Accident Severity
Impact of an Accident on Traffic Congestion
Challenge Project

Pedro Silva, Ramyad Raadi
uc2023235452@student.uc.pt, uc2023205634@student.uc.pt

University of Coimbra, Faculty of Science and Technologies
Bachelor of Data Science and Engineering, ML Course

march 2025

1 Introduction

Reducing traffic accidents is an important public safety challenge. Road accidents not only pose a significant risk to human life, but also contribute to severe traffic congestion, leading to increased travel time, fuel consumption, economic losses, etc. Understanding how accidents may impact traffic congestion can help city planners, traffic managers and the police to develop effective strategies to mitigate delays, improve accident response and improve road safety, in general. Therefore, the main goal of this study is to investigate how accidents can affect traffic congestion.

This study will follow the workflow shown below:

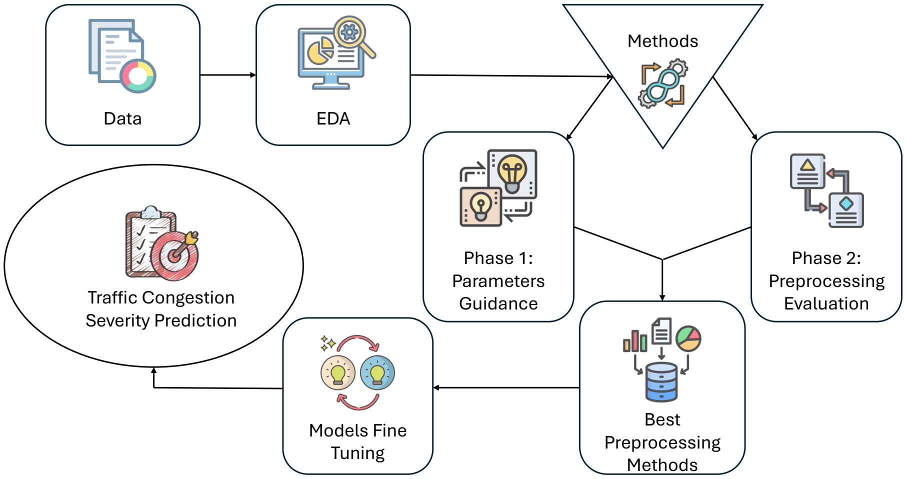

Data
EDA
Methods
Phase 1: Parameters Guidance
Phase 2: Preprocessing Evaluation
Traffic Congestion Severity Prediction
Best Preprocessing Methods
Models Fine Tuning

Figure 1: Graphical Abstract of This Study Workflow

2 Materials

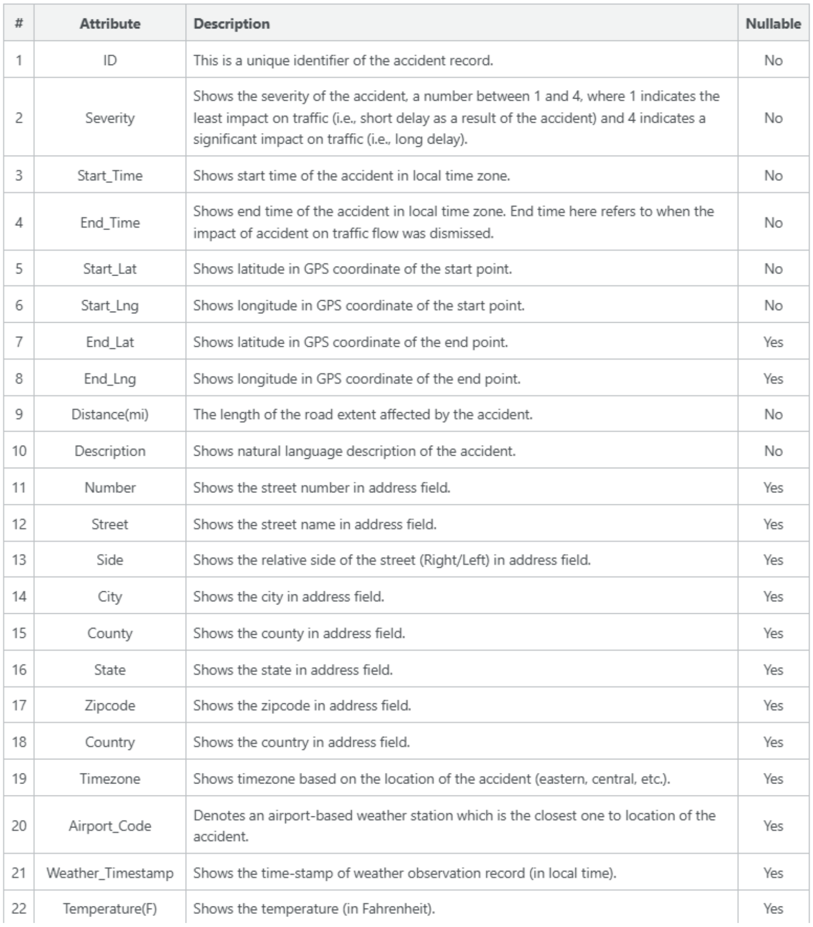
2.1 Data Description

This project considers a country-wide car accident dataset that covers 49 states in the United States. This dataset can be found in `https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents/data` and is distributed by the Creative Commons Attribution-Noncommercial-ShareAlike license (CC BY-NC-SA 4.0). [1]

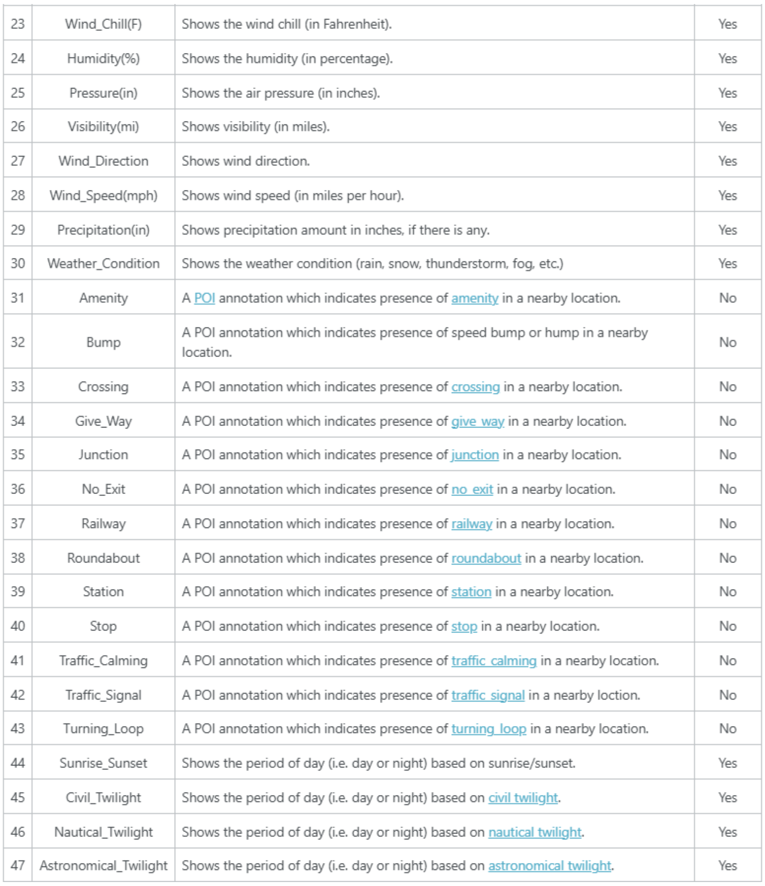
The dataset contains 7.7 million accidents and 47 features, recorded by multiple API's that provide streaming traffic incidents data, from February 2016 to March 2023. For the purposes of this project, we will use a sample of this dataset, provided on the same web page, with 500 000 accidents. The features, and it's description, are in the figure below and you can find more information about the dataset here.[2]

[1] https://creativecommons.org/licenses/by-nc-sa/4.0/

[2] https://smoosavi.org/datasets/us_accidents



#	Attribute	Description	Nullable
1	ID	This is a unique identifier of the accident record.	No
2	Severity	Shows the severity of the accident, a number between 1 and 4, where 1 indicates the least impact on traffic (i.e., short delay as a result of the accident) and 4 indicates a significant impact on traffic (i.e., long delay).	No
3	Start_Time	Shows start time of the accident in local time zone.	No
4	End_Time	Shows end time of the accident in local time zone. End time here refers to when the impact of accident on traffic flow was dismissed.	No
5	Start_Lat	Shows latitude in GPS coordinate of the start point.	No
6	Start_Lng	Shows longitude in GPS coordinate of the start point.	No
7	End_Lat	Shows latitude in GPS coordinate of the end point.	Yes
8	End_Lng	Shows longitude in GPS coordinate of the end point.	Yes
9	Distance(mi)	The length of the road extent affected by the accident.	No
10	Description	Shows natural language description of the accident.	No
11	Number	Shows the street number in address field.	Yes
12	Street	Shows the street name in address field.	Yes
13	Side	Shows the relative side of the street (Right/Left) in address field.	Yes
14	City	Shows the city in address field.	Yes
15	County	Shows the county in address field.	Yes
16	State	Shows the state in address field.	Yes
17	Zipcode	Shows the zipcode in address field.	Yes
18	Country	Shows the country in address field.	Yes
19	Timezone	Shows timezone based on the location of the accident (eastern, central, etc.).	Yes
20	Airport_Code	Denotes an airport-based weather station which is the closest one to location of the accident.	Yes
21	Weather_Timestamp	Shows the time-stamp of weather observation record (in local time).	Yes
22	Temperature(F)	Shows the temperature (in Fahrenheit).	Yes



23	Wind_Chill(F)	Shows the wind chill (in Fahrenheit).	Yes
24	Humidity(%)	Shows the humidity (in percentage).	Yes
25	Pressure(in)	Shows the air pressure (in inches).	Yes
26	Visibility(mi)	Shows visibility (in miles).	Yes
27	Wind_Direction	Shows wind direction.	Yes
28	Wind_Speed(mph)	Shows wind speed (in miles per hour).	Yes
29	Precipitation(in)	Shows precipitation amount in inches, if there is any.	Yes
30	Weather_Condition	Shows the weather condition (rain, snow, thunderstorm, fog, etc.)	Yes
31	Amenity	A POI annotation which indicates presence of amenity in a nearby location.	No
32	Bump	A POI annotation which indicates presence of speed bump or hump in a nearby location.	No
33	Crossing	A POI annotation which indicates presence of crossing in a nearby location.	No
34	Give_Way	A POI annotation which indicates presence of give_way in a nearby location.	No
35	Junction	A POI annotation which indicates presence of junction in a nearby location.	No
36	No_Exit	A POI annotation which indicates presence of no_exit in a nearby location.	No
37	Railway	A POI annotation which indicates presence of railway in a nearby location.	No
38	Roundabout	A POI annotation which indicates presence of roundabout in a nearby location.	No
39	Station	A POI annotation which indicates presence of station in a nearby location.	No
40	Stop	A POI annotation which indicates presence of stop in a nearby location.	No
41	Traffic_Calming	A POI annotation which indicates presence of traffic_calming in a nearby location.	No
42	Traffic_Signal	A POI annotation which indicates presence of traffic_signal in a nearby loction.	No
43	Turning_Loop	A POI annotation which indicates presence of turning_loop in a nearby location.	No
44	Sunrise_Sunset	Shows the period of day (i.e. day or night) based on sunrise/sunset.	Yes
45	Civil_Twilight	Shows the period of day (i.e. day or night) based on civil twilight.	Yes
46	Nautical_Twilight	Shows the period of day (i.e. day or night) based on nautical twilight.	Yes
47	Astronomical_Twilight	Shows the period of day (i.e. day or night) based on astronomical twilight.	Yes

Figure 2: Feature table and it's description. Source: `https://smoosavi.org/datasets/us_accidents`

2.2 Tools and Packages

The project was implemented using the following tools and packages:

- **Scikit-Learn:** Utilized for constructing, training, and evaluating the machine learning models.
- **Pandas, Numpy, Itertools, Collections:** for data manipulation, preprocessing and feature engineering tasks.
- **Imblearn, CategoryEncoders:** Utilized for implementing preprocessing methods.

- **Requests:** to request cities names from OpenWeather API.
- **Scipy:** Utilized for statistics in exploratory data analysis (EDA)
- **Matplotlib, Seaborn and Plotly:** Used for visualizing the data and understanding trends in EDA.

3 Exploratory Data Analysis (EDA)

In this section, we'll explore in depth the dataset. To do this task, we'll try to answer some questions about the data with graphics, tables and statistics. The following subsections divide the main aspects that we think that are relevant, in this problem context.

3.1 Data Overview

In data overview, we studied:

1. **Inspection of data types and missing values:** Here, we discovered that we will handle integers, floats, objects (strings) and booleans. There are some objects that are not constituted by a single string. In "Description" variable, we have a set of strings, and in "Start_Time" feature we have a timestamp, but in a set of strings.
2. **Missing values:** detection of missing values. We discovered that "End_Lng" and "End_Lat" have around 44% of missing values. There are other features that have a high rate of missing values and we will disconsider them in our work (Table 4). We'll handle them in Feature Selection/Extraction section.

3.2 Severity Analysis

In severity analysis, we studied:

1. **Severity Distribution:** The proportion of accidents by severity levels are on Figure 3. We can notice that the classes are very unbalanced: the class 2 has around 80% of the data. Meanwhile, the class 3 has around 16% and the others have less than 3%. Furthermore, we'll transform the severity feature into a variable with 2 classes (severe or not severe), so it will be almost 80% for severe congestion and 20% for

the other class. With this insight, we need to handle with the unbalance. Under sampling the majority class is the tool that we worked with to make the classes more balanced.

2. **Factors Influencing Severity:** We concluded that there are not any continuous or binary variables that influences linearly severity, using a correlation heatmap (Figure 4). Besides that, we inspected the categorical variables that were important (ID, for example, don't give any information) to check if there was correlation, using Cramér's V correlation matrix (Figure 5), and we concluded that "City", "County", "Airport_Code", "State" and "Timezone" were correlated with each other, as well as the "Sunrise_Sunset", "Civil_Twilight", "Nautical_Twilight", and "Astronomical_Twilight" variables.
   Both of correlation matrices will be important on the Feature Selection section.

3.3 Spacial Analysis

In spacial analysis, we studied:

1. **Geographical Accident Distribution:** Analyzing the Figure 6 and Table 5, we determined that the top 10 states with higher number of accidents in the USA are: California, Florida, Texas, South Carolina, New York, North Carolina, Pennsylvania, Virgina, Minnesota and Oregon.

2. **State and City Analysis:** In Table 6, we concluded that the top 10 cities with most accidents are: Miami, Houston, Los Angeles, Charlotte, Dallas, Orlando, Austin, Raleigh, Nashville and Baton Rouge. Despite California being the state with more accidents (in double, compared with Florida in the top 2), it only appears one city in the top 10, in third place.

3. **Missing cities Analysis** There were 19 missing cities in the data and we thought that the there were a relationship between the missing cities and some external factor. So, we request the names of the cities to OpenWeather API, using the coordinates of the accident. We discovered that, in 19 missing cities, 7 were "Fairmount Heights" and 3 were "Glassmanor", indicating that probably went something wrong during a while (with the sensors/actuators, for example) assigning this cities into the dataset.

3.4 Temporal Analysis

In temporal analysis, we studied:

1. **Accident Trends Over Time:** In Table 7 and in Figure 7, we can clearly affirm that the number of accidents are increasing year by year. Note that we have less accidents in 2023 because this dataset only has data until march 2023. Also, we can infer that between 2016 and 2022, the number of accidents increased 4 times.

2. **Time of Month Analysis:** Analyzing the Figure 8, we can deduce that in the winter, more in specific, between November and February, we have more accidents than in summer/spring. December is the month where we have more accidents, in average.

3. **Time of Week Analysis:** In the Figure 9, it's evident that weekdays there are more accidents compared to weekends, with weekend accident frequencies being at least 2/3 times lower. This trend may be attributed to the reduced volume of vehicles on the road during weekends (since the majority of the people work in weekdays). Also, we can say that the number of accidents in Friday is statistical higher than the other weekdays, by 5% of significance levels. We applied the Mann-Whitney U test (Non-parametric test), and we obtained `p_value=0.0168<0.05`.

4. **Time of Day Analysis:** Analyzing the Figure 10, it's clear that between 7h to 9h and 16h to 18h we have more accidents. This corresponds to the rush hours in USA. In addition, we have more accidents during the day rather in night as spectated and there are more accidents in the afternoon than in the morning.

3.5 Environmental Factors Analysis

In environmental factors analysis, we studied:

1. **Weather Conditions:** In figure 11, and we infer that the majority of the accidents were on clear or cloudy weather, wich is counterintuitive. This can relate to traffic distractions being one of the main issues in traffic accidents [3].

3.6 Road Analysis

Finally, in road conditions analysis, we studied:

1. **Road Types and Conditions:** Our study revealed that the top 3 most frequented road features involved in traffic accidents were in traffic signals, in crossings and in junctions (Figure 12). Note that we can deduce that roundabouts are a lot safer than junctions, since there are almost none accidents in roundabouts.

4 Methods

After exploring the data, we're going to work on the main goal of this project: predicting traffic congestion severity. To achieve this, it is essential to preprocessing the data. In order to use a good preprocessing, we design two setups of experiments:

1. **Phase 1:** The first phase consists into plan of our prediction of what could be the best preprocessing workflow, with several experiments to check the general performance, using recall, precision and f-score metrics, of tree basic models - Decision Trees, K-Nearest Neighbors and Naive Bayes - to predict what could be the best hyperparameters for methods and models.

2. **Phase 2:** The second phase aims to assess how different preprocessing methods affect model performance when using the baseline classifiers. The key goal is to determine the most effective preprocessing strategy before fine-tuning the models, using a design of experiments.

The following subsections explain our workflow to complete both phases:

4.1 Phase 1: Feature Selection/Extraction

The goal of Feature Selection/Extraction is to provide better conditions for the models. Extracting features helps highlight the most relevant patterns in the data, improving model performance and interpretability, while feature selection aims to remove redundant or irrelevant variables, reducing dimensionality and computational complexity, retaining essential information. The work done in this subsection is indexed below:

1. **Duration of an accident extraction:** We extracted the duration of each accident making the difference between the "End_Time" and the "Star_Time". This new feature can be relevant for predicting Severity.

2. **Extraction of year, month, weekday and day:** This extraction, from "Star_Time", was very helpful in EDA to understand trends and patterns, but it is also important for the models, since it's stratified and simple to use.

3. **Reduce the number of weather conditions:** This task was made in EDA too and it was crucial because we had more than 100 unique weather conditions. Now we have less (12) and more readable weather conditions. The missing values of this variable were handled too.

4. **Bining of Severity:** This bining was made to try to help the model to get better results, leaving it with two classes: class 0 - not severe; class 1 - severe.

5. **Drop of columns:** Selecting the relevant columns is essential to reduce dimensionality and computational complexity, retaining essential information. The dropped columns, and the explanation of why they were dropped, are below:

   (a) `"ID"`, `"Source"`, `"Description"`, `"End_Lat"`, `"End_Lng"` because they do not provide substantial information for our analysis;

   (b) `"End_Time"`, `"Start_Time"` because it was made an extraction of what timestamp of the starting time of the accident;

   (c) `"Airport_Code"`, `"Country"`, `"County"`, `"County"`, `"State"`, `"Street"` , `"Timezone"`, and `"Zipcode"` because coordinates are enough for localizing the accident;

   (d) `"Weather_TimeStamp"`, `"Sunrise_Sunset"`, `"Civil_Twilight"`, `"Nautical_Twilight"`, and `"Astronomical_Twilight"` because these fields may not be directly relevant to our analysis;

   (e) `"Temperature(F)"`, `"Wind_Chill(F)"`, `"Humidity(%)"`, `"Precipitation(in)"` , `"Pressure(in)"`, and `"Wind_Direction"` because of a combination of several missing values and these fields could not make sense to our analysis.

4.2 Phase 1: Missing Values

We already checked if there were missing values, but knowing that we drop almost half of the variables, its evident that we need to do it again. In fact, we reduced the list of features with missing values, as spectated. So, we had to handle with "Wind_Speed(mph)" and "Visibility(mi)" missing values.To handle with them, we used the median of the existed values.

4.3 Phase 1: Data Preparation

4.3.1 Balancing and Under Sampling the Dataset

To get better results whether in model predictions, whether in reducing the computational complexity, its crucial to balance both classes of severity (Not Severe: 80%; Severe: 20%). For that, we used the random undersampling, with sampling_strategy set to 0.7, since the dataset has a more natural balance of 55.7% for class 0 and 44.3% for class 1.

4.3.2 Splitting Data into train and test

We split the data in way to get 80% of them into training and 20% for the test part.

4.3.3 Leave-One-Out Encoding

Since the models and the Tomek Links method (next subsection) need numeric data, it is imperative to use encoding methods for `Weather_Condition`. The Leave-One-Out (LOO) Encoding is a encoding technique used for categorical variables. It replaces each category with the mean of the target value, calculated by excluding the current row to prevent data leakage. This method helps retain useful information while reducing overfitting compared to standard target encoding.

4.3.4 Tomek Link Method

To help models classify the samples, we used Tomek Link method. Tomek Links is an undersampling technique used to clean class imbalances in datasets. It identifies and removes pairs of nearest neighbors where one belongs to the majority class and the other to the minority class.

4.3.5 Data Standardization

It is crucial to standardize data for the models, specially for KNN, since the weights of each features are decisive.

4.3.6 Principle Components Analysis

Principle Components Analysis, or PCA, is a dimensionality reduction technique that transforms high-dimensional data into a smaller set of uncorrelated variables called principal components (PC's). It captures the most important information by maximizing variance, helping to reduce noise and improve model efficiency. To set the best number of PC's, we created the Scree Plot of PCA (Figure 13) and we concluded that 20 PC's would be a good trade off between the explained variance and reducing dimensionality.

4.4 Phase 1: Training Models

As discussed previously, the three models used are:

1. **Decision Tree (DT):** A hierarchical model that splits data based on feature conditions, creating a tree-like structure. It is easy to interpret but can overfit without pruning or depth constraints. For this phase, max_depht hyperparameter was set to 20 as default.

2. **K-Nearest Neighbors (KNN):** Considered a lazy algorithm that classifies a data point based on the majority class of its k-nearest neighbors. It is simple but can be computationally expensive for large datasets - curse of dimensionality. For this phase, n_neighbors hyperparameter set to 7 as default.

3. **Naive Bayes (NB):** A probabilistic model based on Bayes theorem, assuming feature independence. It is fast and effective but may struggle when feature independence is violated.

4.4.1 Testing Hyperparameter Tuning

To better evaluate the hyperparameters of the KNN and Decision Tree models, we used precision, recall, and F-score. Additionally, we generated a graphic that illustrates the model's performance for each chosen hyperparameter. For `max_depth`, we tested the values 3, 7, 10, 15, 20, and 30. For `n_neighbors`, we used 3, 5, 7, 11, and 21, as shown in Figures 14 and 15, respectively.

4.5 Conclusion of Phase 1:

After all these tests, we concluded that the best hyperparameters for each method/model are:

1. **Random Undersampling Technique:** `sampling_strategy` = 0.7
2. **PCA:** `n_components` = 20
3. **Decision Tree:** `max_depth` = 10
4. **KNN:** `n_neighbors` = 3 (this value could vary, once the metrics don't change for different k's)

4.6 Phase 2: Feature Selection

The feature selection is almost the same as in phase 1. The major differences are that this time we will test 2 different feature selections: the first implemented in phase 1 - Feat. Selection default and the other one in other columns selection - Feat. Selection T1. Compared to the default feature selection, the T1 selection includes `Humidity(%)`, `Pressure(in)`, and `Civil_Twilight`. These features provide more environmental context, which may help in understanding how weather conditions influence accident occurrences. Additionally, T1 don't have `Bump`, which was included in the default selection.

By incorporating these new variables, feature selection T1 aims to enhance model performance by leveraging weather-related factors, which might capture different accident patterns compared to the default selection.

4.7 Phase 2: Missing Values

By selecting new columns - to check if there are any difference to the previous feature selection - it's mandatory to verify once again the missing values. Besides the detected missing values earlier in phase 1, we found that the added features contained missing values. To solve this issue, we applied the median of the samples of each feature into the missing values on `Humidity(%)` and `Pressure(in)`, and the mode on `Civil_Twilight`, since we transformed it into a binary variable.

4.8 Phase 2: Data Preparation

This subsection is very similar to the first phase. However, to improve the preprocessing, for each experiment, we ran the process five times, changing the `random_state` in each run. This helped reduce the impact of randomness in the data split and performance results. The following tables indicates what type of experiment it was done and it's results.

Experiment	Feat. Selection def	Feat. Selection T1	PCA	Tomek Links
1-Default	Yes	No	Yes	Yes
2-Feat. Selection	No	Yes	Yes	Yes
3-PCA	Yes	No	No	Yes
4-Tomek Links	Yes	No	Yes	No
5-No PCA/Tomek	Yes	No	No	No
6-Undersampling	Yes	No	Yes	Yes

Table 1: Comparison of Different Preprocessing Methods

Experiment	Model	Precision	Recall	F-score
1-Default	Decision Tree	0.61	0.66	0.63
	KNN	0.61	0.64	0.62
	Naive Bayes	0.44	0.92	0.60
2-Feat. Selection	Decision Tree	0.62	0.63	0.63
	KNN	0.60	0.63	0.62
	Naive Bayes	0.45	0.91	0.60
3-PCA	Decision Tree	0.51	0.00	0.01
	KNN	0.62	0.65	0.63
	Naive Bayes	0.43	0.97	0.59
4-Tomek Links	Decision Tree	0.62	0.61	0.62
	KNN	0.62	0.59	0.60
	Naive Bayes	0.45	0.92	0.60
5-No PCA/Tomek	Decision Tree	0.51	0.00	0.01
	KNN	0.63	0.59	0.61
	Naive Bayes	0.43	0.97	0.59
6-Undersampling	Decision Tree	0.61	0.66	0.63
	KNN	0.60	0.64	0.62
	Naive Bayes	0.44	0.93	0.60

Table 2: Performance of Different Preprocessing Methods Across Models

Here are some notes in order to understand better the experiments done:

1. By default, it is understood that this is the best preprocessing obtained in phase 1.
2. In all experiments, the undersampling was equal (`random_state` $= 17$), except in experiment 6, where both `random_state` from undersampling and data splitting were changed between runs.

4.9 Results of preprocessing

In summary, after analyzing the first two experiments, we found that the new feature selection (T1) did not improve or worsen performance. The F-score remained the same, and the other metrics were similar in both experiments. This suggests that using fewer features (default feature selection) is just as effective. Additionally, we observed that using PCA helps the Decision Tree achieve better results, while Tomek Links slightly improves performance. Naive Bayes, on the other hand, maintains the same performance regardless of preprocessing changes.

Based on these findings, we conclude that the best preprocessing approach involves using default feature selection, as it reduces the number of features while maintaining performance. Furthermore, combining PCA and Tomek Links provides better results. Lastly, we confirmed that different undersampling techniques do not significantly affect the performance metrics.

4.10 Fine-Tuning

We have already performed some initial fine-tuning of the model hyperparameters. However, in this section, we will explore a more comprehensive approach, testing additional parameters to find the best model configurations. To achieve this, we will use `GridSearchCV`, which systematically searches through combinations of hyperparameters to identify the most effective settings.

The following hyperparameters will be tested for each model:

- **Decision Tree**
  - Maximum depth: {7, 9, 11, 13}
  - Minimum samples split: {2, 5, 10}
  - Minimum samples leaf: {1, 2, 4}

- **K-Nearest Neighbors (KNN)**
  - Number of neighbors: {3, 7, 15, 17, 19}
  - Weights: {"distance"}
  - Distance metric: {"euclidean", "manhattan"}
- **Naive Bayes**
  - No hyperparameter tuning required.

To ensure robust evaluation, we apply `GridSearchCV` with:

- **5-fold cross-validation** for reliable performance estimation.
- **30 independent runs** to ensure result stability.
- **F1-score optimization**, while also tracking precision, recall, and accuracy.

After running the hyperparameter tuning process using `GridSearchCV`, we identified the best-performing configurations for each model. The table below summarizes the best parameters for each model, along with their corresponding precision, recall and F-score metrics.

Model	**Best Parameters**	**Accuracy**	**Precision**	**Recall**	**F-score**
Decision Tree	Max Depth: 11 Min Samples Split: 2 Min Samples Leaf: 5	0.69	0.69	0.69	0.69
KNN	Neighbors: 19 Weights: distance Metric: euclidean	0.73	0.73	0.73	0.73
Naive Bayes	*No hyperparameter*	0.52	0.45	0.63	0.52

Table 3: Best Hyperparameters and Performance Metrics for Each Model

5 Results and Discussion

The results of the experiments provide valuable insights into the impact of different preprocessing techniques and model configurations. Various approaches were tested, and their effectiveness was assessed based on precision, recall, F1-score, and accuracy, in fine-tuning.

Analyzing the table above, we can retrieve that the best model achieved was KNN, with all metrics scoring 73%, wich is reasonable, taking into account the road safety context and the difficulty of prediction, since it involves countless variables. Checking the confusion matrix (figure 16), we can infer that the best model, KNN, classifies correctly most cases but struggles with misclassifications. It detects 20,000 non-severe and 13,344 severe congestion cases correctly but also produces 7,256 false positives and 6,402 false negatives. This means that the false positive rate might lead to unnecessary interventions and the false negative rate could be problematic for traffic management.

One of the key findings, that was not mentioned in this report, was the limitation of using SMOTE combined with Tomek Links. While SMOTE generated synthetic samples for the minority class, Tomek Links removes firstly the tomek links. This contradiction led us to think and we decided to not test it.

Furthermore, converting the original four-class classification into a binary classification problem introduced potential drawbacks. By merging severity levels, the models had to generalize over a larger range of cases, which may have affected performance, as we talked to professor Francisco. This aligns with previous research, such as the work by Amritesh Kumar [11], who used the same dataset and achieved a performance of 79%. This suggests that, while the current models show promising results, attending the dificulty of prediction in the fiel of road safety, there is still room for improvement with more refined feature selection, hyperparameter tuning, and advanced modeling techniques.

Despite these challenges, some preprocessing techniques, such as feature selection, dimensionality reduction through PCA and Tomek Link method demonstrated positive effects in improving model stability and reducing overfitting.

6 Future Work

While this study provides valuable insights into accident severity classification, several areas could be explored to further improve performance. One key aspect is refining feature selection, as the initial choice of features plays a crucial role in model effectiveness. Additionally, integrating preprocessing steps with fine-tuning in a more dynamic manner could lead to better results. Rather than treating these as separate phases, a more adaptive approach where preprocessing techniques are selected based on model feedback could optimize performance.

Alternative sampling techniques should also be considered, as the combination of Random Undersampling and Tomek Links could have limitations.

Beyond model selection, deeper feature engineering could contribute to better performance. Additionally, exploring deep learning approaches, such as neural networks, could provide a more powerful way to extract linear and non-linear patterns from the dataset, potentially leading to more accurate predictions.

By implementing these improvements, a future study can aim to develop more robust models, leading to a better traffic congestion severity prediction and more effective decision-making in traffic management and road safety.

7 Acknowledgments

This project was a key learning experience in understanding Machine Learning. From data preprocessing to model tuning, every step helped us handle the challenges of building and optimizing predictive models. It was a hands-on journey that made us appreciate the importance of good data, the right features, and careful fine-tuning to achieve good results.

We would like to mention that AI [ChatGPT] was used to:

1. Correct some syntax errors and make the text more cohesive and coherent.

2. Get some insights into what we could explore in EDA.

3. On understanding `GridSearchCV` and other functions hyperparameters.

4. In creating tables in LateX.

5. Building guide lines between sections.

8 References

1. Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath. “A Countrywide Traffic Accident Dataset.”, 2019.
   Check https://arxiv.org/abs/1906.05409.

2. Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu Teodorescu, and Rajiv Ramnath. ”Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights.” In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2019.
   Check https://arxiv.org/abs/1909.09638.

3. https://www.nhtsa.gov/risky-driving/distracted-driving

4. https://www.kaggle.com/code/kelixirr/us-accidents-severity-prediction-end-to-end#EDA---Exploratory-Data-Analysis

5. https://www.kaggle.com/code/yasirkhan1811/us-accidents-data-analysis-2016-2023

6. https://www.kaggle.com/code/whisgv/us-accidents-analysis

7. https://www.kaggle.com/code/muhammadfurqan0/us-data-analysis-beginner-notebook

8. https://www.kaggle.com/code/mohammedhamdy98/us-accidents-v2#Impact-of-Traffic

9. https://www.kaggle.com/code/whisgv/us-accidents-analysis

10. https://www.kaggle.com/code/mariusborel/us-accident-state-and-year-maps/notebook

11. https://www.kaggle.com/code/kelixirr/us-accidents-severity-prediction-end-to-end

12. https://www.geeksforgeeks.org/smote-for-imbalanced-classification-with-python/

13. https://stats.stackexchange.com/questions/97555/handling-unbalanced-data-using-smote-no-big-difference

14. https://medium.com/@rafaelnduarte/class-weight-smote-random-over-and-under-sampling-bca603378e02

15. https://scikit-learn.org/stable/modules/grid_search.html#multimetric-grid-search

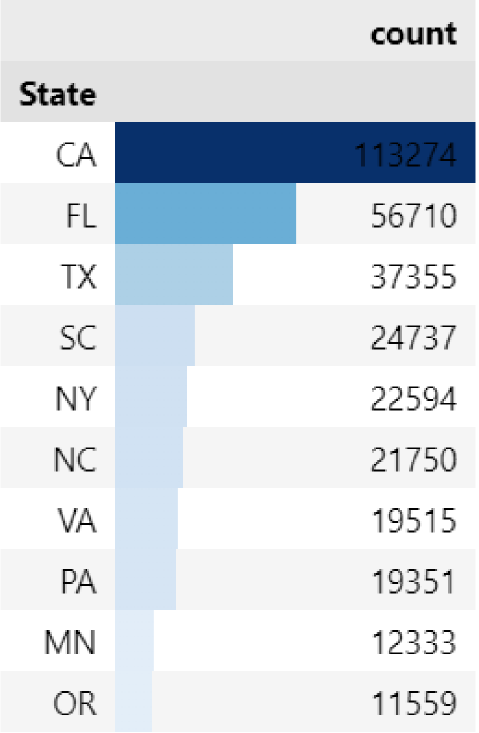
16. https://medium.com/@manindersingh120996/understanding-categorical-correlations-with-chi-square-test-and-cramers-v-a54fe153b1d6

17. https://www.geeksforgeeks.org/calculate-cramer-s-coefficient-matrix-using-pandas/

9 Tables and Figures

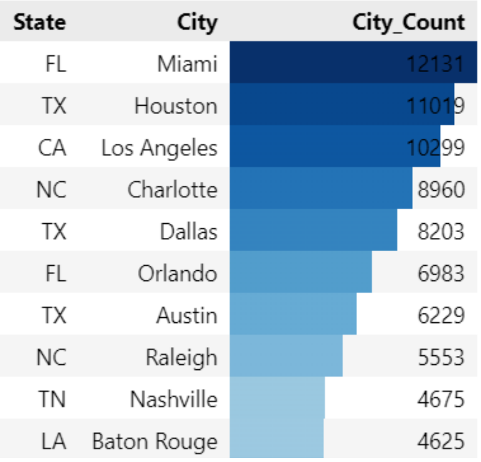
Feature	**Missing values** (%)
End_Lng	44.0754
End_Lat	44.0754
Precipitation(in)	28.5232
Wind_Chill(F)	25.8034
Wind_Speed(mph)	7.3974
Visibility(mi)	2.2582
Wind_Direction	2.2394
Humidity(%)	2.2260
Temperature(F)	2.0932
Pressure(in)	1.7856
Weather_Timestamp	1.5348
Nautical_Twilight	0.2966
Sunrise_Sunset	0.2966
Civil_Twilight	0.2966
Astronomical_Twilight	0.2966
Airport_Code	0.2892
Street	0.1382
Timezone	0.1014
Zipcode	0.0232
City	0.0038
Description	0.0002

Table 4: Missing Data Percentage for Features



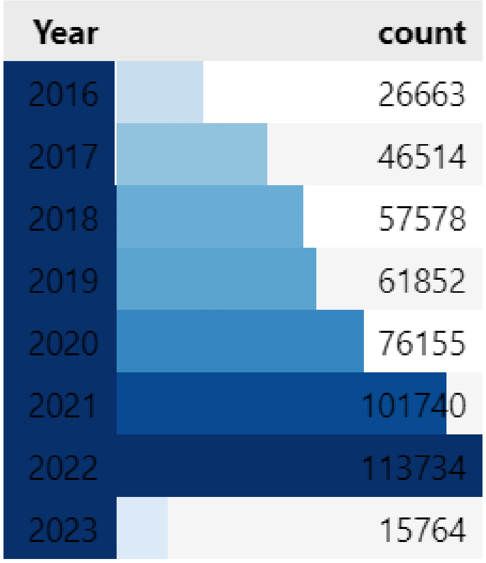
	count
State	
CA	113274
FL	56710
TX	37355
SC	24737
NY	22594
NC	21750
VA	19515
PA	19351
MN	12333
OR	11559

Table 5: Accidents per state



State	City	City_Count
FL	Miami	12131
TX	Houston	11019
CA	Los Angeles	10299
NC	Charlotte	8960
TX	Dallas	8203
FL	Orlando	6983
TX	Austin	6229
NC	Raleigh	5553
TN	Nashville	4675
LA	Baton Rouge	4625

Table 6: Accidents per city



Year	count
2016	26663
2017	46514
2018	57578
2019	61852
2020	76155
2021	101740
2022	113734
2023	15764

Table 7: Accidents per year

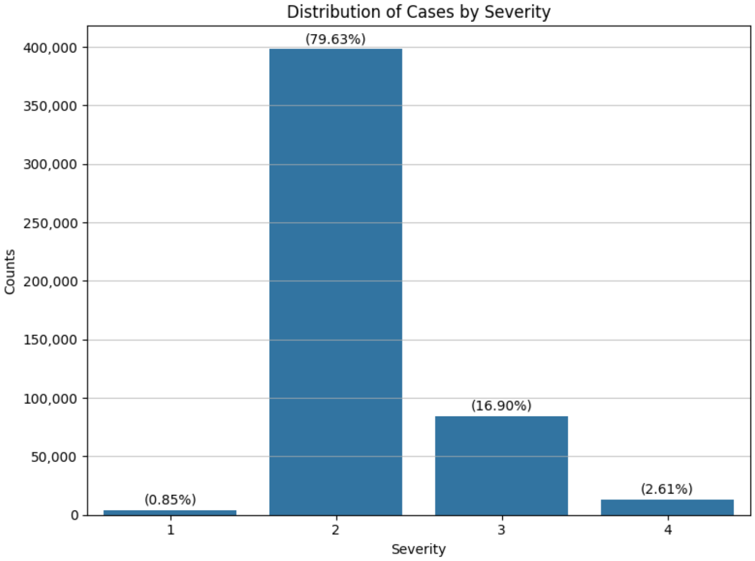


Figure 3: Distribution of accidents by severity

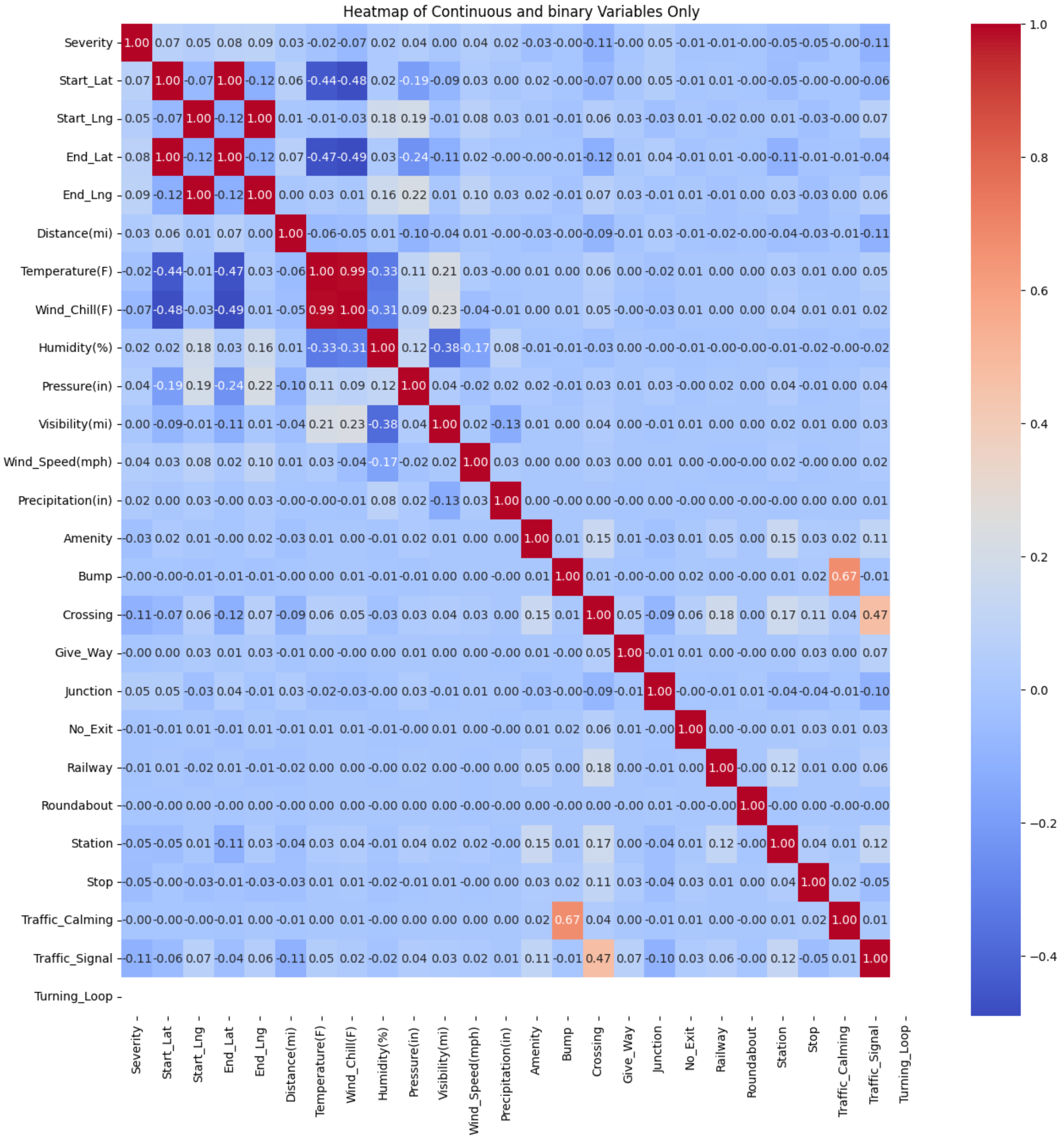


Figure 4: Correlation Matrix: Continuous and binary features

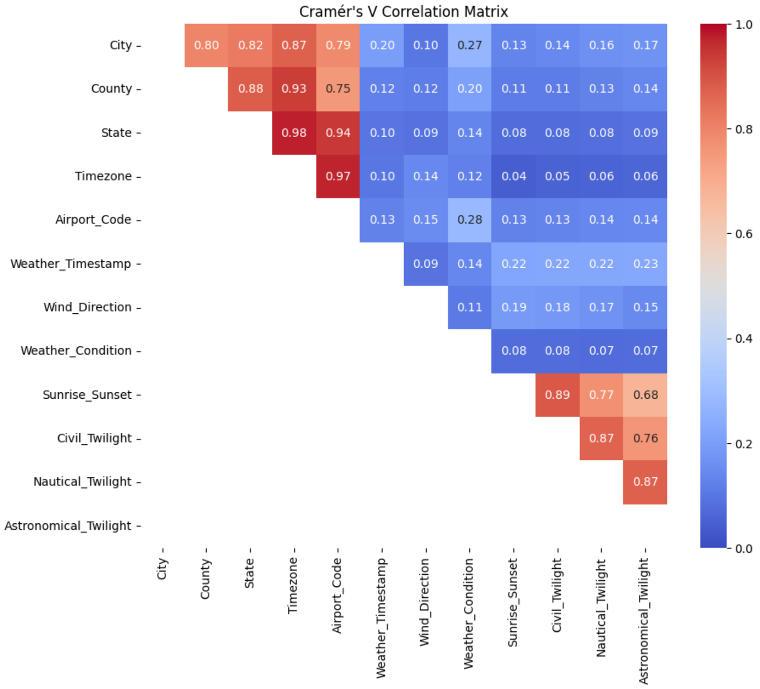


Figure 5: Crámer’s V Correlation Matrix

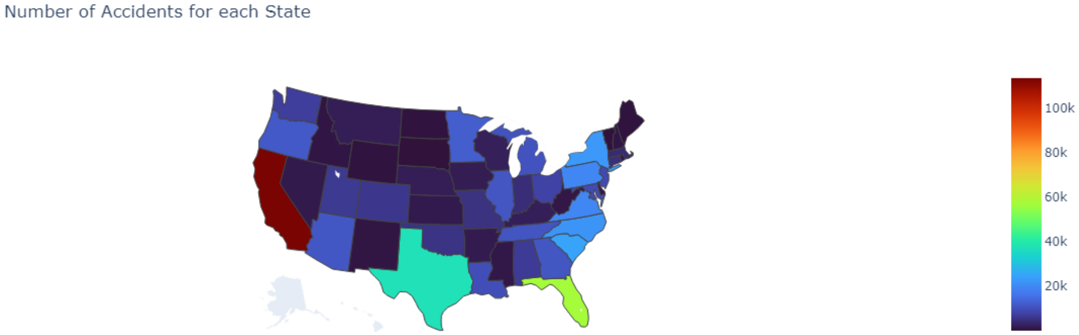


Figure 6: Accidents in USA map, including states

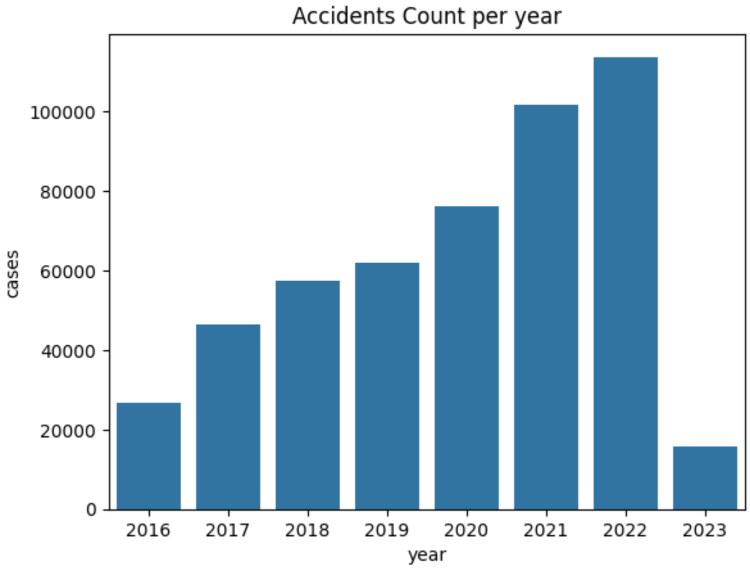


Figure 7: Accidents per year

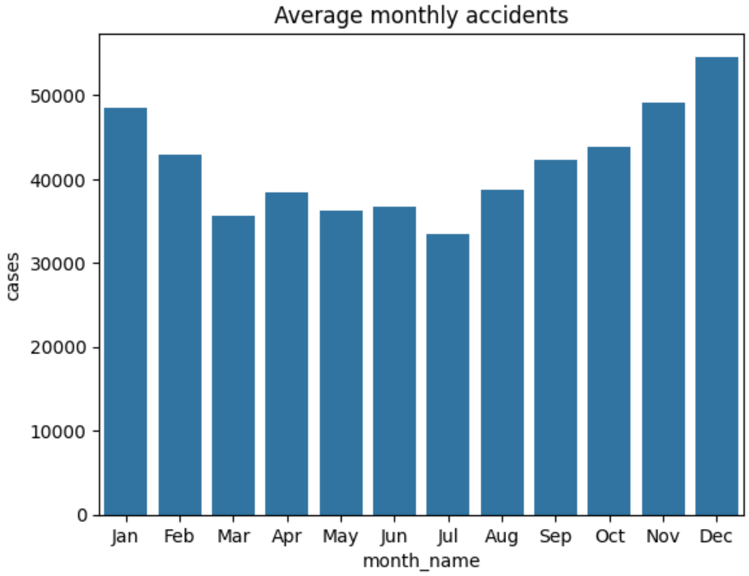


Figure 8: Accidents per month, in average

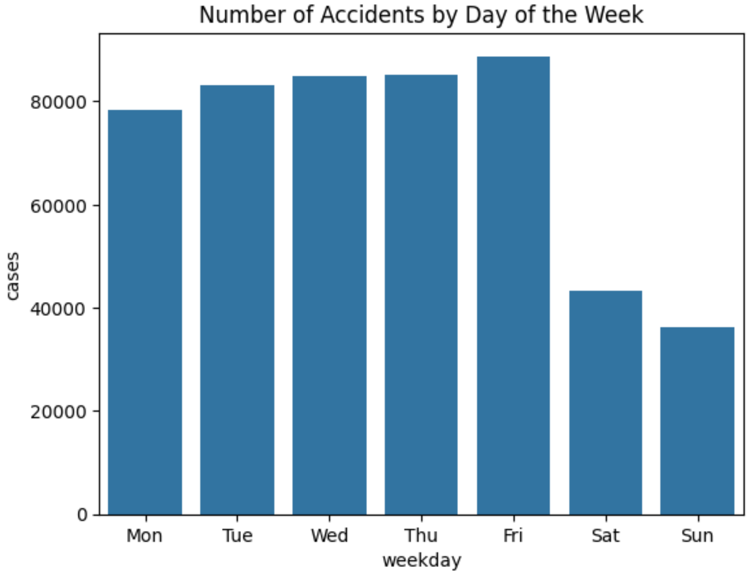


Figure 9: Accidents per day of the week, in average

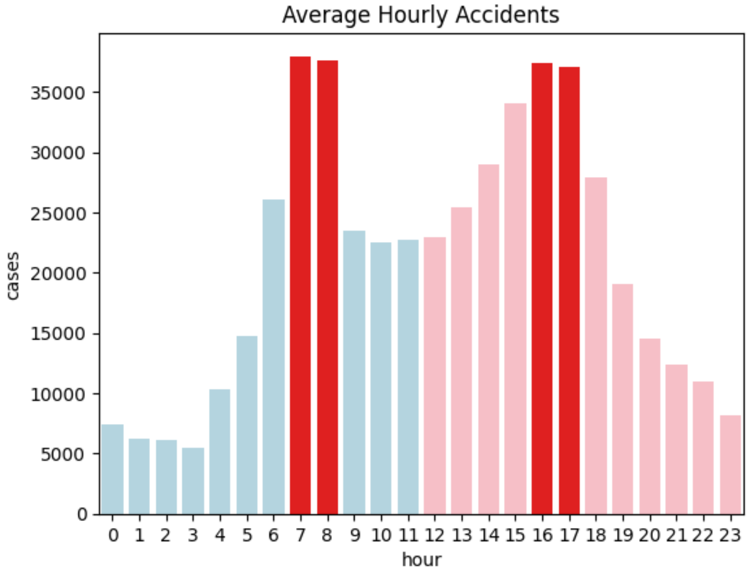


Figure 10: Accidents per hour, in average

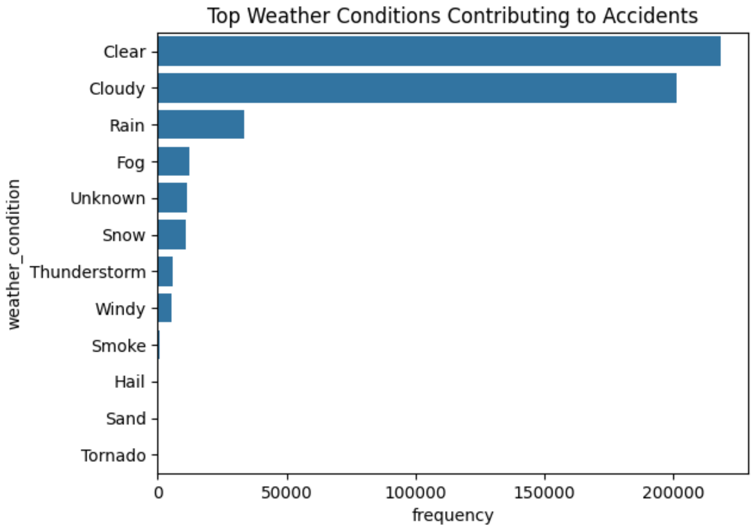


Figure 11: Top weather conditions on accidents

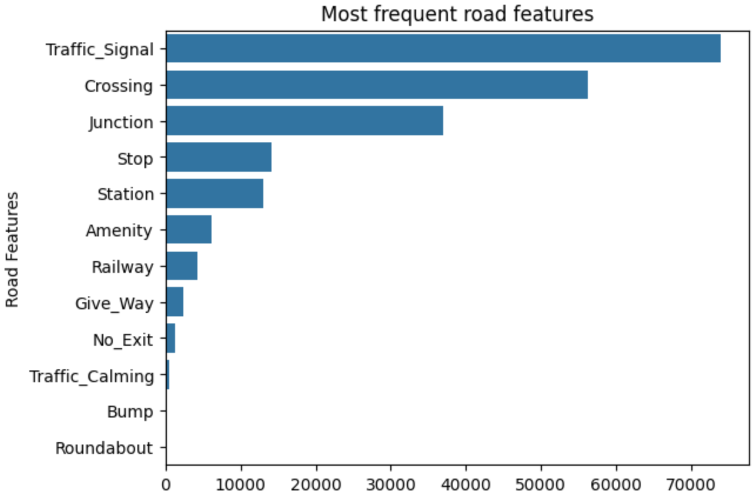


Figure 12: Most frequent road features on accidents

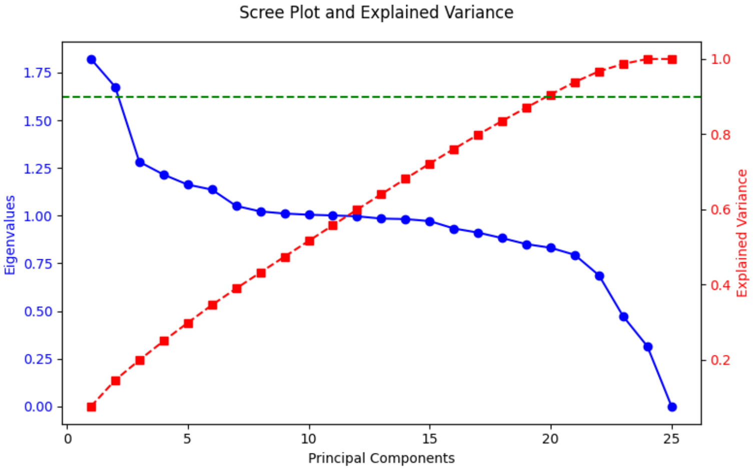


Figure 13: PCA's Scree Plot

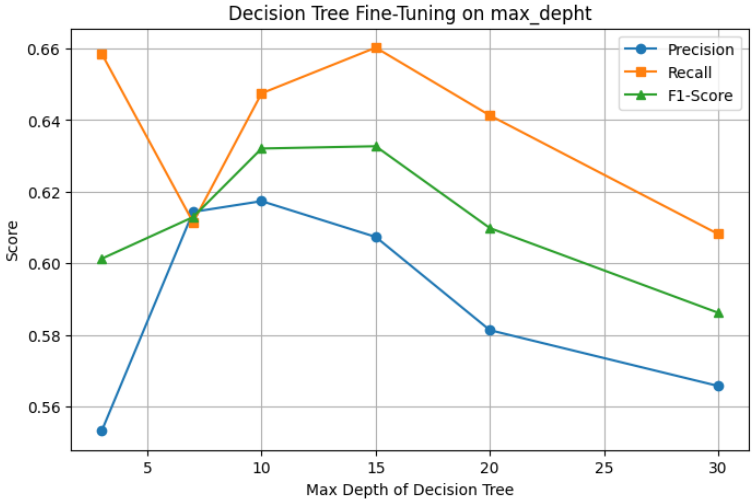


Figure 14: Fine-tuning Decision Tree

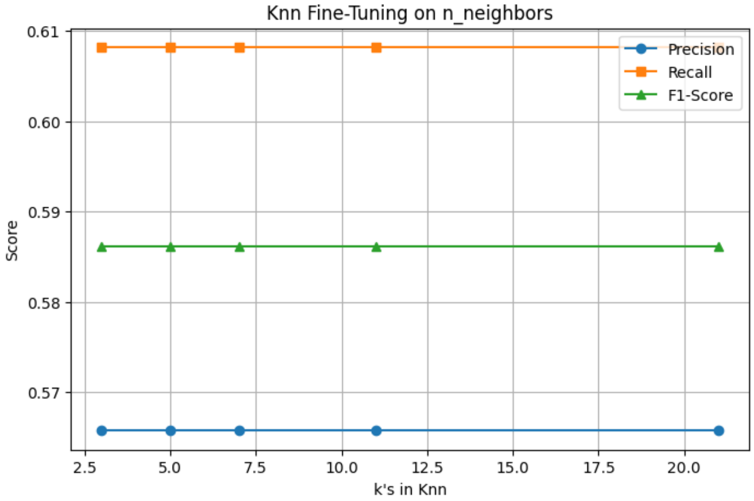


Figure 15: Fine-tuning KNN

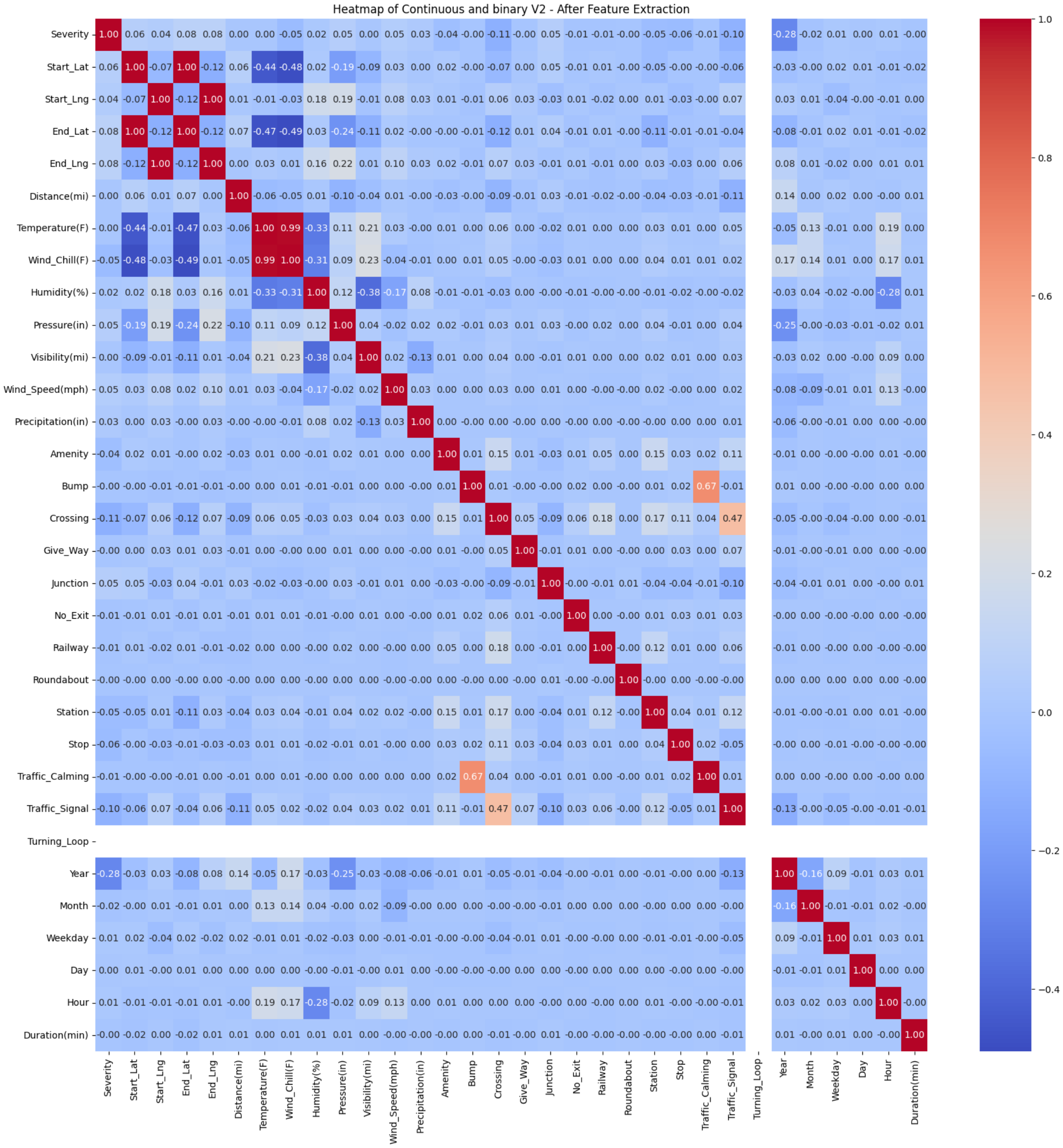


Figure 16: Correlation matrix, including extracted features

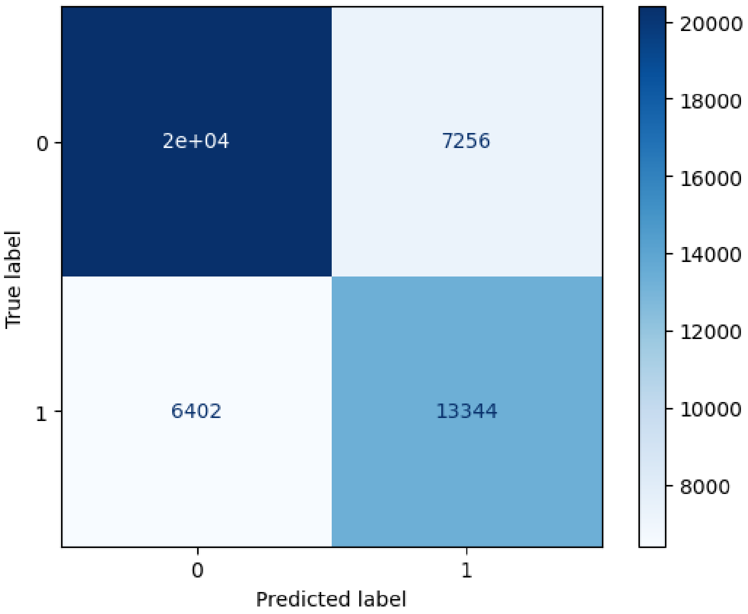


Figure 17: Confusion Matrix