UNIVERSITÀ DI PISA

Data Mining

Data Mining Report - Group 7

Students:

Difranco Giuliano - mat. 578893
Marco Lavorini - mat. 581307

Lecturers:

Prof. Anna Monreale
Prof. Mattia Setzu
Prof. Lorenzo Mannocci

Academic Year 2024/25

Contents

1. Understanding

In the first phase of the project, we analyzed the provided datasets[1]. Three datasets were available: `crashes`, `people` and `vehicles`, containing 912,164, 2,003,340, and 1,860,568 entries, respectively. The primary identifier is the `CRASH_RECORD_ID`. This identifier can be used to link records across the two datasets, `people` and `vehicles` to the main dataset, `crashes`. The `crashes` dataset provides insights into the general conditions and details of each incident while excluding any personally identifiable information. For each entry, the dataset includes information such as weather and road conditions, the number of units involved, the presumed primary cause, and other relevant details.

1.1 Original Datasets

The following analyses were performed on the three original datasets, prior to their aggregation into the monthly incident profile for each department. This preliminary examination enabled us to identify and fix various issues in the data.

1.1.1 Feature Description

Due to the large amount of feature, we refer to the official site for a complete feature description[1]. Since almost all features were stored as objects (i.e., strings), our primary preprocessing step was to convert key columns into appropriate data types. In particular, we converted date columns into datetime objects and transformed columns containing "Y"/"N" values into booleans. Numerical columns were correctly stored and loaded as integer/float.

1.1.2 Incoherent and missing data

For all three datasets, no dataset-specific unique identifiers had any duplicate. Many columns had more than 99% missing values (e.g., `DOORING_I` and `WORK_ZONE_TYPE`) and were dropped.

According to the data source, the records date from 2015 onward. Consequently, eight entries from the `crashes` dataset with dates prior to 2015 were removed. Moreover, since city-wide data is only available from September 2017 onward, we will later restrict our analysis to data from 2018 to 2024 to ensure a consistent monthly distribution for each department when creating the monthly-department profile.

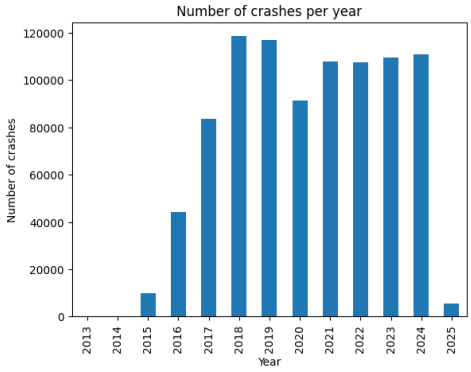

Figure 1.1: Crashes per year

For the `POSTED_SPEED_LIMIT` column, we replaced all entries with a value of 0 (approximately 4000 entries) and all anomalous values (e.g., 1, 2, and 11, which accounted for around 400 entries) with `NaN`.

Due to the dataset structure, where most features are categorical with string values, analyzing the missing data was not an easy task, as missing values could be represented as `UNKNOWN`, `NOT APPLICABLE`, or many other strings. Therefore, we decided to focus only on the columns of interest and set

these entries to -1, so that we could handle them later during Feature Engineering. Finally, for the `crashes` dataset we decide to drop any rows with missing `BEAT_OF_OCCURRENCE`.

For the `people` dataset, the first step was to only keep entries which had a `CRASH_RECORD_ID` also present in the `crashes` dataset in order to reduce the dataset size from 2003340 to 1667514 entries.

The `AGE` column contained several incorrect entries: 11 entries had a negative age, and 15,526 entries reported an age of 0. Since an age of 0 might be valid for passengers, we analyzed the distribution according to the type of person involved in the crash (driver or non-driver). Ultimately, we decided to keep only the people categorized as drivers in the dataset, as identified by the `PERSON_TYPE` column and set all ages ≤ 0 to `NaN`. The `AGE` column reported a 29% missing value ratios.

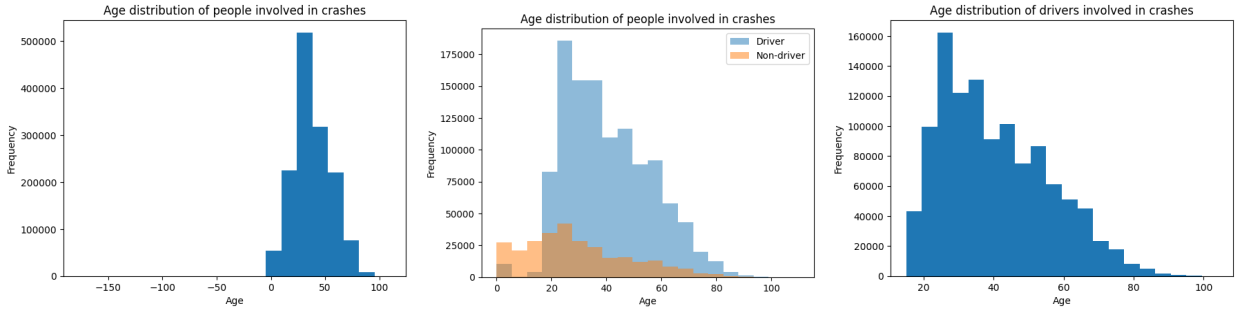

Figure 1.2: Age distribution: (i) all values, (ii) values excluding negatives and grouped by person category, and (iii) only drivers after pre-processing.

For the `vehicles` dataset, the same pre-processing of removing all entries without a valid `CRASH_RECORD_ID` was performed. We then observed how the number of vehicles associated with each crash did not match the number reported in the `NUM_UNIT` column of the `crashes` dataset for 1726 crashes. We decided to use the values from the `crashes` dataset for our analysis.

1.1.3 Duplicates

We checked the coordinates of the crashes using Plotly [2] and noticed that 1415 coordinates were pointing to the same exact spot, as shown in figure 1.3. Therefore, we set these coordinates to `NaN`.

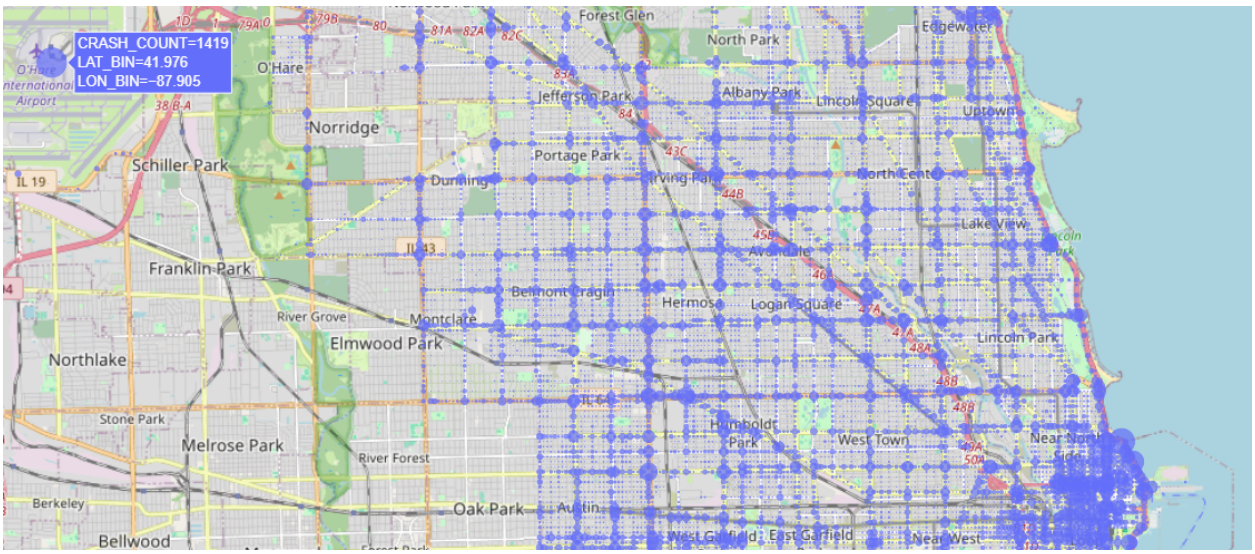

Figure 1.3: Cluster of the same coordinates pointing to the airport in the top-left corner.

For the `people` dataset, we observed that, even after removing all non-driver entries, 1391 crashes had a number of driver exceeding the number of vehicles involved in the crash, as for the feature `NUM_UNIT`. To address this discrepancy, we constructed a quasi-identifier (QID) for the people as follows:

$$\begin{aligned}\text{QID} = \{&\texttt{CRASH\_DATE}, \texttt{CITY}, \texttt{STATE}, \texttt{ZIPCODE},\\ &\texttt{SEX}, \texttt{AGE}, \texttt{DRIVERS\_LICENSE\_STATE}, \texttt{DRIVERS\_LICENSE\_CLASS}\}\end{aligned} \tag{1.1}$$

By aggregating the data according to these attributes, we identified 2790 potential duplicate entries, and, as shown in Figure 1.4, we determined that 77% of our inconsistent data was due to these duplicates. We then decided to drop these value.

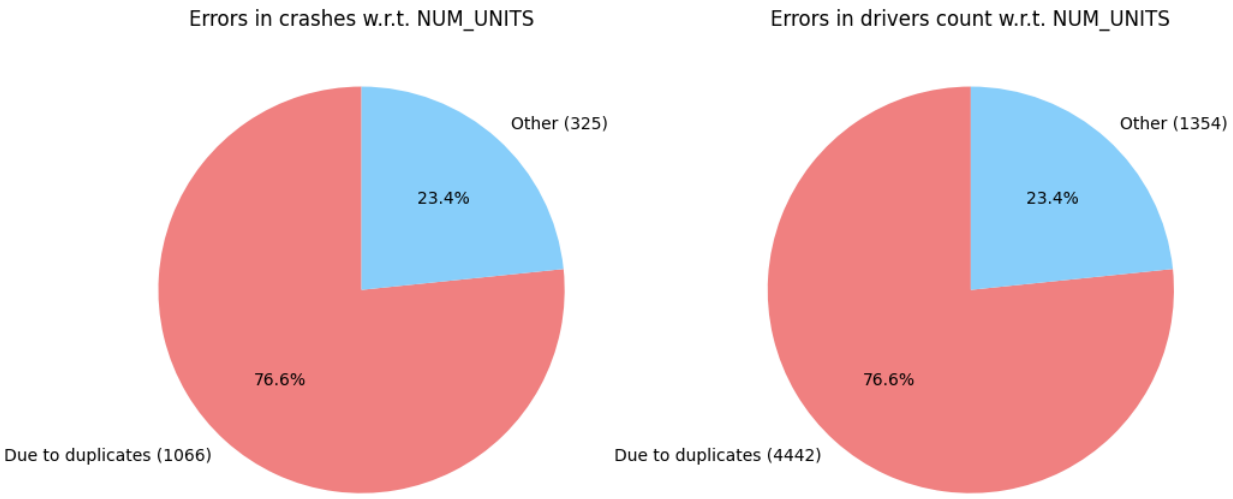

Figure 1.4: Percentage of duplicate on the miss-matched data between number of units and number of drivers

The same approach was applied to the `vehicles` dataset. In this case, the quasi-identifier was defined as follows:

$$\begin{aligned}\text{QIDv} = \{&\texttt{CRASH\_DATE}, \texttt{VEHICLE\_DEFECT}, \texttt{UNIT\_TYPE}, \texttt{VEHICLE\_YEAR},\\ &\texttt{VEHICLE\_TYPE}, \texttt{MANEUVER}, \texttt{OCCUPANT\_CNT}, \texttt{FIRST\_CONTACT\_POINT}\}\end{aligned} \tag{1.2}$$

We found that 4,665 vehicles had at least one duplicate entry. By dropping these duplicates, the number of mismatched crashes was reduced from 1,563 to 451, with nearly 16,000 entries removed from the `vehicles` dataset overall.

1.1.4 Outliers

We analyzed the `AGE` feature in the `people` dataset after removing non-driver entries. We then decided to set to `NaN` the age of the driver records with an age less than 15—the legal driving age in America—or greater than 104, in order to eliminate extreme cases (a total of 12 entries).

We also analyzed other extreme values occurrences in the available numerical features, such as `NUM_UNITS`. The few cases with a high number of vehicles corresponded to the number of entries in the vehicle dataset; hence, we decided to retain these records.

1.1.5 Feature Engineering

Due to the dataset structure, we defined a severity score for various categorical features. In this approach, categories with similar severity levels were grouped together and assigned an increasing score based on the

perceived severity. Values that could not be classified were assigned a default value of -1. For instance, the weather condition feature was transformed using the mapping in Table 1.1.

Category	Severity Score
CLEAR	0
CLOUDY/OVERCAST	1
RAIN	2
FREEZING RAIN/DRIZZLE	2
FOG/SMOKE/HAZE	3
SNOW	3
SLEET/HAIL	3
BLOWING SNOW	3
SEVERE CROSS WIND GATE	3
BLOWING SAND, SOIL, DIRT	4
OTHER	-1
UNKNOWN	-1

Table 1.1: Mapping of weather conditions to severity scores.

A similar approach was applied to other features, as detailed in following Table 1.2.

Feature	Score Range	Description
weather_severity	0–4	Severity of weather using WEATHER_CONDITION.
lighting_severity	0–2	Severity of lighting using LIGHTING_CONDITION.
crash_severity	0–1	Severity of the crash computed as a weighted score of the fatality and incapacitating rates for each crash type.
roadway_severity	0–6	Combination of ROAD_DEFECT and ROADWAY_SURFACE_CATEGORY.
responsibility_score	0–1	Severity of the responsibility of the crash cause, derived from PRIM_CONTRIBUTORY_CAUSE.

Table 1.2: Mapping of original features to score ranges and their descriptions. Long descriptions will wrap to the next line.

Additionally, we extracted several new features from the dataset, such as:

1. tow: a lower bound counter of crashes involving a towed vehicle, derived from the CRASH_TYPE and INJURIES_TOTAL features when the value INJURY AND / OR TOW DUE TO CRASH was present.
2. neo_patented_driver: an indicator for drivers aged 21 or younger.
3. senior_driver: an indicator for drivers aged 65 or older.

Finally, a variety of additional categorical features were extracted, covering aspects such as vehicle type and the principal cause of the crash. The principal cause was further categorized into three main groups: speeding, drinking, and the use of the telephone. The new categorical values were represented as boolean columns or counters, designed to facilitate grouping for the creation of the incident profile.

1.1.6 Distribution and Trends

Figure 1.5 illustrates both the yearly and monthly distributions of crashes. The data shows a significant reduction in crash occurrences during the pandemic, particularly evident in April—the first full month of lockdown. Furthermore, the monthly distribution shows that the majority of crashes tend to occur during the summer months.

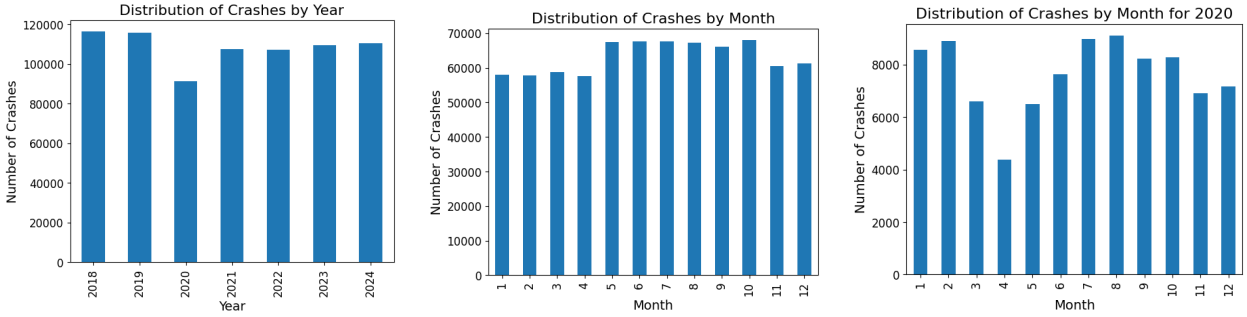

Figure 1.5: Crashes distribution (i) by year, with the clear effect of the pandemic, (ii) by month, (iii) for the pandemic year

From Figure 1.6, we observe interesting trends in the hourly distribution of crashes. Notably, the distribution differs on weekends, with an increase in crashes during the early morning hours.

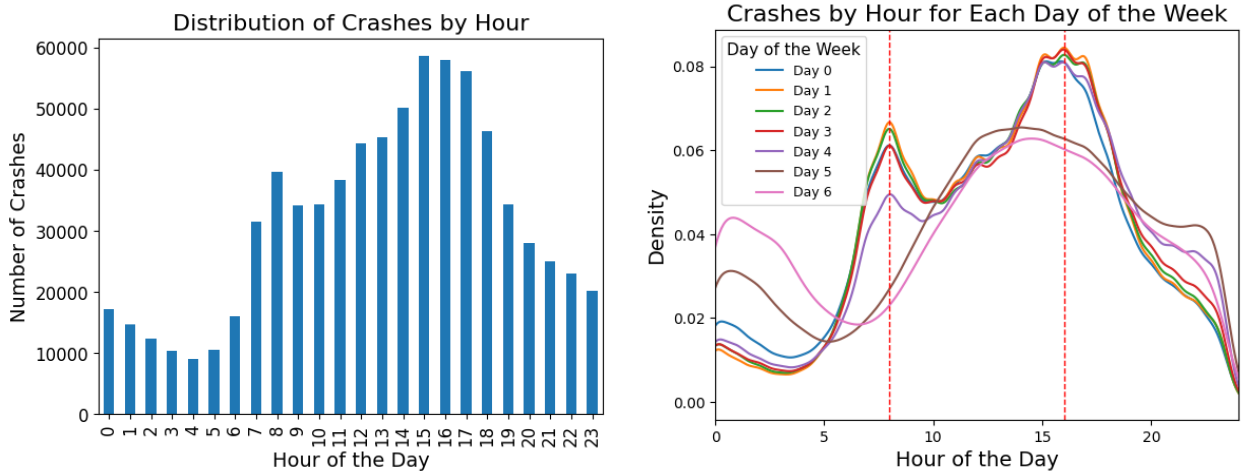

Figure 1.6: Crash distributions: (i) by hour of the day and (ii) grouped by day of the week, showing a different pattern for weekends. (with 8:00 A.M. and 4:00 P.M. highlighted)

By looking at the rate of fatal and incapacitating injuries with respect to the crash type, in Figure 1.7 we can observe how the two feature correlates. As mentioned in Table 1.2 we used this correlation to create our `crash_severity_score` feature.

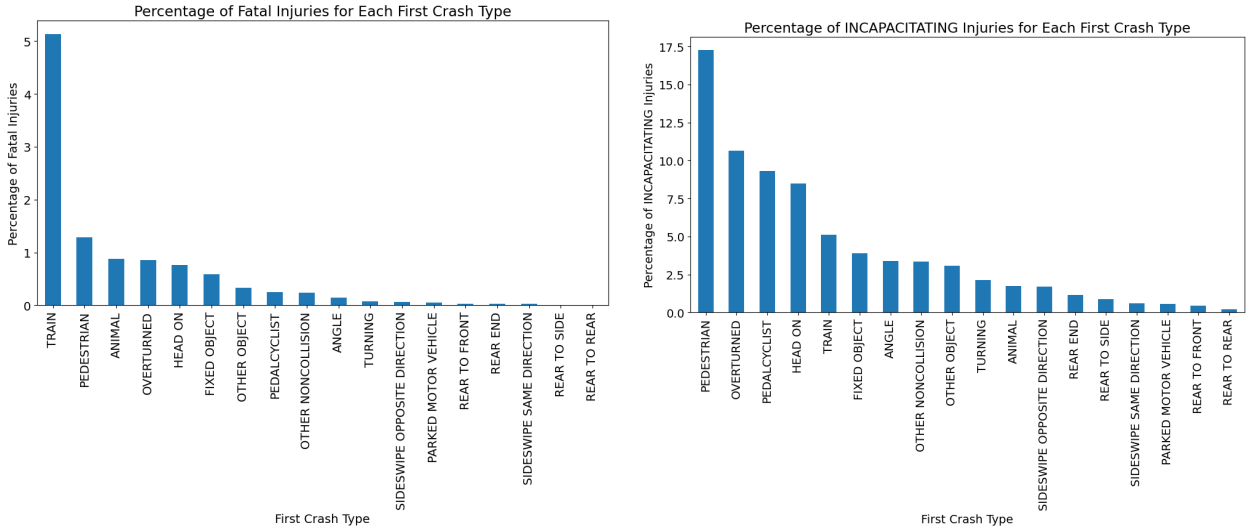

Figure 1.7: (i) Fatal and (ii) Incapacitating injuries ratio with respect to the crash type.

By grouping the average posted speed limit by each age, we observe a clear trend in Figure 1.8: older individuals tend to be involved in crashes in areas with lower speed limits.

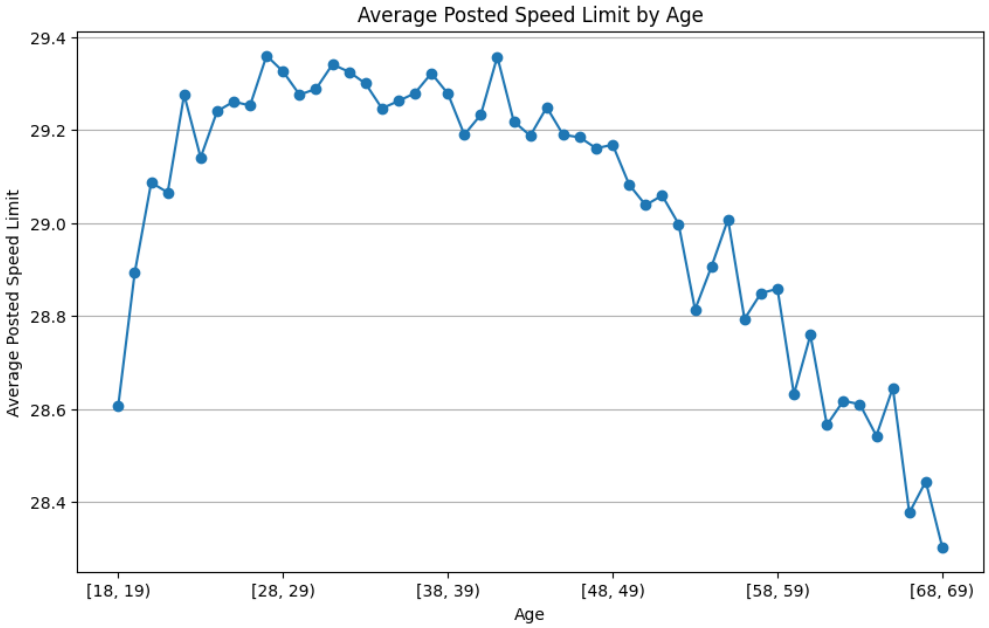

Figure 1.8: Trend of the average posted speed limit with respect to each age category.

1.2 Incident Profile

The project required the creation of an incident profile for each month and year for the police department. The first step in achieving this goal was to merge the three datasets prior to grouping the data by these three indicators.

1.2.1 Merging the Datasets

As described in Section 1.1.5, most features in the `people` and `vehicles` datasets were treated as categorical variables. This approach allowed us to simply sum the occurrences of these features when grouping by crash, effectively merging the information with the `crashes` dataset. However, this method was not applicable to the `AGE` feature. Although we created categorical groupings for age, we decided to retain the actual age values by calculating an average for each crash. To support this, we computed both the sum of the available ages and a count of the non-null entries, which enables us to derive the average age per crash in subsequent aggregations taking in consideration the missing age values.

1.2.2 Creation of the Incident Profile

As mentioned earlier, we grouped the data by month, year, and police department, resulting in a new dataset containing 22,680 entries (with each department contributing 84 entries, corresponding to 7 years of data with 12 months per year). During this aggregation, we primarily focused on computing the average score for the newly created categories relative to the total count for that category over the month—be it the number of crashes, people, units, or injured individuals. Some of the final features defining the incident profile were:

Feature	**Description**
`avg_age`	Average age of all the people involved in crashes during the month.
`avg_responsibility_score`	Average responsibility score, derived from crash causes, ranging from 0 to 1.
`avg_weather_severity`	Average monthly weather score derived from the weather conditions.
`avg_roadway_severity`	Average monthly roadway condition score derived from both road conditions (e.g., potholes) and road state (e.g., wet).
`avg_lighting_severity`	Average monthly lighting severity, ranging from daylight to dark.
`total_crashes`	Total number of crashes counted for the specific month and department.
`fatality_rate`	Percentage of fatal injuries relative to total injuries.
`severe_injury_rate`	Percentage of incapacitating injuries relative to total injuries.
`avg_crash_severity_score`	Average crash severity score obtained based on the severity of injuries (severe and deadly) for each type of crash.
`injury_severity_score`	A weighted index aggregating different injuries with various weights to reflect overall severity.
`vehicle_involvement_rate`	Percentage of vehicles involved in crashes for the department during the month, relative to the total vehicles involved across all departments for the same month.
`combined_weather_road_severity`	A weighted metric combining weather severity (30% weight) and roadway severity (70% weight) to capture an overall environmental risk factor for the crashes.
`total_units_department`	Total number of vehicles (units) involved in crashes for the department during the month.
`night_crash_rate`	Rate (proportion) of crashes that occurred during nighttime.
`speeding_influence`	Percentage of crashes occurred in area with a posted speed limit higher than 30.
`weekend_crash_rate`	Rate of crashes that occurred during the weekend.
`pct_neo_patented_drivers`	Proportion of neo-patent drivers relative to total crashes.
`pct_senior_drivers`	Proportion of senior drivers relative to total crashes.
`num_towed_units_LB`	Lower bound for the total number of towed unit.
`damage_cost_LB`	Lower bound for the total damage cost.

Table 1.3: Some of the new features describing the incident profile.

For some features, such as `AGE`, we opted to calculate both the sum of ages and the count of available entries. This approach ensures that we can compute the mean age for all individuals involved in crashes, aggregated by month for each department. Furthermore, we kept counters for many categories, such as the number of specific vehicles involved for each month(e.g., motorcycles, trucks, etc.) .

1.2.3 Data Quality

By analyzing the Incident Profile dataset, we identified the following missing value rates for all the features, as shown in Table 1.4.

Feature	Missing Value Rate
`avg_responsibility_score`	0.114638
`avg_roadway_severity`	0.013228
`avg_age`	0.004409
`combined_weather_road_severity`	0.013228

Table 1.4: Final missing value rates for selected incident profile features.

Furthermore, we observed that most missing value entries, as well as outliers (mainly for the severity features), originated from profiles with very few crashes per month. Specifically, the majority of these missing values appeared in the 1st percentile of the dataset when considering the total number of crashes distribution across all profiles, hence we decided to drop all the profiles with less than 7 crashes per month.

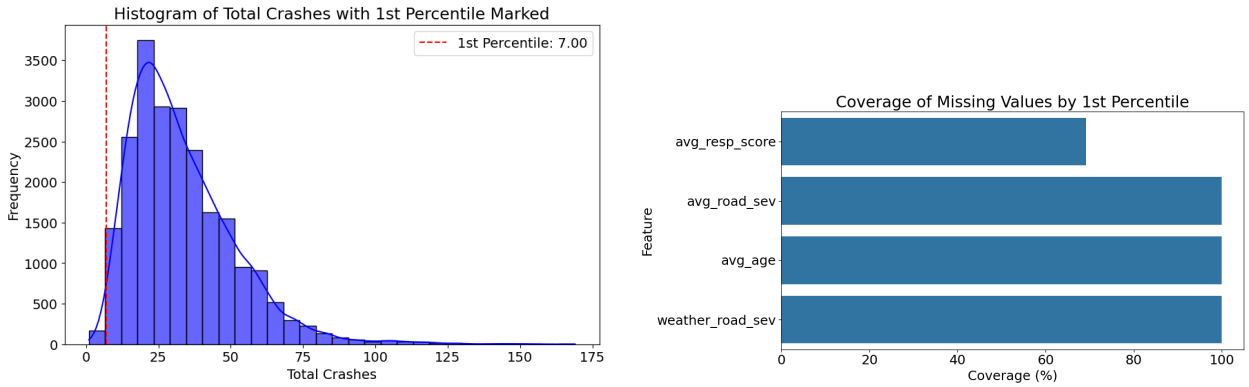

Figure 1.9: (i) Crashes distribution and (ii) ration of the missing value distribution in the 1st percentile.

We imputed the remaining missing values for `avg_responsibility_score` using the average value computed for the same department in the same month, but in a different year.

1.2.4 Correlation

In Figure 1.10, we present the correlation matrix for several key features. To maintain clarity, not all features are included; for instance, we omit many of the categorical features. After examining the highly correlated features, we removed redundant ones. For example, since `night_crash_rate` was strongly correlated with `avg_lighting_severity`, we opted to use only `avg_lighting_severity`. Moreover, we observed an high correlation between `avg_weather_severity` and `avg_roadway_severity`, hence we decided to merged these into a new feature, `combined_weather_road_severity`. This new feature was constructed by weighting the original features at 0.3 and 0.7, respectively, given that the roadway feature was already derived from two other features in the original dataset (see Table 1.2).

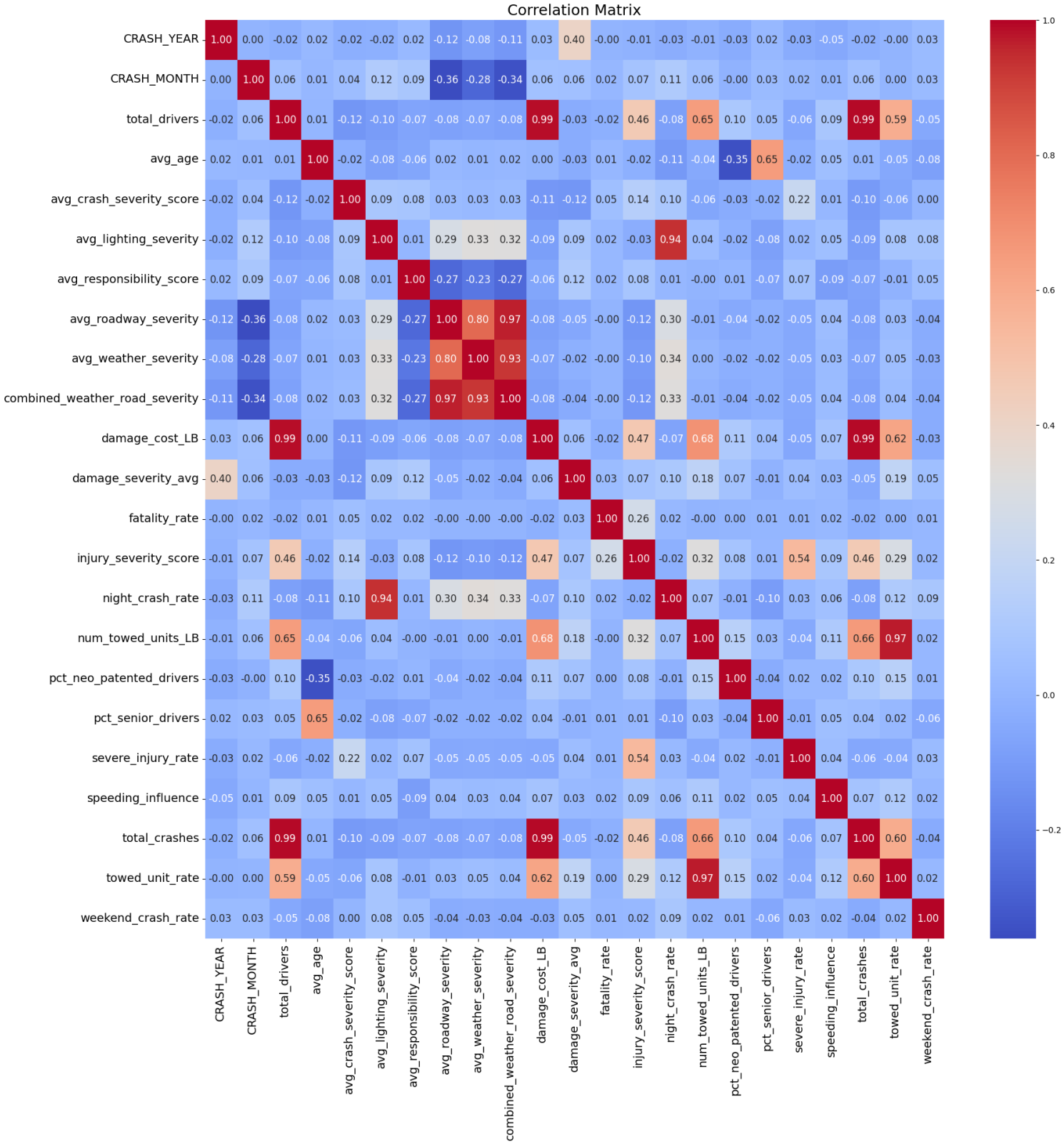

Figure 1.10: Correlation of some of the main features that define the Incident Profile dataset.

1.2.5 Further Data Analysis

In figure 1.11 , we can see both the distribution of the `combined_weather_road_severity` feature and we can also observe a clear trend when confronted with its distribution for each month, with the feature having higher values in the winter months.

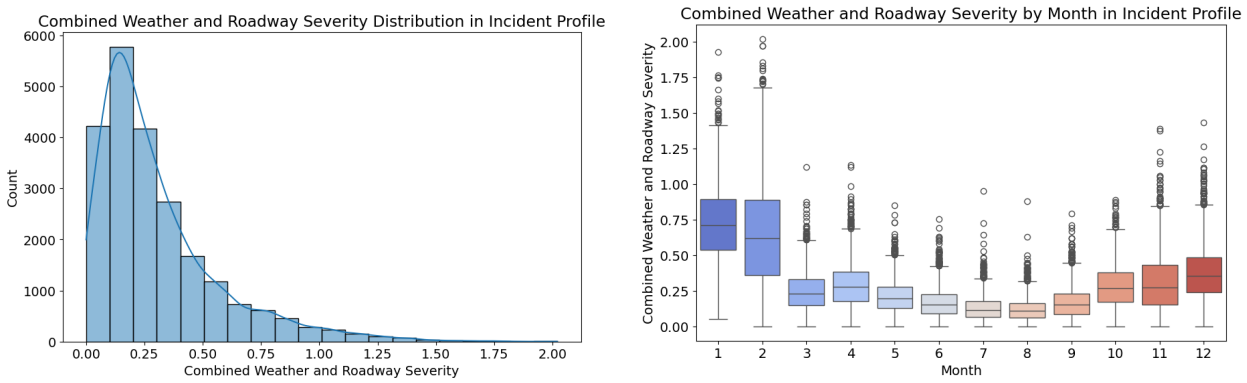

Figure 1.11: (i) Global and (ii) monthly distibution of Weather and Roadway severity combined

In Figure 1.12, we can observe the distributions for the `damage_cost_LB` and `num_towed_units_LB` features. Another interesting observation is that, as expected, the damage increases almost linearly with the number of towed units.

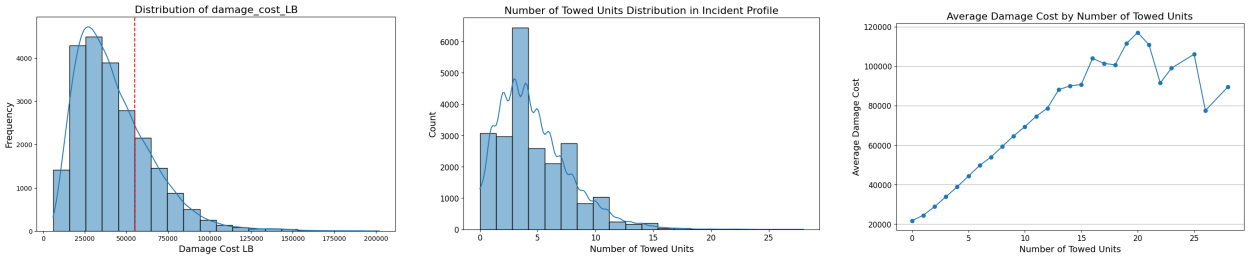

Figure 1.12: Distribution of (i) Damage, (ii) Tow and (iii) Average damage value with respect to the number of towed units.

2. Clustering

In this Chapter we explore different clustering algorithms: K-means, DBSCAN and K-means++. We focused on the capability of the algorithms to capture insight about the age and type of vehicle in each cluster as we thought that could be useful for an insurance company. We used both StandardScaler and the MinMaxScaler normalization techniques.

2.1 Choosing the Features

Starting with the dataset obtained during the first task, we selected features based on their correlation, ensuring that they were not highly correlated with one another to avoid redundancy. Specifically, we selected the following features:

damage_cost_LB
fatality_rate
speeding_influence
combined_weather_road_severity
avg_lighting_severity
avg_crash_severity_score

2.2 Outliers removal

We pre-processed our data by removing outliers from each selected feature. For features with a right-skewed distribution, we applied the IQR (Interquartile Range) method with $k \geq 2$ to eliminate the highest outliers. For other features with unusual distributions (though still right-skewed), we experimented with the Isolation Forest algorithm. Our incremental approach involved first identifying outliers for each feature, then merging the outlier sets, and finally dropping these outliers from the original dataset.

2.3 K-means

We used two methods to determine the optimal number of clusters, K, for the K-means algorithm. The first method involved using the elbow method and silhouette score, while the second method relied on a hierarchical clustering algorithm. In the hierarchical approach, we selected the number of clusters by choosing an appropriate `color_threshold` in the dendrogram, aiming for a compromise between the height of the intervals and the resulting number of clusters. This method has a second advantage since we can also find the K centroids that the K-means algorithm will use as a starting point.

For the first method, we chose $k = 6$, as this value corresponds to the elbow of the function in Figure 2.1 and it achieved the highest silhouette score. We also ran hierarchical clustering with Ward-linkage to extract centroids, but since the results were very similar, we decided to proceed with the centroids obtained from the standard K-means implementation.

For the second approach, we experimented with different linkage methods (complete-linkage, average-linkage, and Ward-linkage) and used Euclidean distance as the metric. With Ward-linkage and by fine-tuning the `color_threshold`, we determined that $K = 3$ was optimal, and we extracted the corresponding centroids to perform K-means clustering.

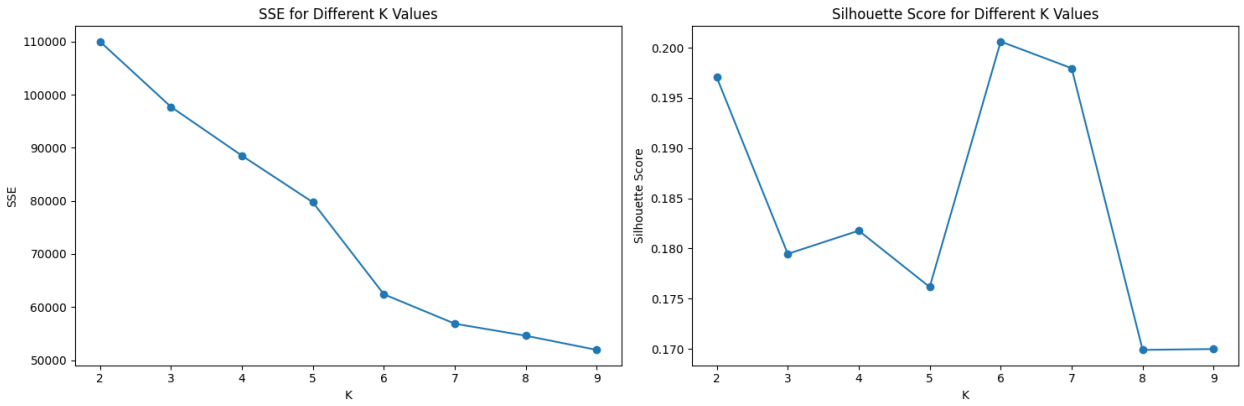

Figure 2.1: K-means: Elbow Method and Silhouette score to identify the best K.

2.3.1 Clusters Characterization

For both methods, we analyzed the distribution of features within each cluster using KDE plots and boxplots. Scatterplots were also used to examine pairwise feature relationships in order to understand where the centroids were located and what each cluster was capturing. Additionally, we used spider plots and line plots to further characterize the clusters, as shown in Figure 2.2. For example, Figure 2.3 shows the centroids of each cluster, revealing that Cluster 0 clearly captured crashes with a high fatality rate. Finally, we utilized boxplots and histograms to explore insights in external features, with a particular focus on the age of individuals involved in crashes and the type of vehicle they were driving. For the first method we found 6 clusters, specifically:

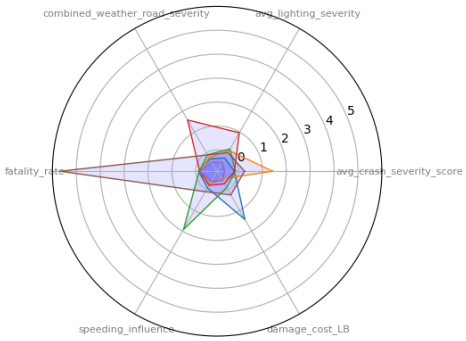

Figure 2.2: Spider plot of the clustering features.

- **Cluster 0: (4287 Elements)** This cluster represents crashes that are considered expensive. It has the highest `damage_cost_LB` out of any cluster, as well as the highest vehicle involvement rate (crashes with more vehicles involved). There is a high percentage of neo-patent drivers and seniors. It also has the highest percentage of truck/commercial vehicles.
- **Cluster 1: (3860 Elements)** This cluster has the highest `average_severity_score` and the lowest `total_crashes` in average. Thus, crashes in this cluster are more likely to be severe.
- **Cluster 2: (2517 Elements)** This cluster has the highest speeding influence, one of the highest rates of speeding as the main reason for crashes, a low fatality rate, and the highest percentage of towed units. It may represent crashes where drivers were speeding and crashed but were not injured.
- **Cluster 3: (3549 Elements)** This cluster may represent crashes caused by weather or road conditions. There are more crashes in winter months compared to other clusters. It has the highest `adverse_weather_crash_rate`, `avg_weather_severity`, and `combined_weather_road_severity`. The main reason for crashes in this cluster is considered to be speeding.
- **Cluster 4: (6916 Elements)** This cluster does not display any particularly distinguishing characteristics; it is average across almost all internal and external features.
- **Cluster 5: (561 Elements)** This cluster has the highest fatality rate and `injury_severity_score`, representing crashes likely to result in fatalities or injuries. Additionally, it has the highest percentage of motorcycles and the highest percentage of crashes due to drinking. A larger proportion of young individuals is also observed in this cluster.

For the second method, we identified 3 clusters with the following characteristics:

- **Cluster 0: (601 Elements)** This cluster may represent crashes that are likely to have a fatality or an injured person because it has the highest `injury_severity_score` and fatality rate. It has the highest percentage of crashes due to drinking, and there are more motorcycles.
- **Cluster 1: (13,527 Elements)** There are many elements in this cluster, and most of the features are in the average range.
- **Cluster 2: (7,562 Elements)** This cluster may represent crashes caused by weather or road conditions. There are more winter months compared to other clusters, and speeding is the main cause of crashes.

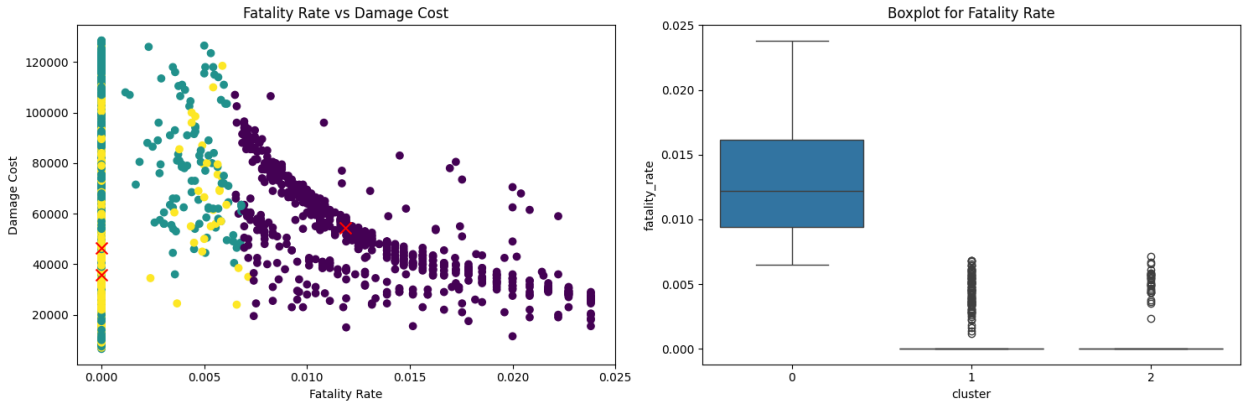

Figure 2.3: K-means: Scatterplot and boxplot to characterize the clusters.

2.3.2 Clusters Evaluation

We evaluated the cluster results using the Silhouette score, the Davies-Bouldin score (to assess separation), SSE and the Similarity Matrix. For the latter we used a subset of 5000 elements due to the size of the dataset. For the first method, with $k = 6$, we had a Silhouette score of 0.20, Separation of 1.26 and SSE of 62407. The correlation between the similarity Matrix (Figure 2.4) and the ideal Similarity matrix was 0.39. While for the second method, with $k = 3$, we had a better Silhouette Score of 0.214 but a worse Separation and SSE of 1.55 and 91632 respectively. The correlation using the similarity matrix was roughly the same as for $k = 6$.

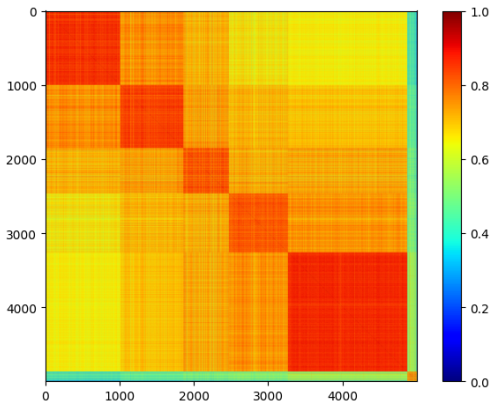

Figure 2.4: Similarity Matrix

2.4 DBSCAN

We used both the StandardScaler and the MinMaxScaler and obtained nearly identical results, so we will discuss only the outcomes with the StandardScaler. To find the best parameters of *eps* and *min_samples* we first plotted the k-distance plots for k ranging from 1 to 20 to identify the elbow for *eps*. We tried a couple of combinations without much success (Only 1 cluster beside the outliers), so we used a Grid-search to facilitate our research. We tried with *min_samples* from 5 to 15 and *eps* from 0.7 to 1.7. We ordered the results based on the Silhouette score and we picked $eps = 0.9$ and $min_samples = 9$, which produced 4 clusters. We tried with a higher number of $min_samples$ and we performed another Grid-search with with *min_samples* from 95 to 105 and *eps* from 1.3 to 2. We only found result with only 2 clusters inside. In Figure 2.5 the K-distance plots with K = 10 and K = 100.

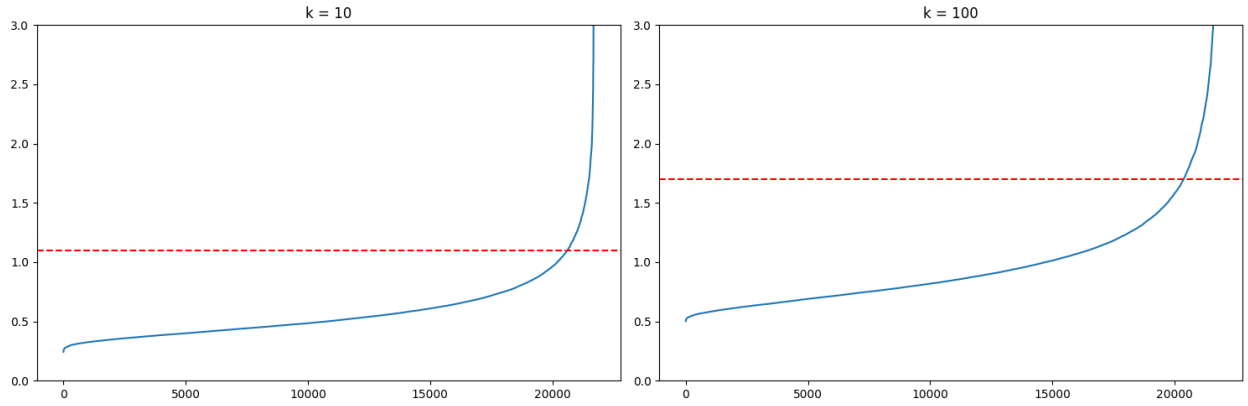

Figure 2.5: DBSCAN: K-distance plot with K = 10 and K = 100.

2.4.1 Clusters Characterization

We used the same plots discussed in the Subsection 2.3.1 and we characterized the obtained clusters as follows:

- **Cluster 0 (20,489 Elements)** This cluster contains almost all the elements and did not capture much specific information.
- **Cluster 1 (96 Elements)** This cluster has the highest average number of crashes, the highest damage, and more vehicles involved in crashes. It may represent crashes that are considered expensive.
- **Cluster 2 (19 Elements)** This cluster has more young drivers and the highest fatality rate. It likely represents crashes with one or more fatalities. Motorcycles are more frequent in this cluster compared to others.
- **Cluster 3 (15 Elements)** This cluster has fewer senior drivers and more young drivers compared to other clusters. It has the highest percentage of crashes where speeding is the main cause. This may indicate that young drivers tend to crash due to speeding.
- **Cluster -1 (1071 Elements)** The found outliers didn't seem to capture much.

2.5 K-means++

We used the implementation of K-means++ from the library *pyclustering*, We used the number of clusters found in Section 2.3, so $K = 6$. The K-means++ initialization did not seem to improve much, we had roughly the same Silhouette score and Separation as the normal K-means. The found clusters are of similar sizes, respectively 6921, 4285, 3857, 3547, 2519 and 561 elements. The analysis of the centroids is roughly the same as well. In Figure 2.6we plotted in the 2-PCA the clusters obtained with kmeans (left) and the ones with kmeans++ (right) and they are very similar.

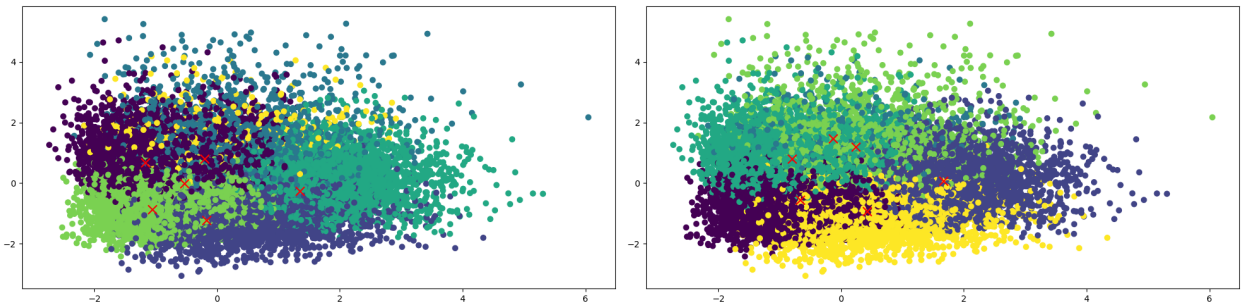

Figure 2.6: On the left the clusters found with K-Means and on the right the clusters with K-means++ plotted in the 2-PCA.

2.6 Final Evaluation

The DBSCAN algorithm performed the worst despite achieving a Silhouette score of 0.39. It identified three small clusters while assigning the majority of data points to one large cluster. In contrast, both K-means and K-means++ performed well, finding interesting and meaningful clusters. Based on these results, we conclude that either K-means or K-means++ is the preferred choice for our clustering task.

3. Predictive Analysis

For the predictive analysis, we considered the problem of predicting for each monthly profile of the department the `DAMAGE` of the incidents.

3.1 Data Preparation

To define our predictive task, we first aggregated the `DAMAGE` crash feature as the sum of each month, each department. Next, we aimed to classify whether a given incident profile should be labeled as `DMG_HIGH`, based on whether it exceeded the 25^{th} percentile of the overall `DAMAGE` distribution across all years (see Figure 3.1(i)).

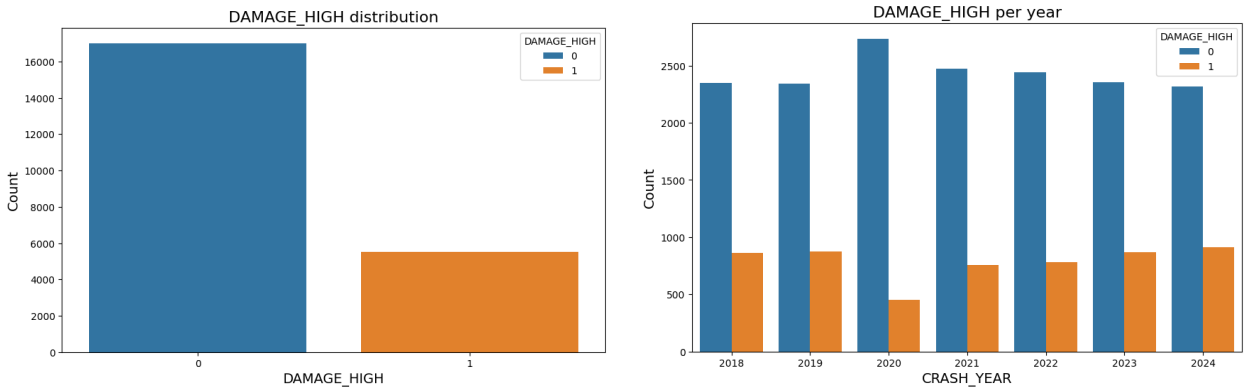

Figure 3.1: Distribution of the target variable: (i) in the whole dataset and (ii) across all years.

The final set of features used in our predictive analysis was as follows:

```
['CRASH_YEAR', 'CRASH_MONTH', 'avg_responsibility_score',
 'avg_lighting_severity', 'total_crashes', 'fatality_rate',
 'severe_injury_rate', 'avg_crash_severity_score',
 'injury_severity_score', 'adverse_weather_crash_rate', 'road_defect_crash_rate',
 'speeding_influence', 'weekend_crash_rate', 'neo_patented_drivers',
 'senior_drivers', 'num_towed_units_LB', 'avg_age',
 'pct_neo_patented_drivers', 'pct_senior_drivers', 'monthly_total_units',
 'vehicle_involvement_rate', 'combined_weather_road_severity',
 'towed_unit_rate', 'DAMAGE_HIGH']
```

These features encapsulate various aspects of crash incidents—ranging from temporal information to severity indices and demographic factors—and were used to build and evaluate our predictive model.

3.1.1 Train - Test Split

We performed the train–test split by using all entries from 2024 as the test set. This resulted in an approximate 83–17 split between training and test data, closely matching our target ratio of 80–20.

3.2 Validation

To validate our models and identify the optimal hyperparameters, we primarily used K-fold cross-validation. For certain models, such as K-Nearest Neighbors (K-NN), we directly evaluated different values of K on the test set. Although this approach can lead to potential overfitting in practice, it was used here for didactic purposes. Additionally, to conduct a more detailed analysis of the Neural Network's performance, we extracted a validation subset from the training data. This allowed us to observe and visualize overfitting by examining the training curves. The best set of hyperparameters for each model can be found in the appendix.

3.3 Models Used

For the classification problem, we experimented with several machine learning models, varying from both traditional algorithms to ensemble methods. The selected models include Decision Tree, Naive Bayes, K-Nearest Neighbors (K-NN), XGBoost, AdaBoost, Random Forest, and Neural Networks (NN).

Model Performance Comparison Using All Features				
Model	Accuracy	Precision (0 / 1)	Recall (0 / 1)	F1-Score (0 / 1)
Decision Tree	0.98	0.98 / 0.96	0.99 / 0.95	0.98 / 0.96
Random Forest	0.97	0.97 / 0.98	0.99 / 0.91	0.98 / 0.94
K-Nearest Neighbors	0.96	0.95 / 0.99	0.99 / 0.87	0.97 / 0.92

Table 3.1: Performance comparison of classification models with class-specific metrics represented as 0 / 1. The dataset used was contained all the available features.

3.3.1 All feature analysis

The initial results, showed in table 3.1 were obtained using all the features listed in Section 3.1.

Due to the extremely high performance of the models, we suspected that one or more features highly correlated with the target variable were driving these positive results. To investigate this further, we explored feature importance using SHAP values and subsequently performed an ablation study, iteratively removing features to analyze their impact on model performance. For instance, as shown in Figure 3.2, the decision was largely influenced by the `total_crashes` feature, which is intuitive since departments with a higher number of crashes tend to have higher costs for that month.

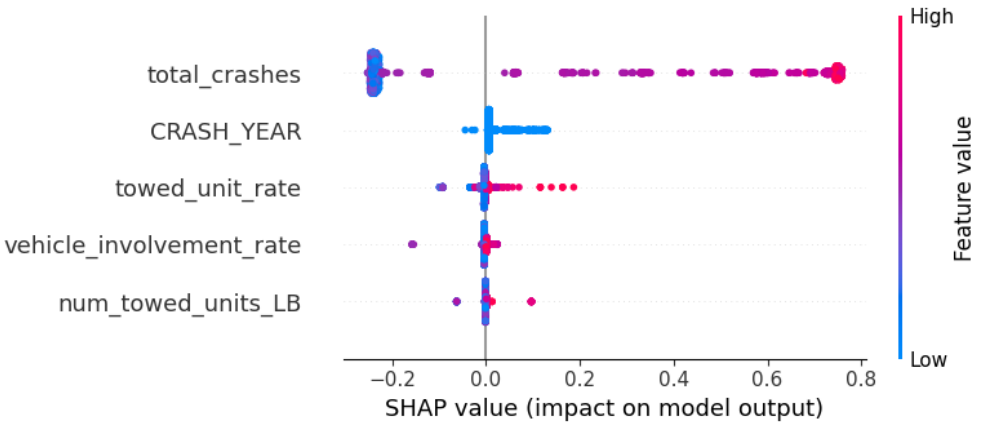

Figure 3.2: Global impact on prediction of the feature of the Decision Tree model.

We then removed the obviously biased features, `total_crashes` and `vehicle_involvement_rate`, and repeated the analysis. As shown in Table 3.2, the results remain extremely high.

Model Performance Comparison Using All Features				
Model	Accuracy	Precision (0 / 1)	Recall (0 / 1)	F1-Score (0 / 1)
Decision Tree	0.94	0.94 / 0.95	0.98 / 0.83	0.96 / 0.89
Random Forest	0.93	0.92 / 0.97	0.99 / 0.78	0.95 / 0.86
K-Nearest Neighbors	0.85	0.86 / 0.81	0.94 / 0.62	0.90 / 0.70

Table 3.2: Same analysis with the removal of the most impactful, and highly correlated features.

Figure 3.3 shows that, after removing the initial biased features, the most impactful features now capture

the magnitude of the number of crashes and, consequently, the expected damage cost. Based on this observation, we decided to drop all features that represent a flat number of instances (`neo_patented_drivers`, `senior_drivers`, and `num_towed_units_LB`) and proceeded with the final analysis of all the models.

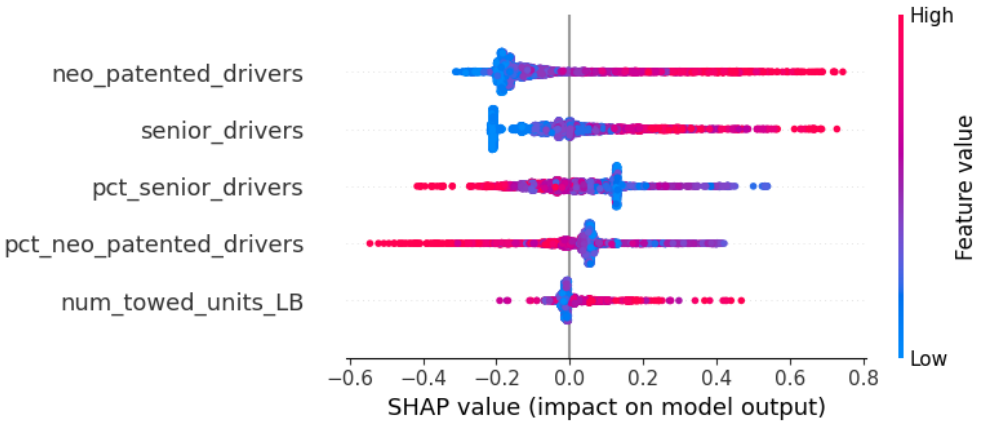

Figure 3.3: Global impact on prediction of the feature of the Decision Tree model.

3.3.2 Discarding Biased Features

For the final dataset—comprising 17 of the original 22 features—we evaluated model performance, as shown in Table 3.3. These results shows how the predictive task remains reasonable and that the models continue to perform very well even when features that are highly correlated with the target variable, or even simply with the magnitune of the number of crashes, have been removed.

Model Performance Comparison Without New Features				
Model	Accuracy	Precision (0 / 1)	Recall (0 / 1)	F1-Score (0 / 1)
Decision Tree	0.85	0.86 / 0.79	0.94 / 0.63	0.90 / 0.70
Naive Bayes	0.75	0.83 / 0.56	0.82 / 0.59	0.83 / 0.57
K-Nearest Neighbors	0.72	0.77 / 0.52	0.87 / 0.34	0.82 / 0.41
XGBoost	0.90	0.90 / 0.93	0.98 / 0.71	0.93 / 0.80
AdaBoost	0.87	0.85 / 0.92	0.98 / 0.57	0.91 / 0.70
Random Forest	0.88	0.87 / 0.93	0.98 / 0.62	0.92 / 0.74
Neural Network	0.86	0.92 / 0.73	0.88 / 0.81	0.90 / 0.77

Table 3.3: Final analysis on all the models proposed.

While the performance of other models was not significantly affected by the magnitude and distribution of the data, the Neural Network showed sub-optimal results on the original data. After scaling the data to have a mean of 0 and a variance of 1, we obtained the improved results shown in the table. In Figure 3.4, the clear difference in training and validation accuracy between the original and scaled data is evident.

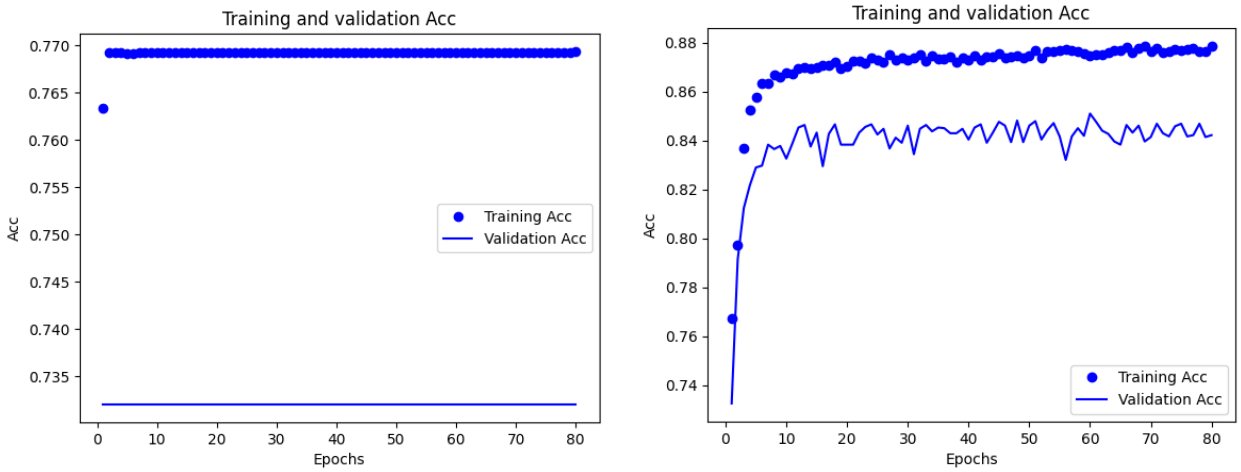

Figure 3.4: Training and validation accuracy (i) with the original (ii) and with the scaled data.

To gain insight into the model predictions, we analyzed the SHAP values for the models. In Figure 3.5, the SHAP value distributions for both the Decision Tree and the best-performing model, XGBoost, are presented. Notably, both models prioritized the same features, which capture different degrees of average crash severity for the whole month.

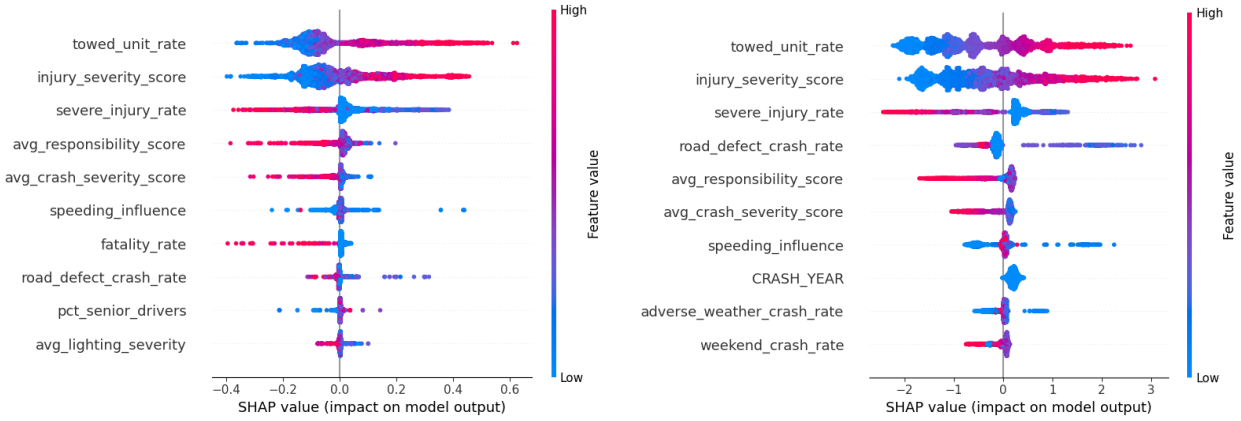

Figure 3.5: SHAP value for (i) Decision Tree and for (ii) XGBoost, the best performing model.

In Figure 3.6, we can see the confusion matrix for XGBoost, allowing for a comparison with the results in Table 3.3, alongside the ROC curve for all models. Notably, from the confusion matrix we can see how predicting the label 1 (high damage) proved to be more challenging, likely due to dataset imbalance.

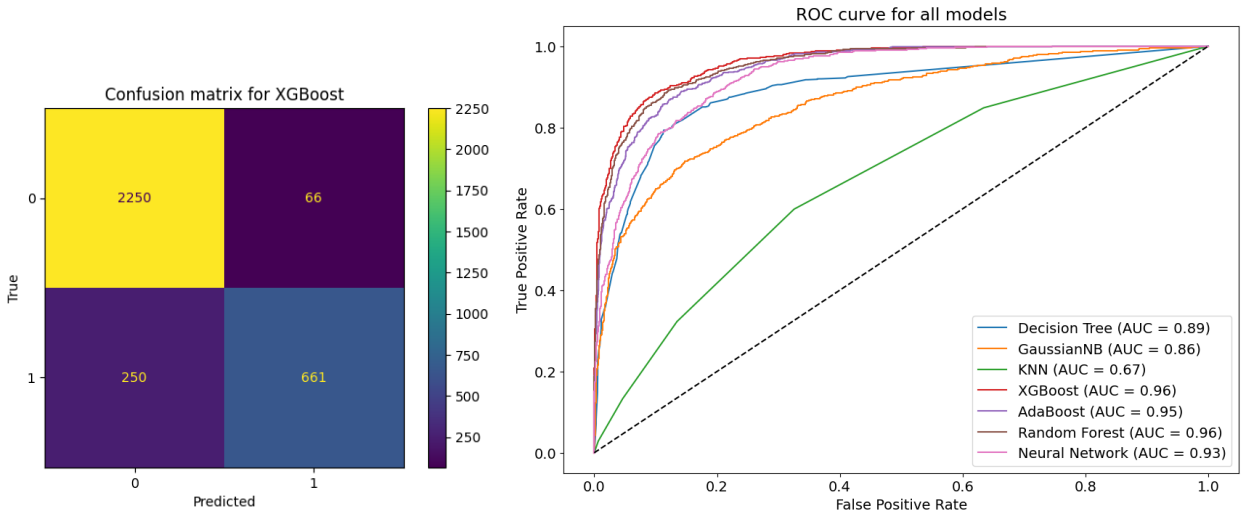

Figure 3.6: (i) Confusion Matrix for XGBoost and (ii) the ROC curve for all the models.

3.3.3 Addressing The Imbalance

Figure 3.1 shows the imbalance in the feature distribution resulting from our definition of a high-damage profile. As shown in Table 3.3, predicting the minority class (high damage) proved to be harder. Consequently, we conducted an experiment using a balanced version of the dataset, reducing the total number of training entries from 19,289 to 9,190. As shown in Table 3.4, the performance of the models deteriorated, likely due to the reduced volume of training data.

Model Performance Comparison Using All Features				
Model	Accuracy	Precision (0 / 1)	Recall (0 / 1)	F1-Score (0 / 1)
Decision Tree	0.80	0.88 / 0.63	0.84 / 0.71	0.86 / 0.67
Random Forest	0.82	0.88 / 0.68	0.87 / 0.71	0.87 / 0.69
XGBoost	0.78	0.85 / 0.59	0.83 / 0.64	0.84 / 0.62

Table 3.4: Analysis with the balanced dataset by undersampling the majority class

3.3.4 Rule Based Classification with Ripper

For a final experiment,we decided to explore a rule-based classification approach using the RIPPER algorithm. Despite using only 1000 balanced training samples, the model produced good results.

The following output shows an example of the classification outcomes. The first element of the tuple indicates the correctness of the predictions, while the second element lists the extracted rules for each class:

```
([True, False, True, True, True],
 [[<Rule [towed_unit_rate=0.00054-0.00065]>],
  [],
  [<Rule [road_defect_crash_rate=0.018-0.022]>,
   <Rule [severe_injury_rate=0.0093-0.011]>],
  [<Rule [injury_severity_score=14.0-18.0^severe_injury_rate=0.011-0.016]>,
   <Rule [speeding_influence=0.016-0.028]>],
  [<Rule [injury_severity_score=14.0-18.0^severe_injury_rate=0.011-0.016]>])
```

3.4 Conclusions

Through SHAP value analysis and an ablation study, we identified and removed redundant or biased features, such as `total_crashes` and `vehicle_involvement_rate`. Despite these removals, the predictive models continued to perform well. We also observed how the Neural Network benefited significantly from data scaling, achieving improved performance when the data were normalized.

Further experimentation with a balanced version of the dataset—where the number of training samples was reduced to address class imbalance—revealed a deterioration in performance, likely due to the decreased training data volume.

A. Hyperparameters of the Models

The following table summarizes the best hyperparameters found through model validation. The experimental results obtained are illustrated in Table 3.3.

Model	**Parameters**
Decision Tree	min_samples_split: 10, min_samples_leaf: 4, max_depth: 9, criterion: 'entropy'
Knn	K: 5
XGBoost	n_estimators: 100, max_depth: 4, learning_rate: 0.1
Random Forest	n_estimators: 64, min_samples_split: 16,min_samples_leaf: 2, max_depth: 11 ,criterion: 'gini',
AdaBoost	n_estimators: 250, learning_rate: 0.1
Neural Network	optimizer: 'adam', first layer of 256, second layer of 64
RIPPER	prune_size: 0.5, k: 1

Table A.1: Hyperparameters for the models used in the final experiments.

Bibliography

[1] City of Chicago. *Traffic Crashes (Crashes) [Data set]*. `https://data.cityofchicago.org/Transportation/Traffic-Crashes-Crashes/85ca-t3if/about_data`. Accessed: 2025-02-15. 2024.

[2] Plotly Technologies Inc. *Collaborative data science*. 2015. URL: `https://plot.ly`.

[3] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Boston, MA: Addison-Wesley, 2005. ISBN: 9780321321367.