics can offer insights into different facets of the model's performance. In this research, author employed two classification matrix ways called confusion matrix and classification report [11].

2.4.1 Confusion Matrix

A confusion matrix is a valuable tool in classification, primarily employed in supervised learning to evaluate the performance of an algorithm. Its utility extends beyond just identifying errors made by a classifier; it also reveals the nature of these errors.

In multi-class classification scenarios, the confusion matrix scales accordingly. In such cases, the matrix assumes a size of $n \times n$, where n corresponds to the number of classes under consideration. Each row in the matrix corresponds to the instances predicted for a particular class, while each column corresponds to the instances belonging to the actual class.

Visualizing a confusion matrix can help researchers quickly identify if the model is making specific types of errors more than others or if it struggles with certain classes in multi-class problems.

2.4.2 Classification Report

The classification report is a valuable tool available in machine learning libraries like scikit-learn, designed to aid in the evaluation of prediction quality produced by a classification algorithm. It provides crucial metrics such as precision, recall, and the F1-score for each individual class. This detailed breakdown of metrics is especially valuable when dealing with multi-class classification problems.

classification_report is a convenient function provided by the sklearn.metrics module in the scikit-learn library for Python [12]. Given true labels and predicted labels, classification_report generates a detailed report showing the aforementioned metrics for each class and averages of these metrics.

2.5. Enhancement using Voting Classifier

2.5.1 Principles and Implementation

The Voting Classifier is a model ensemble technique that combines multiple different models into a single model, which is then used to make predictions. The main premise behind the Voting Classifier is that by combining multiple models, the ensemble can often produce better results than any individual model on its own. This can help reduce overfitting and often yields a more robust and accurate model.

In a voting classifier, there are two primary voting methods: hard voting and soft voting. Hard voting involves each base model in the ensemble casting a "vote" for a class, and the class with the most votes becomes the ensemble's prediction. In contrast, soft voting has each model in the ensemble predict class probabilities instead of assigning class labels.

2.5.2 Advantages

Voting classifiers can help in reducing overfitting and combine the strengths of diverse models. They often perform better than individual models, especially if the base models make independent errors.

2.5.3 Models Incorporated

In this case, author chose soft voting to combine Random Forest model and XGBoost model. Soft voting takes the average of the predicted class probabilities from each model for each class. The highest average probability should be decided to be the final class. By considering the confidence of each model in its prediction, soft voting can achieve better performance, especially when the models are well-calibrated.

3. Results

3.1. Model Performance Metrics

3.1.1 Accuracy of Random Forest

The accuracy of Random Forest is 0.921. The classification report for the random forest is as shown in Table 7.

Table 2. Classification report for random forest model

	Precision	Recall	F1-score	Support
1	0.87	0.6	0.71	12863
2	0.94	0.97	0.95	1181247
3	0.87	0.81	0.84	252707
4	0.51	0.28	0.36	38811
Accuracy			0.92	1485628
Macro Avg	0.8	0.67	0.72	1485628
Weighted Avg	0.92	0.92	0.92	1485628

Figure 6 is the ranking of feature importance derived from the Random Forest model. The factor with the strongest correlation is 'street'.

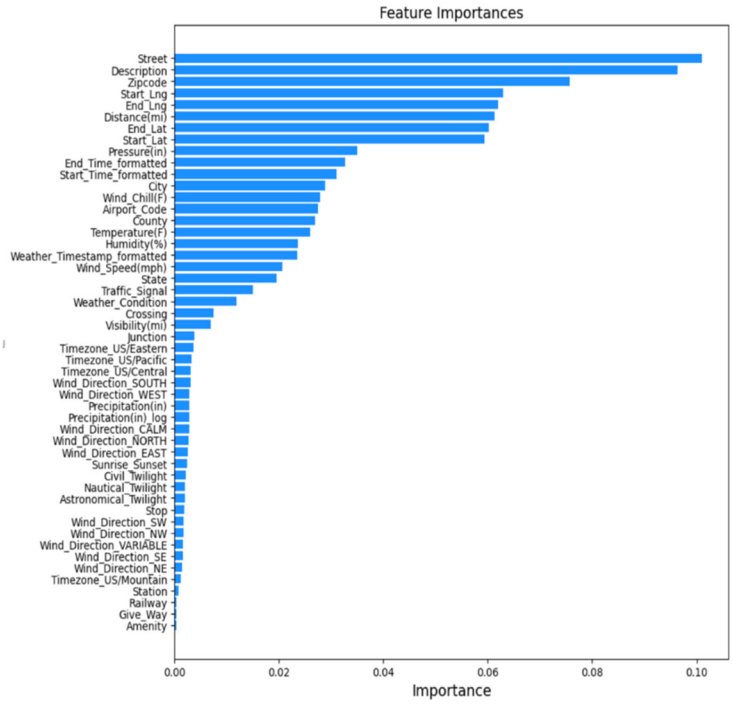

Fig. 6 The bar chart for feature importance

3.1.2Accuracy of XGBoost

The accuracy of the XGBoost model is 0.917. The confusion matrix of its results is presented in Figure 7.

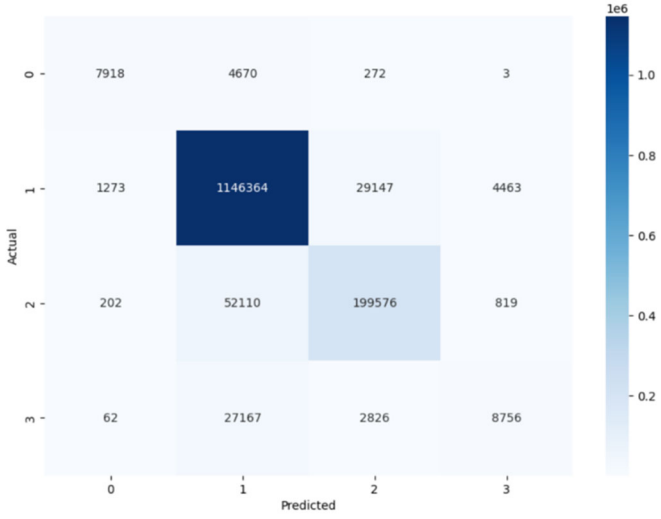

Fig. 7 Confusion matrix for XGBoost classifier

3.2. Efficacy of Voting Classifier

We employed Cross Validation to evaluate the results of the Voting Classifier. The outcomes are presented in Table 8.

Table 8. Improvement in prediction outcomes

scores				
0.91600687	0.91618357	0.9163956	0.91601017	0.91659746

4. Discussion

4.1. Comparative Analysis

4.1.1 Individual Model vs. Voting Classifier Outcomes

The experimental results demonstrate that Random Forest outperforms XGBoost in terms of accuracy.

The output accuracy of the Voting Classifier is lower than the individual performances of the initial two models.

4.2. Significance of Findings

4.2.1 Machine Learning's Role in Traffic Research

Not every algorithm is suitable for every dataset. In this case, Random Forest outperformed XGBoost. Ensemble Methods aren't always superior. While the Voting Classifier aims to improve accuracy by combining several models, it might not outperform the individual models in some scenarios. As previously mentioned, the handling of data, feature engineering, and feature selection can impact the final performance of the model. Using cross-validation is a good practice as it ensures that the model isn't overfitting and provides a more accurate estimation of the model's generalization capabilities. While some models might perform well with default parameters, fine-tuning them might enhance the model's performance further. For instance, XGBoost might perform better with some parameter adjustments.

4.2.2 Implications for Traffic Safety

Given that the ‘street’ feature was of primary importance, it highlights that certain streets or locations might be more prone to severe accidents. The fact that the Voting Classifier did not outperform individual models like Random Forest or XGBoost implies that combining models doesn't always yield better results. For traffic safety implications, this means it might be more effective to rely on one robust model for predictions rather than combining various models.By understanding the factors most influencing traffic accident severity, policy makers can make informed decisions. For example, if time of day consistently appears as a significant factor, then targeted interventions like improved street lighting or increased patrolling during high-risk times could be implemented.

4.3. Limitations and Challenges

Due to the constrained computational resources and the vast dataset size, it was not practical to efficiently carry out multiple rounds of model experimentation.Moreover, the opportunity to employ additional fine-tuning methods for testing and comparison was restricted.

4.4. Recommendations for Future Studies

Exploring the interrelationships between features is another crucial step that could further shed light on their collective impact on prediction accuracy. By understanding how features interact or correlate with each other, we can potentially enhance the predictive power of the model, refine feature selection, and even unearth new insights from the data that might have been overlooked in a more superficial analysis. For instance, one could derive the duration of an event by utilizing the 'End_Time' and 'Start_Time' columns or apply more sophisticated textual analysis on textual data columns.

Additionally, there are more advanced data preprocessing techniques that can be explored with this dataset [13]. For instance, to address the imbalanced data distribution of the target variable 'Severity', the SMOTE (Synthetic Minority Over-sampling Technique) can be employed for balancing [14].

For the final model, fine-tuning can be further enhanced using methods like grid search.

5. Conclusion

The 'street' feature emerged as a paramount indicator in predicting the severity of accidents. Among the machine learning models applied, Random Forest outperformed XGBoost in terms of accuracy. This indicates the robustness of decision trees and the ensemble method in analyzing and predicting complex datasets like traffic data.

The Voting Classifier, an ensemble technique, did not surpass the individual performances of either Random Forest or XGBoost. This serves as a reminder that combining multiple models doesn't guarantee improved accuracy and that sometimes, simpler approaches can be more effective.

Key data preprocessing steps like one-hot encoding, frequency encoding for high-cardinality categories, and handling missing values significantly improved model performance. High model accuracy (e.g., 0.917 for XGBoost) indicates that our selected features are essential in predicting accident severity. This underscores the value of collecting data related to these features.

In summary, our analysis provides a holistic view of the factors influencing traffic accidents' severity. The insights gained can be instrumental in guiding traffic safety measures, influencing policy decisions, and shedding light on areas that require further research.

References

[1] WHO, V. (2018). Global status report on road safety 2018. World Health Organization.

[2] Abdulla, R., Qader, B., & Sdiq, K. (2023). Traffic Accident Traits and Driver Characteristics Implication on Road Accidents using Descriptive Analysis: A Cross Sectional Study in Sulaymaniyah, Iraq. Engineering, Technology & Applied Science Research, 13(2), 10372-10376.

[3] Rana, V., Joshi, H., Parmar, D., Jadhav, P., & Kanojiya, M. (2019). Road accident prediction using machine learning algorithm. International Research Journal of Engineering and Technology (IRJET), 6(03), 0.

[4] Pourroostaei Ardakani, S., Liang, X., Mengistu, K. T., So, R. S., Wei, X., He, B., & Cheshmehzangi, A. (2023). Road Car Accident Prediction Using a Machine-Learning-Enabled Data Analysis. Sustainability, 15(7), 5939.

[5] US Accidents (2016 - 2023). Available online: https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents?datasetId=199387&sortBy=commentCount&language=Python&sort=votes (accessed on 1 August 2023)

[6] Yan M, Shen Y. Traffic accident severity prediction based on random forest [J]. Sustainability, 2022, 14(3): 1729.

[7] Breiman, “Random Forests”, Machine Learning, 45(1), 5-32, 2001.

[8] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." The Annals of Statistics, 29(5) 1189-1232 October 2001.

[9] dmlc.XGBoost, https://xgboost.readthedocs.io/en/latest/tutorials/model.html

[10] Introduction to Boosted Trees-Xgboost v: satble Documentation. Available online: https://xgboost.readthedocs.io/en/latest/tutorials/model.html (accessed on 31 August 2023).

[11] Chakraborty D, Elzarka H. Advanced machine learning techniques for building performance simulation: a comparative analysis[J]. Journal of Building Performance Simulation, 2019, 12(2): 193-207.

[12] Classification report-scikit.learn. Available online: https://scikit-learn.org/stable/modules/model_evaluation.html#classification-report (accessed on 31 August 2023)

[13] Mohanta, B. K., Jena, D., Mohapatra, N., Ramasubbareddy, S., & Rawal, B. S. (2022). Machine learning based accident prediction in secure iot enable transportation system. Journal of Intelligent & Fuzzy Systems, 42(2), 713-725.

[14] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of artificial intelligence research, 2002, 16: 321-357.