UNIVERSITY OF PISA
DEPARTMENT OF COMPUTER SCIENCE
Master's degree in Computer Science

Data Mining Project
Gun incidents in the USA

Prof.
Monreale Anna
Mannocci Lorenzo

Candidates:
Gaetano Nicassio (658073)
Niko Paterniti (638257)
Roberto Claudio Moramarco (666140)

ACADEMIC YEAR 2023/2024

CONTENTS

1. Introduction

The pervasive and complex issue of gun incidents in the United States demands a comprehensive understanding to inform effective policy-making and public safety measures. As the nation grapples with the multifaceted challenges surrounding firearms, the application of advanced data mining techniques emerges as a powerful tool to unravel hidden patterns and insights within the vast datasets related to gun incidents.
The aim of this project is a data-driven exploration, seeking to harness the potential of data mining methodologies to discern trends, identify risk factors, and ultimately contribute to evidence-based decision-making in addressing the issue of gun violence in the USA. Through the lens of data mining, we aim to uncover hidden correlations and develop a deeper understanding of the dynamics surrounding firearm-related events in the USA.

2. Data understanding and preparation

The **data understanding** phase aims to unify the dataset in order to improve the result of subsequent operations. In particular, we focused on data analysis in order to eliminate duplicate values, handle missing or corrupted data, and identify outliers.

2.1. Data Overview

Before manipulating the data, the first task performed is to understand the semantics of each attribute we need to work with. For each attribute we also differentiate between the original data type and the cast data type that we transformed to better usage in the data cleaning phase.
The incidents dataset contains 239677 total observations, in the following plot we show a summary of the NaN values for each attribute in the dataset.

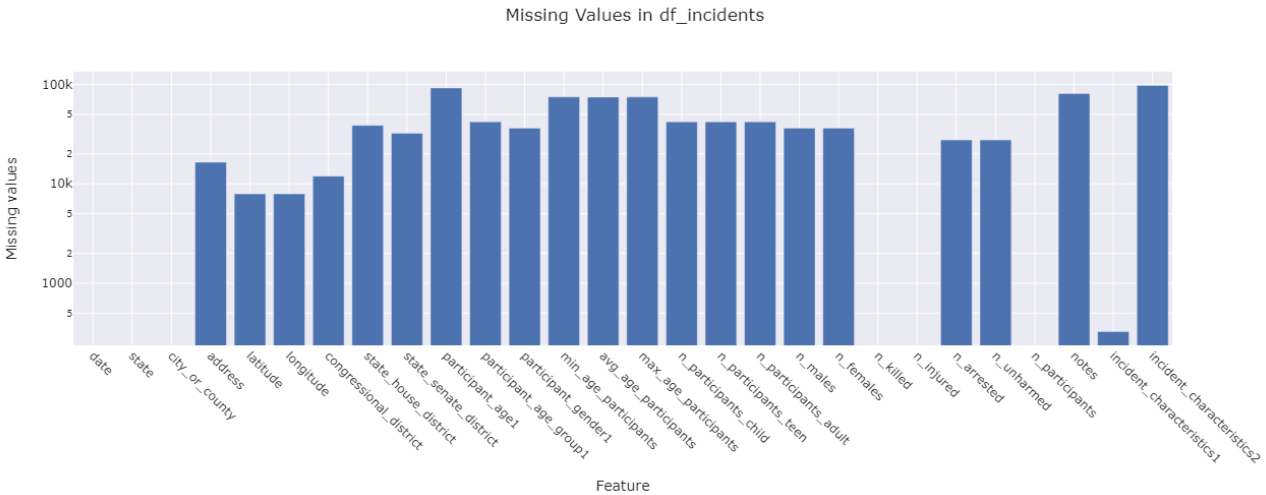


Figure 1 - NaN values distribution

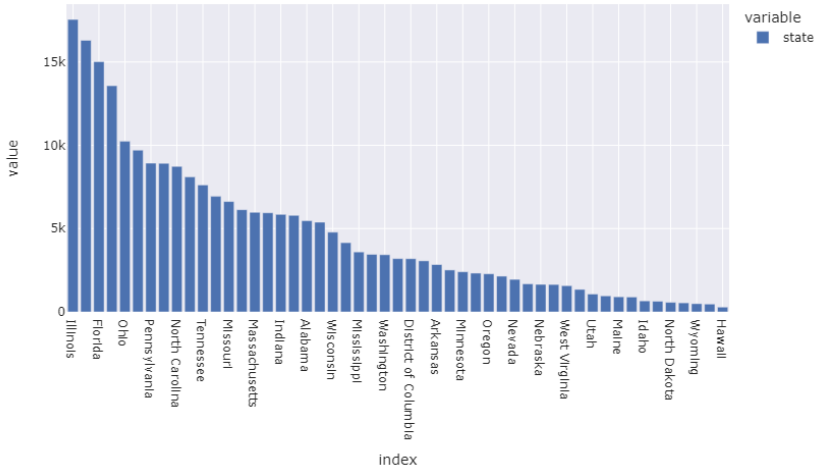
2.2. Features understanding

Here we describe the semantic analysis for every attribute in each dataset that we have.

The **incidents.csv,** contains information about gun incidents in the USA. In the dataset there are the following variables:

1. *date*: date of incident occurrence (ranging from 2013/ 01/01 to 2030 /11/ 28, since 2030 and other dates appearing in the dataset are not possible, they've been handled in the cleaning phase)

2. *state*: state where incident took place (Top 3 states with the most incidents are: ILLINOIS with 17556 incidents, CALIFORNIA with 16306 incidents and FLORIDA with 15029 incidents)



3. *city_or_county*: city or county where incident took place (Chicago is the city with the most incidents)
4. *address*: address where incident took place
5. *latitude*: latitude of the incident
6. *longitude*: longitude of the incident
7. *congressional_district*: congressional district where the incident took place (we notice that Chicago's 7th district is the one with the most incidents)
8. *state_house_district*: state house district
9. *state_senate_district*: state senate district where the incident took place.
10. *participant_age1*: exact age of one (randomly chosen) participant in the incident (the ages ranging from 18 to 30 are the one most involved in incidents)
11. *participant_age_group1*: exact age group of one (randomly chosen) participant in the incident
12. *participant_gender1*: exact gender of one (randomly chosen) participant in the incident
13. *min_age_participants*: minimum age of the participants in the incident
14. *avg_age_participants*: average age of the participants in the incident
15. *max_age_participants*: maximum age of the participants in the incident
16. *n_participants_child*: number of child participants 0-11
17. *n_participants_te*en: number of teen participants 12-17
18. *n_participants_adult*: number of adult participants (18 +)
19. *n_males*: number of males participants
20. *n_females*: number of females participants
21. *n_killed*: number of people killed
22. *n_injured*: number of people injured
23. *n_arrested*: number of arrested participants
24. *n_unharmed*: number of unharmed participants
25. *n_participants*: number of participants in the incident
26. *notes*: additional notes about the incident
27. *incident_characteristics1*: incident characteristics (52 possible values, shooting incidents are the most common)
28. *incident_characteristics2*: additional incident characteristics (90 possible values, officer involved incidents are the most common)

For the sake of readability, the tables relating to what was previously described have not been reported here. For the complete result of the data understanding the reader can consult the notebook regarding this part.

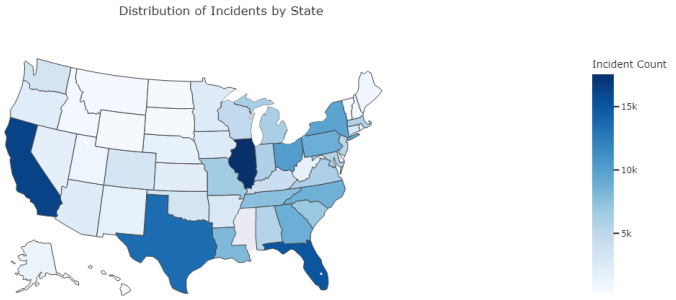


Figure 2 – Incident distribution by states

The **povertyByStateYear.csv** dataset contains information about the poverty percentage for each USA state and year, so it includes the following variables:

1. *state*
2. *year*
3. *povertyPercentage*: poverty percentage for the corresponding state and year

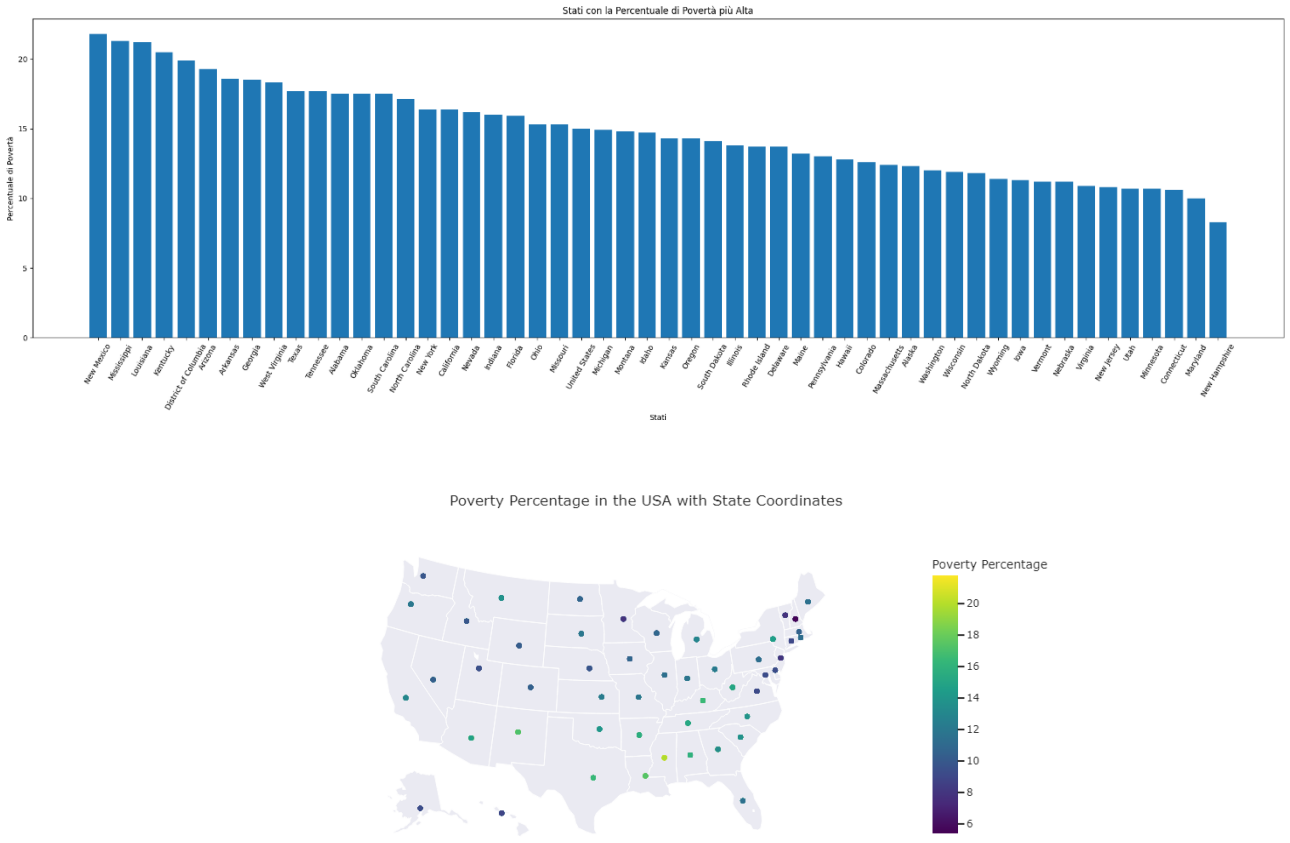


Figure 3 - Poverty percentage distribution per State in different styles of presentation

The **year_state_district_house.csv** dataset contains information about the winner of the congressional elections in the USA, for each year, state and congressional district. It includes the following variables:

1. *year*
2. *state*
3. *congressional_district*
4. *party*: winning party fort the corresponding congressional_district in the state, in the corresponding year ('REPUBLICAN' ,'DEMOCRAT', 'FOGLIETTA (DEMOCRAT)' , 'DEMOCRATIC-FARMER-LABOR', 'INDEPENDENT' , 'INDEPENDENT REPUBLICAN'
5. *candidateVotes*: number of votes obtained by the winning party in the corresponding election 6. totalVotes: number total votes for the corresponding election

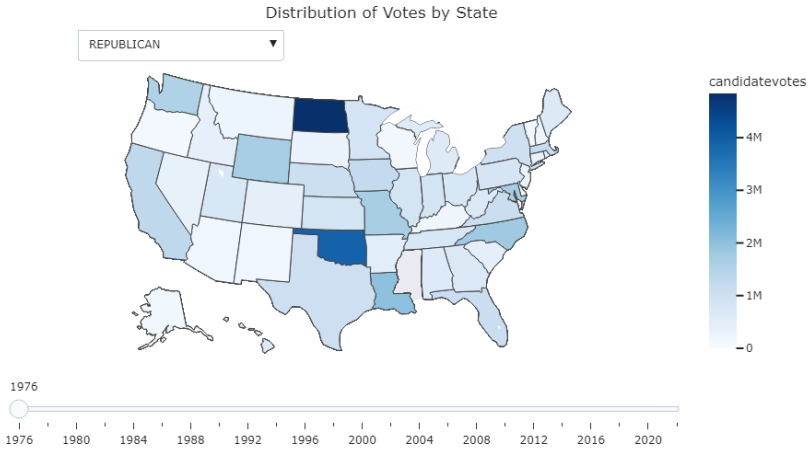


Figure 4 - Total votes distribution per state

2.3. Data Cleaning

We dropped the following columns since we decided they were not significant to our purposes: '*address'*, '*latitude'*, '*longitude'*, '*state_house_district'*, '*state_sentate_district'*, '*participant-age1'*, '*participant_age_group1*', '*notes'*.

The dataset presented anomalous dates (2030,2029,2028), that we decide to transform in 2020,2019,2018 respectively assuming the error was due to misspelling.
Concerning the ages present in the dataset (*min_age_participants*, *avg_age_participants*, *max_age_participants*) we checked for anomalous values, in which case if a value > 100 appeared the value has been set to NaN.

For all the features that contains a "number of events", like *n_participants_child*, *n_participants_teen*, *n_participants_adult*, *n_males*, *n_females*, *n_killed*, *n_injured*, *n_arrested*, *n_unharmed*, we checked for anomalous values in these columns and if in one of the columns appeared a value greater than *n_participants* we set the value to NaN.

All the anomalous values in each column like alphanumeric values have been set to NaN.

Since the age values and the number of involved are by nature integer numbers, we transformed their type to Int64 to be also able to represent and later handle NaN values.

We dropped all the rows where every significant features useful to understand the incident presented Nan values (we found 49 rows where all the following columns presented NaN values (*min_age_participants, avg_age_participants, max_age_participant, n_participants_child, n_participants_teen, n_participants_adult, n_males, n_females, n_killed, n_injured, n_arrested, n_unharmed, n_participants, incident_charaterisctics1*).

We then performed some data driven analysis in order to fill the appearing NaN values and clean the dataset.
Initially we analyzed the distribution for the features *min/avg/max_age_participants*

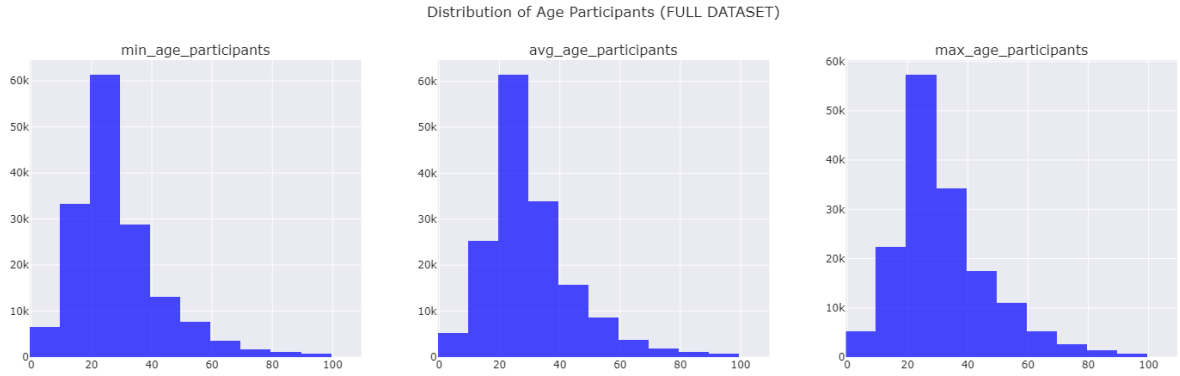


Figure 5 - Age distribution

But being this distribution analysis too general and due to the fact that the distribution are probably different w.r.t the year or the state we are considering, we decided to further analyze the distribution using the mean for each year.
After analyzing the plots, we concluded that the median would have probably been a better metric for our purposes since it is less sensitive to outliers.

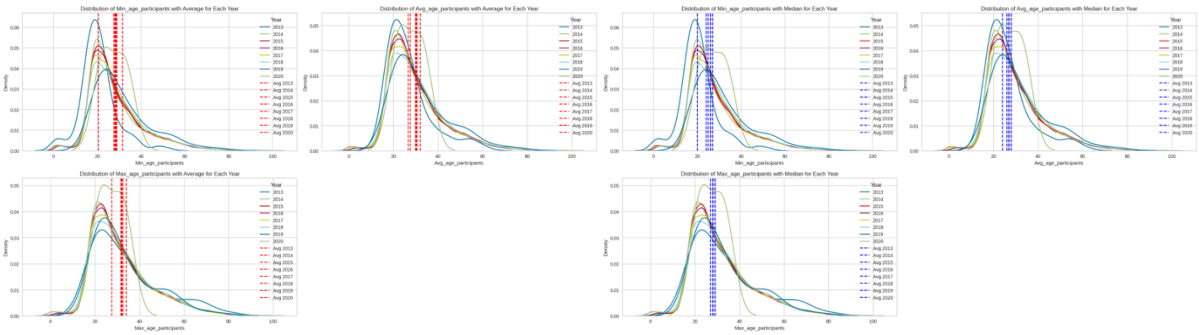
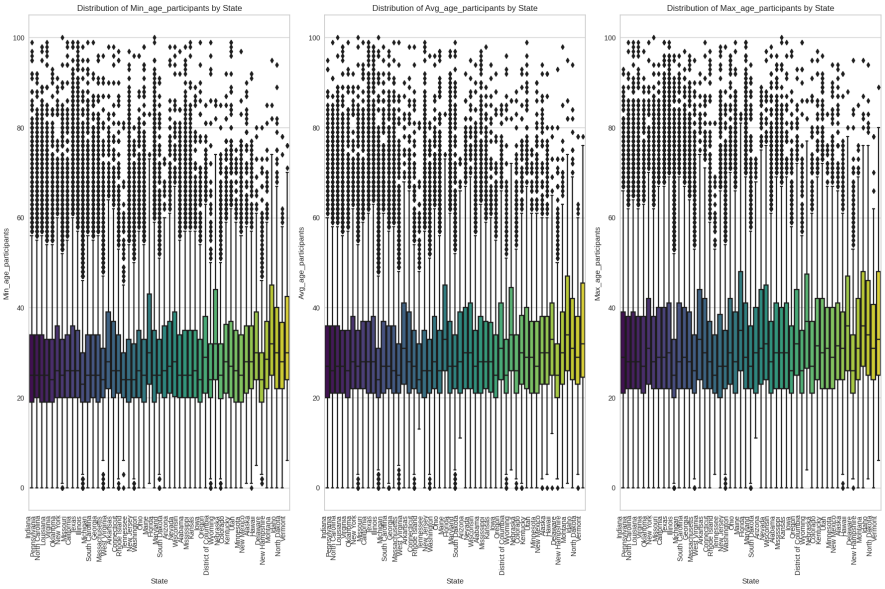


Figure 6 - Average and median over the age distribution

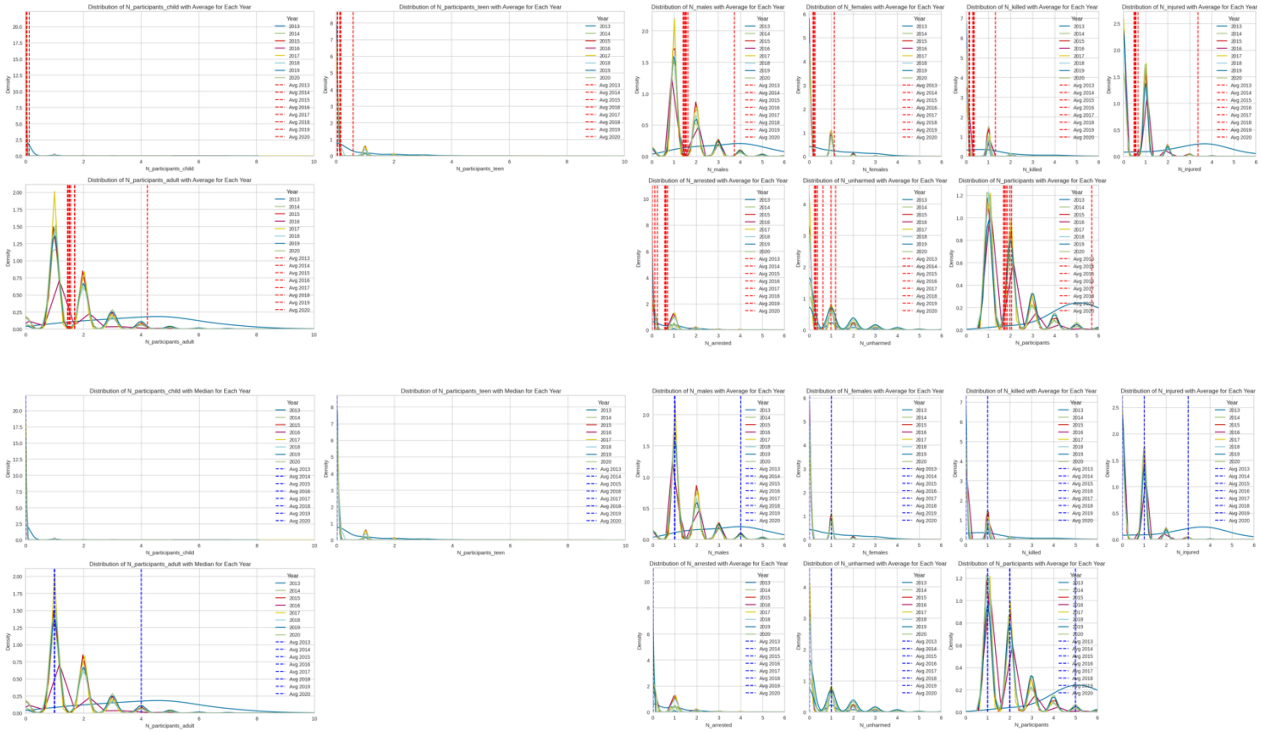
We then proceeded into analyzing the distributions considering each state since each state might have a different distribution, and the participants for some states might be generally older giving us useful insights about which states present generally younger or older participants. Also, in this case we analyzed both the mean and the median, the latter being the preferred choice for the same reason stated previously.



After having analyzed the distributions with the mean and median values for each year and for each state we came to the conclusion that both information are crucial because the distribution of some states differs based on the year we are considering and vice versa, so we decided to fill the NaN values as follows:

- Fill with the median calculated for each state and each year in order to be as accurate as possible.
- The choice of the median instead of the mean as previously stated comes from the consideration of the presence of outliers and being the median less sensitive to them it is a better choice.

We performed the same analysis for all the other columns as shown in the plots below:



For all the columns we decided to use the median calculated for each state in each year.
For all the 3 dataset we decided to transform the state in uppercase for coherency between them and to later merge them.
About the elections dataset we decided to drop all the dates not in the range [2013,2020] because the incidents information range between these years, so values outside of the range are not significant to our analysis.
Since the elections are held in even years, we don't have information about odd years, so we extended the dataset with the repeated information of the previous even year for each state and each congressional district.
The dataset presented some NaN values in the votes columns, we researched official information regarding the elections and we were able to retrieve some records for the missing data, we also found out that the rest of the missing records presented NaN values because the official information had no votes recorded due to the winner party uncontested win (more information are available here: https://en.wikipedia.org/wiki/2018_United_States_House_of_Representatives_elections_in_Florida#District_10)

We then merged the 3 datasets trying to retrieve the missing values in the '*congressional_district'* feature of the incidents dataset from the official information present in the elections dataset. We were able to retrieve some of them, the rest of the NaN values were dropped.

2.4. Data Preparation: Feature extraction

In this chapter we will describe a new dataset obtained from the definition of interesting new features useful for describing the incidents.

2.4.1. Sex Participation rate

Provide information about how many males and females are involved in the incident w.r.t the total number of males involved in incidents for the same city and the same period. The new feature added are *male_participation_rate* and *female_participation_rate*

2.4.2. Involved Participation rate

Provide information about how many killed and injured people have been involved w.r.t the total number of killed and injured people in the same congressional district and in the same period of time. The new feature added are *killed_participation_rate* and *injured_participation_rate*

2.4.3. Killed rate

Ratio of the number of killed people in the incident w.r.t the number of participants in the incident (it is an indicator of the killing factor in the incidents that occur. The new feature added is *killed_ rate*.

2.4.4. Unharmed rate

Ratio of unharmed people in the incident w.r.t the average of unharmed people involved in incidents for the same period. The new feature added is *unharmed_ rate*.

2.4.5. Adult/Teen/Child Participation rate

Provides information about how many adult/teen/child participants are involved w.r.t the total number of adults, teens, childs involved in incidents for the same city and in the same period. The new features added are *adult_participation_rate, teen_participation_rate, child_participation_rate*.

2.4.6. Arrested rate

Ratio of arrested people w.r.t the total number of arrested people for the same city and the same period, it provides informations about the activity of law enforcement operations). The new feature added is *arrested_participation_rate*

2.4.7. Total arrested rate

Ratio of the number of arrested people w.r.t the number of participants in the incident. The new feature is *arrested/participants_rate.*

2.4.8. Votes percentage

Percentage of votes obtained by the winner party candidate, helpful to see if the winner party had a crushing victory in the election. The new feature added is *winner_party_votes_percentage.*

2.5. Data Correlation

Before proceeding into the clustering analysis, we analyzed the correlation matrix of the final dataset. The following images represent the correlation matrix correlation of the entire dataset (on the left) and the correlation matrix of the features extracted (on the right)

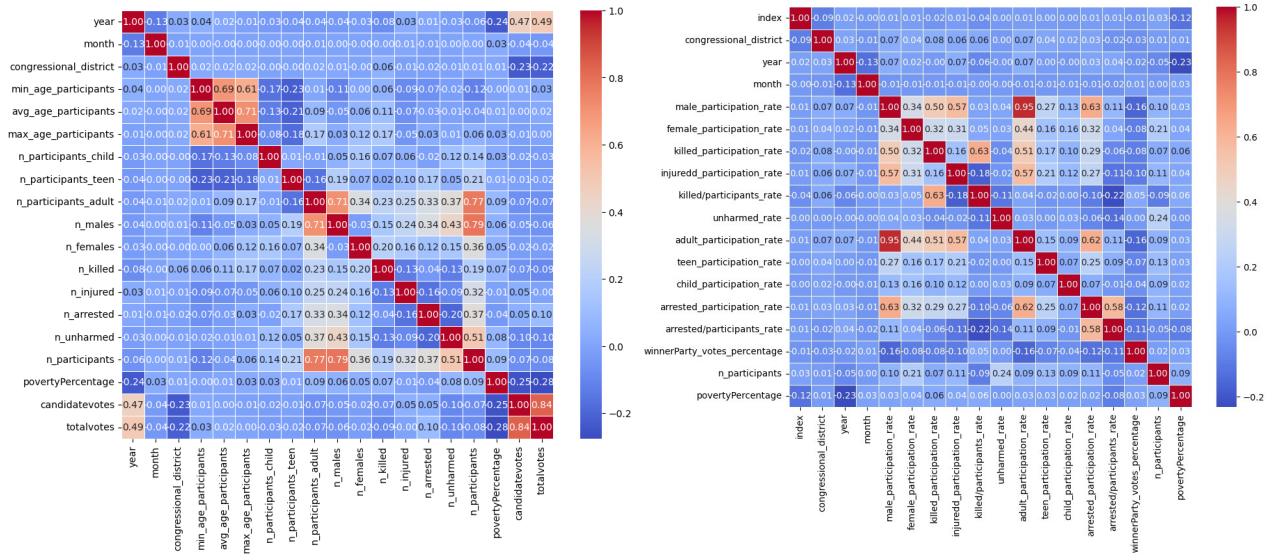


Figure 7 - Matrices correlation

3. Clustering analysis

In the Figure 7 - Matrices correlation is shown the Correlation Matrix of the dataset including the new features that we added.
And then before applying any clustering algorithm we need to choose the features, taken from a threshold of 0.6 that we think fit the best

- *male_participation_rate*: The rate of male participation might indicate certain patterns or trends.
- *female_participation_rate*: Similar to the male participation rate, you might want to include the female participation rate.

- *killed_participation_rate*: The rate of fatalities could be an important factor in understanding the severity of incidents.
- *injured_participation_rate*: Similar to fatalities, the rate of injuries could be crucial.
- *arrested_participation_rate*: The rate of arrests might indicate law enforcement activity and could be an interesting feature.
- *adult_participation_rate*, *teen_participation_rate*, *child_participation_rate*: Breakdowns by age groups might reveal different patterns.

For the feature selection, we choose these 4 that better describe the behavior of the incidents:

- *killed_participation_rate*
- *injured_participation_rate*
- *arrested_participation_rate*
- *povertyPercentage*

These features were chosen based on their relevance in capturing different aspects of incidents.

3.1. Preprocessing

In our analysis, after experimentation, we observed that the **K-Means** clustering algorithm yielded better results when applied to data normalized using Standard-Scaler. This adjustment in the normalization technique aims to enhance the algorithm's performance by standardizing the features, contributing to a more robust clustering analysis.

3.2. K-Means

The choice of the parameter "K" in the K-Means Algorithm approach is crucial given that it identifies the number of the clusters from the algorithm execution. In fact, before choosing the best "K" we let the Algorithm work for a given interval of values and in different ranges. As described after we identify the number of "K" equal to 4, according to the **Elbow Rule.**

3.2.1. The best K

In the application of the K-Means clustering algorithm, we employed several evaluation metrics to assess the quality of the clustering results. These metrics serve as quantitative measures, providing insights into the effectiveness and coherence of the clusters formed.

1. *SSE* is a fundamental metric in clustering analysis, representing the sum of the squared distances between each data point and the center of its assigned cluster. A lower SSE indicates tighter and more compact clusters.
2. *Silhouette Score* quantifies how similar each data point within a cluster is to the other points in the same cluster compared to those in neighboring clusters. The more cohesion is given by a range from -1 +1
3. *Davies-Bouldin Score* evaluates both the cohesion within clusters and the separation between clusters. A lower score indicate more cohesion and distinct clusters.

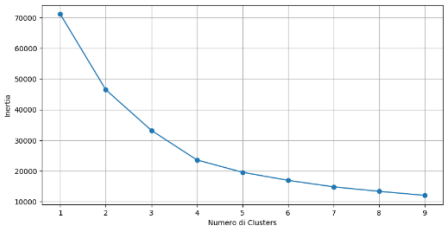


Figure 8 - Elbow Plot to see the best number of K

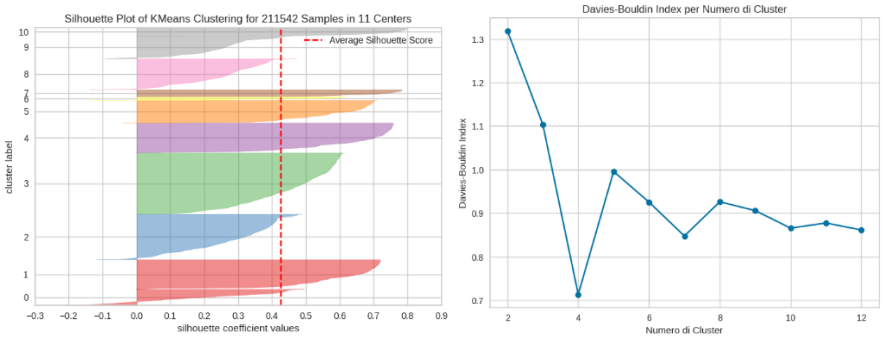


Figure 9 - Silohuette score and Davies Bouldin Score

Among the various considerations and analyzes made where we compared the different features. In the present scatter plot, we examined the distribution of the data with respect to the *"killed_participation_rate"* and *"arrested_participation_rate"* features, using the K-Means method to divide the data into four distinct clusters.
What immediately emerges is the notable distinction between the clusters and the poor similarity between the two features considered. It is evident, from the clear separation between the data points, that there is a marked distinction between the clusters generated by K-Means. The data appears to aggregate into well-defined regions of the two-dimensional feature space. Instead there is an overlap between the clusters could indicate a similarity in the distribution of the data for the two features considered. In particular, it appears that some data points belonging to different clusters have similar values for both *"povertyPercentage"* and *"arrested_participation_rate"*.

Other considerations are written on the notebook with all combinations of the features selected.

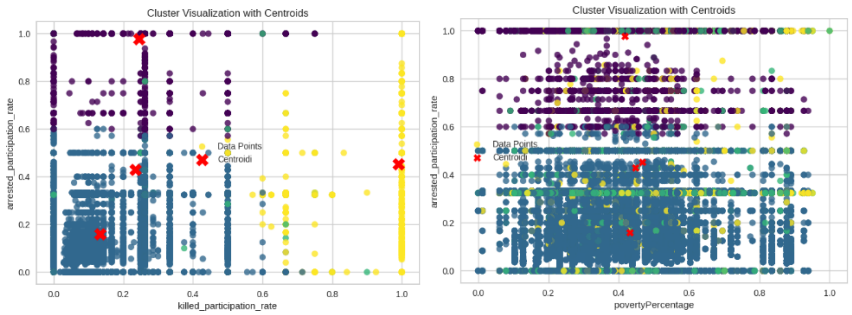


Figure – Different combinations of the features selected

The fact that emerges from the K-Means is that it converges quickly. The cons is that it requires less time and gives us the possibility to be applied several times and choose the best solution from the iterations taken.

3.3. DBSCAN

We selected the state of ILLINOIS because in the Data Understanding phase we noticed a high number of incidents, therefore in the DBSCAN we could have large dimensional results on the clusters. We still decide to consider the elected features taken before.

3.3.1. Eps and Min_samples

For selecting the best values for eps and *min_samples* we apply a grid search. The results are written on the notebook. The use of *eps*=0.2 and *min_samples=6*. These are the parameters that, based on our analysis, lead to a better silhouette score, indicating good cohesion within clusters and good separation between clusters.

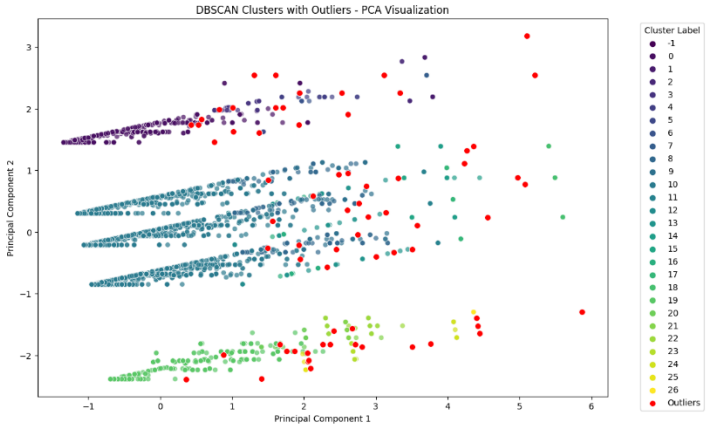


Figure – DBCAN for density and for the Outliers

From the DBSCAN view, there are 5 dense areas in the PCA plots, it could be an interesting indication of the structures. The 5 dense areas could represent regions of the state that share similar characteristics in the selected features. For example, they might correspond to areas with similar levels of crime, participation, or poverty that might correspond to specific geographic subdivisions in the state. Other information that can be retrieved are related to the Outliers that can help us to considers them atypical compared to the other data.

Respect to the K-Means we do not choose before the number of cluster but the DBSCAN itself discovers for us. It shows us clearly the differences for the outliers detection. And the cons are for the DBSCAN where the parameter (eps and min) are harder to tune respect of K-Means where there is only one parameter.

3.4. Hierarchical Clustering

In this phase we use we tried the four main linkage types (single, complete, average and ward) and each of them is been showed and analyzed. Tanking in consideration of always the elected 4 features taken as before. Following the Elbow method from the K-Means where we had 4 as best value of K, we follow this as the best "cut" result.

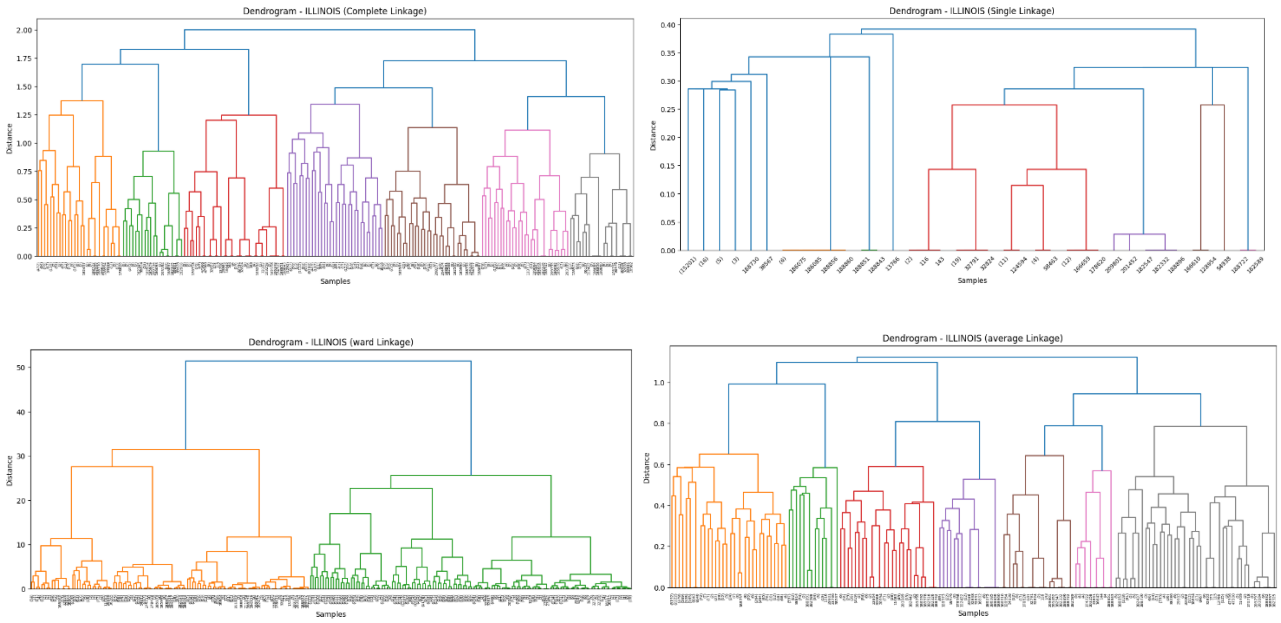


Figure 10 - Different Hierarchical methods

Then we repeated the same Hierarchical methods with a different metric, the *Manhattan distance* ('cityblock') that is not suitable for the ward hierarchical method. Then we show the indicator the silhouette coefficient results you obtained provide an indication of the quality of the clustering using different linking methods.
The results show that, based on the silhouette coefficient, the "Average" and "Ward" linkage methods appear to produce the best clustering method to apply.

If we want to consider the differences of K-Means and hierarchical for instance if we would have taken 1000 observations to be illustrated in the dendrogram, as a result it is extremely hard to be examined and extremely computationally expensive furthermore, the more observation we take the slower will gets while K-Means doesn't have this kind of issue.

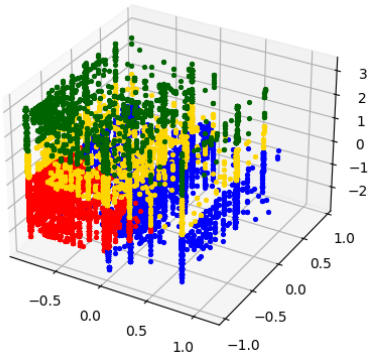
3.5. Comparisons of Clustering Techniques

- **K-Means**: Has been able to derive important analysis comparing the different feature providing different consideration.
- **DBScan**: Help the analysis to identify feature as noise, allowing the demonstration of all features selected.
- **Hierarchical**: Considering the different selections between Hierarchical and K-Means has been shown as the best interpretation is given by the usage of the K-Means, even if with the Hierarchical we consider only the Illinois as a state.

3.6. Other Clustering Approach

In this phase we also tested the Cure Clustering Method, taken from *PyClustering*.

- **X-Means**: The X-Means algorithm is a variant of the K-Means algorithm that attempts to solve the problem of choosing the optimal number of clusters automatically. X-Means starts with an initial cluster number and, after running K-Means on each cluster, uses some evaluation criteria (such as log-likelihood) to decide whether to divide each cluster into smaller sub-clusters. This new method of X-means can help to have better improvements instead of using the K-Means. Considering using the same features as discussed above we have shown a better cluster aggregation respect to the K-Means



- **Cure:** We applied the CURE algorithm to perform clustering on our data. CURE, an acronym for Clustering Using Representatives, is an algorithm that initially selects random representatives and enlarges them to efficiently capture the structure of clusters. The algorithm was configured with a desired number of clusters equal to N. After running CURE, we analyzed the results obtained, exploring the subdivision of the data into clusters based on the

selection and enlargement of the representatives. The use of CURE was motivated by its ability to deal with clusters of different shapes and sizes, as well as its robustness in handling outliers.

4. Predictive Analysis

The objective of the classification task was to identify if in the incident there have been at least a killed person or not. To do that, the starting dataset used is the "*incidents dataset*" that comes from the previous task (Data understanding and preparation).

4.1. Data Preparation

To do the experiments on the classification, we prepare the dataset. The dataset contains 239628 records and was not labeled but the '*n_killed*' feature allows you to identify whether at least one person was killed in the accident. With a simple rule ("n_killed >0") we labeled the entire dataset obtain the following distribution:

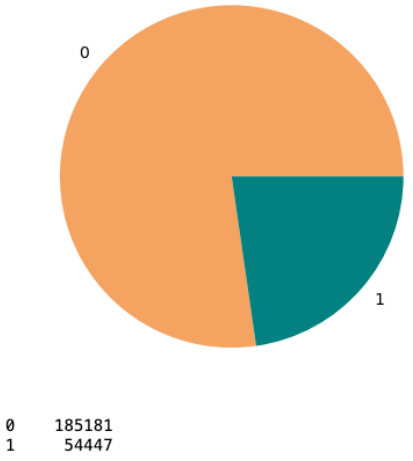


Figure 11 - No-killed/Killed incidents distribution

As can be seen from figure 1, the distribution is quite unbalanced on the "no-killed" class, which corresponds to 77% of the entire dataset.

Furthermore we decided to remove also those feature that we considered irrelevant for the classification such as *'totalvotes'*, *'candidatevotes'*, *'congressional_district'*, *'min_age_participants'*, *'max_age_participants'*,*'n_males'*,*'n_females'*,*'n_participants_child'*,*'n_participants_teen'*, *'n_participants_adult'*, removing also those features that could have a high correlation between them.

The dataset then, contains some categorical features that we discretize to perform the classification. These features were *'state'*, *'city_or_country'*, *'incident_characteristics1'*, *'incident_characteristics2', 'party'* that we removed after the discretization process, keeping those discretized.

4.2. Balancing the data

As we stated before, the classes are unbalanced. The problem is that the classifiers trained on imbalanced data sets can perform bad as they tend to overfit towards the majority class. So, we thought of doing an experiment on the normal dataset and one on the dataset where SMOTE was applied on the minority class, which is an oversampling technique that synthesizes new records.
After applying this technique, we obtained the following distribution:

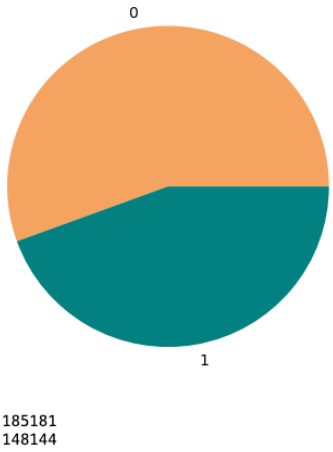


Figure 12 - No-Killed/Killed incidents distribution after SMOTE

Before applying any of the following classification algorithms, we splitted the dataset into training, validation and test sets, the latter consisting of 20% of the original data set. Furthermore, we used the *StandardScaler* algorithm from the scikit-learn library to standardize our data.

All the models were trained performing 6-fold cross validation on the entire development set to find the best hyperparameters. After CV, the model with the best found set of hyperparameters is retrained on the whole development set data, then tested on test set.

The models are implemented with scikit-learn and based on the model itself, the hyperparameters are properly chosen.

4.3. Classification Methods

Once the dataset was ready, we performed the analysis using different classifiers both on balanced and unbalanced method. We tested different classification models and performed a grid search with a cross-validation of 6 to find the best parameters. In the following part we will discuss the results on both the dataset where SMOTE has been applied and were not.

In the first experiments and analyses, we subjected the entire dataset to the different classifiers obtaining "extremely" optimal results. After careful analysis, we decided to remove features that were highly correlated with the number of people killed, as well as the n_killed feature itself. So, we decided to also remove the following features from the original dataset: '*n_killed'*, '*incident_characteristics1_num*, '*incident_characteristics2_num*'.

Evaluation. In order to evaluate the goodness of our models, we'll show classification metrics like accuracy, precision, recall and F1 score. The miss-classified patterns can be numerically viewed by using the confusion matrix (label 0 for non-killed, 1 for killed) and graphically by plotting the ROC curve. Talking about the test set, it is made up of imbalanced data, having the same distribution as the original dataset, thus it's without any sampling strategy applied. This means that any of the following algorithms should beat the dummy classifier.

4.3.1. Decision Tree

Decision tree is a classification method which produces interpretable results. It's not one of the best algorithms we tested, but, anyway, its performances are pretty good.

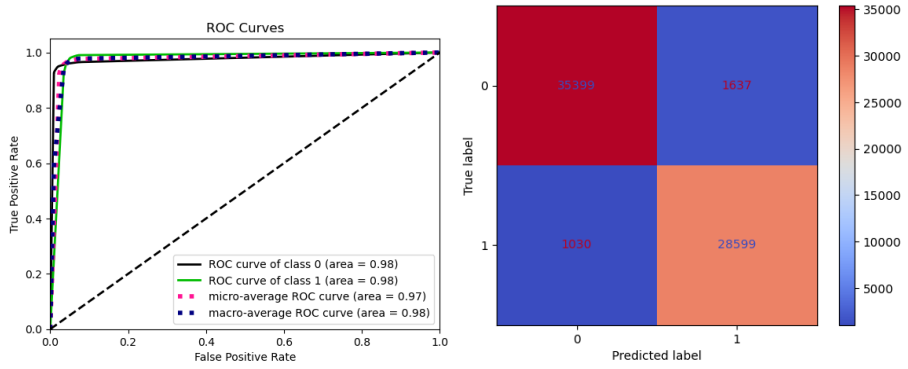


Figure 13 - Roc and Confusion Matrix for Decision Tree in balanced dataset

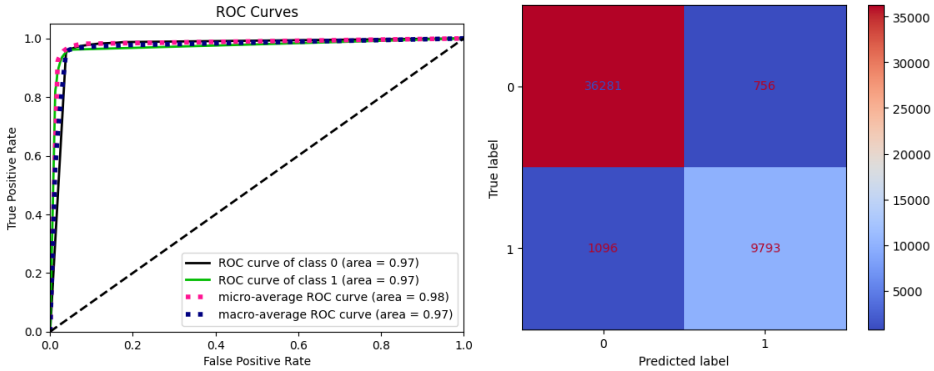


Figure 14 - Roc and Confusion Matrix for Decision Tree in unbalanced dataset

4.3.2. Random Forest

It's an ensemble method specifically designed for decision trees, it combines the predictions made by multiple decision trees and outputs the class that is the mode of the class' output by individual trees.

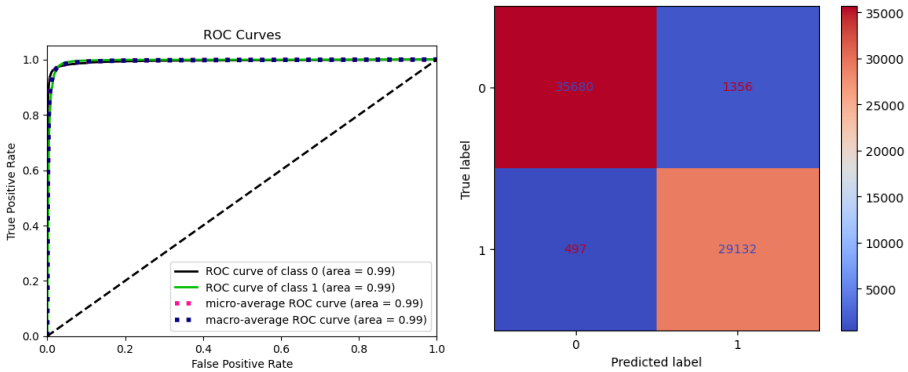


Figure 15 - Roc and Confusion Matrix for Random Forest in balanced dataset

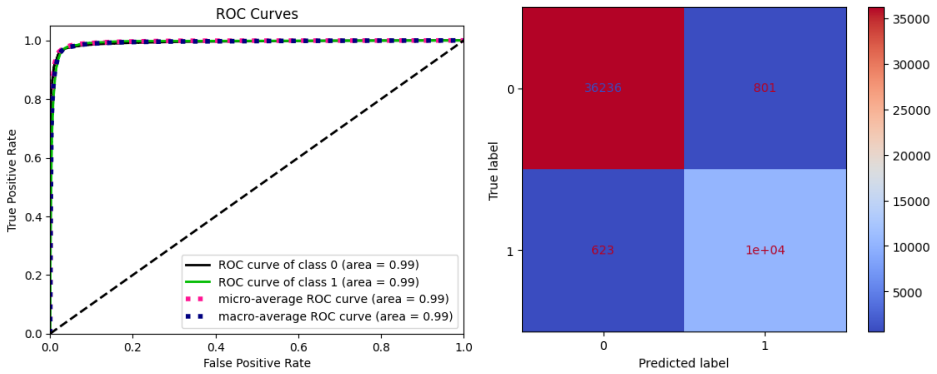


Figure 16 - Roc and Confusion Matrix for Random Forest in unbalanced dataset

4.3.3. K-nearest neighbors (KNN)

The KNN is an instance-based classifier, it uses class labels of nearest neighbors to determine the class label of unknown records.

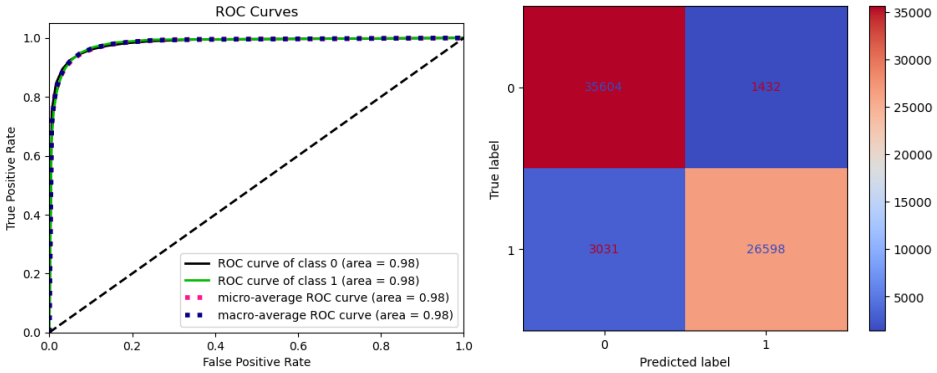


Figure 17 - Roc and Confusion Matrix for KNN in balanced dataset

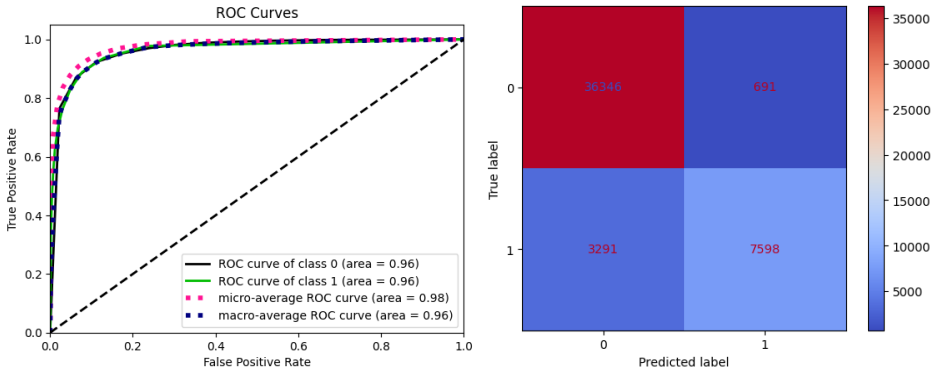


Figure 18 - Roc and Confusion Matrix for KNN in unbalanced dataset

4.3.4. AdaBoost

It can be used as ensemble of based classifiers to improve performance. The output of the base learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers.

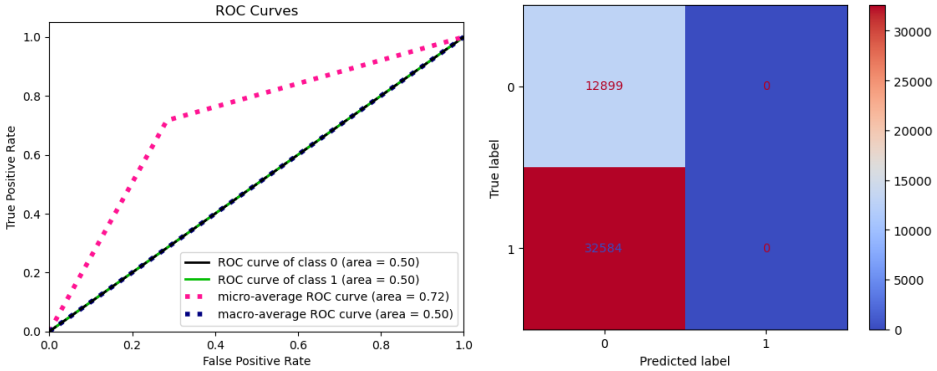
4.3.5. Rule-based

The rule-based classifier is a classification scheme that makes use of IF-THEN rules for class prediction. Due to a high computational demand and the imminent deadline, it was not possible to report here the results of the application of the model to both the dataset. For the sake of transparency, we report the results obtained on the first experiments carried out.

```
## Test:
                precision    recall  f1-score   support

        killed       0.28      1.00      0.44     12899
     no-killed       0.00      0.00      0.00     32584

      accuracy                           0.28     45483
     macro avg       0.14      0.50      0.22     45483
  weighted avg       0.08      0.28      0.13     45483
```



4.3.6. Naïve Bayes

The Naive Bayes classifier is a "probabilistic classifier" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

4.3.7. Support Vector Machine (SVM)

SVM is a robust classifier based on statistical learning frameworks, it maps training examples to points in space to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. Due to a high computational demand and the imminent deadline, it was not possible to report here the results of the application of the model to the unbalanced dataset. The reader can find the code in the notebook.

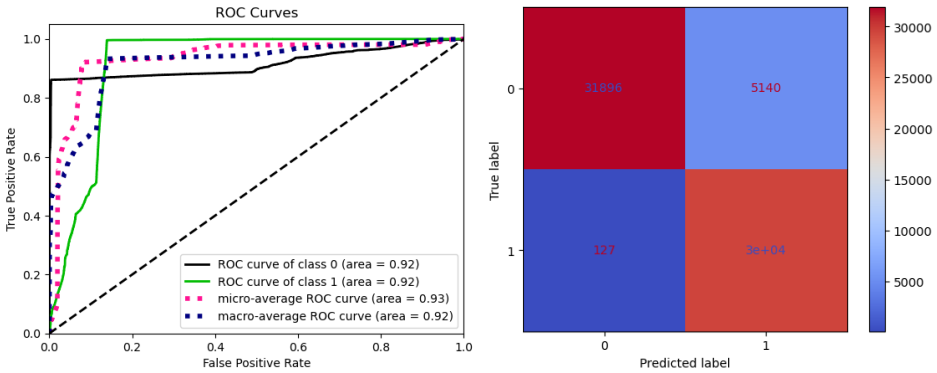


Figure 19 - Roc and Confusion Matrix for SVM in balanced dataset

4.3.8. Neural Network

We developed an FF neural network made up of two hidden layers whose neurons are 'activated' by the ReLU function and a softmax on the output layer. To ensure the generalization capability, we use the dropout technique and the layer normalization, to stabilize the training phase. In the end, we used the categorical cross entropy as loss function and early stopping to avoid overfitting.

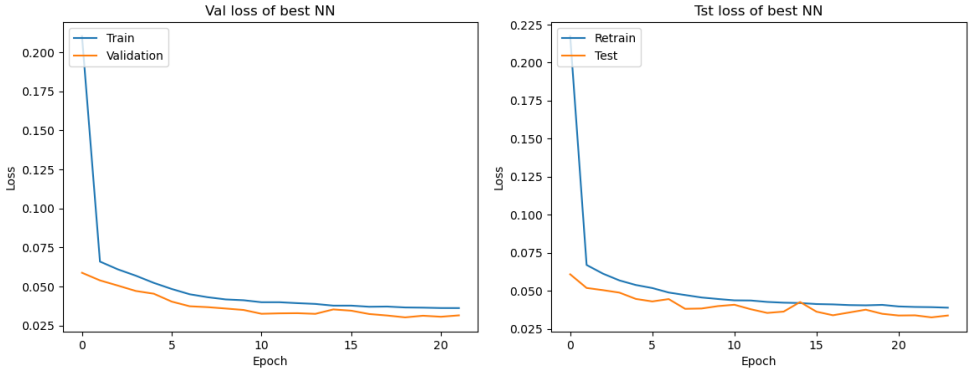


Figure 20 - Validation and Test Loss for Neural Network classifier in balanced dataset

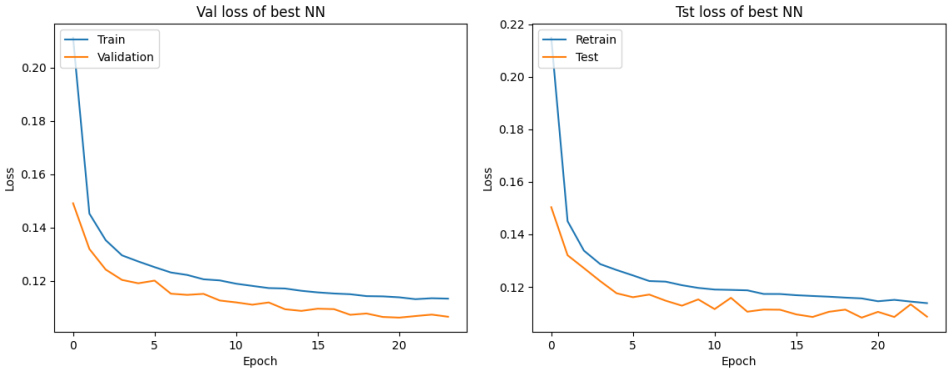


Figure 21 - Validation and Test Loss for Neural Network classifier in unbalanced dataset

4.4. Comparison

The results of the forenamed algorithms are shown in the following tables. In Table 1 we report the performance of the algorithms on a balanced dataset, in Table 2 we report the performance of the algorithms on unbalanced dataset. In general there is no model that outperforms the others, but those that have performed better can be distinguished.

The application of the models in an unbalanced dataset is consistent with what was expected from the theory, classification models tend to perform sub optimally with imbalanced datasets.

Model	Class	Test			
		Accuracy	Precision	Recall	F1
Decision Tree	killed	0.96	0.97	0.96	0.96
	no-killed		0.95	0.97	0.96
KNN	killed	0.93	0.92	0.96	0.94
	no-killed		0.95	0.90	0.92
Random Forest	killed	0.97	0.99	0.96	0.97
	no-killed		0.96	0.98	0.97
SVM	killed	0.69	0.72	0.74	0.73
	no-killed		0.66	0.63	0.65
AdaBoost	killed	0.99	1.00	0.99	0.99
	no-killed		0.98	1.00	0.99
Rule based	killed	0.28	0.28	1.00	0.44
	no-killed		0.00	0.00	0.00
Bayesian Network	killed	0.72	0.77	0.70	0.73
	no-killed		0.66	0.74	0.70
Neural Network	killed	0.99	0.99	0.99	0.99
	no-killed		0.99	0.99	0.99

Table 1 - Comparison result for different models (balanced dataset)

Model	Class	Test			
		Accuracy	Precision	Recall	F1
Decision Tree	killed	0.96	0.97	0.98	0.98
	no-killed		0.93	0.90	0.91
KNN	killed	0.92	0.92	0.98	0.95
	no-killed		0.91	0.70	0.79
Random Forest	killed	0.97	0.98	0.98	0.98
	no-killed		0.93	0.94	0.94
SVM	killed	NaN	NaN	NaN	NaN
	no-killed		NaN	NaN	NaN
AdaBoost	killed	0.96	0.97	0.98	0.98
	no-killed		0.93	0.91	0.92
Rule based	killed	0.28	0.28	1.00	0.44
	no-killed		0.00	0.00	0.00
Bayesian Network	killed	0.80	0.79	0.99	0.88
	no-killed		0.86	0.12	0.21
Neural Network	killed	0.96	0.98	0.97	0.97
	no-killed		0.90	0.93	0.91

Table 2 - Comparison result for different models (unbalanced dataset)

5. Time Series Analysis

5.1. Time Series

In this section we extracted the time series from the incidents that occurred in the years [2014,2015,2016,2017], computing for each week of the 4 years a score based on the severity of the incident:

$$\frac{(n_{killed} + n_{injured} + n_{arrested})}{n_{participants}}$$

Each value of the time series corresponds to the score value of a certain city for a certain week of a certain year. We filtered out all the entries with a number of weeks with incidents less than 15% of

the total number of weeks in the years. The goal of this analysis is to group similar cities based on the severity score.

We plotted the time series for the severity score of three cities:

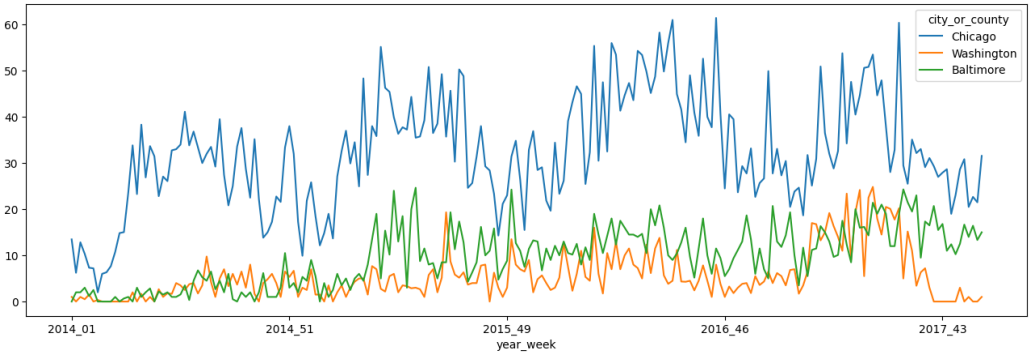


Figure 22 - Example of cities' timeseries

Taking into consideration the city with the highest scores (Chicago) we show the average score for each year:

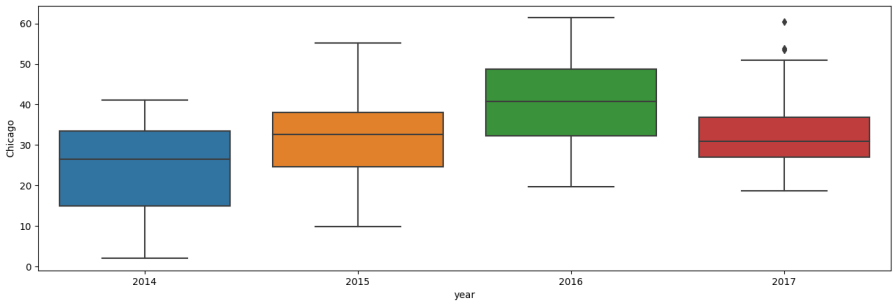


Figure 23 - Avarage year score for Chicago

We used the k-means clustering from the *tslearn* package to find those similar groups of cities and in order to choose the best values of k (the number of clusters) we based our choice on the SSE, silhouette and Davies-Bouldin scores. Afterwards, we took the best k-value and we used it to do the final clustering on the timeseries dataframe.

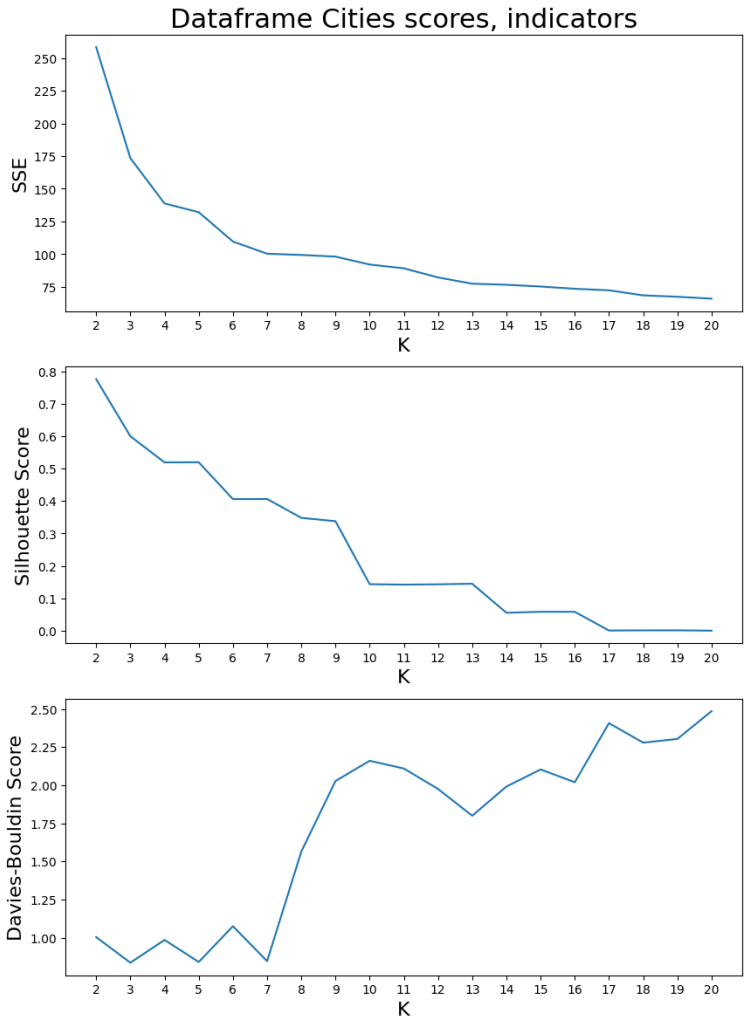


Figure 24 - Different plots for chosing the best k value

As the figure shows, by using the elbow rule and by looking at the other two values we took 4 as the best value for k and the final clustering analysis provide the following results.

As we can see the cluster are well-separated based on their severity score, but we can see an outlier which is Chicago whose severity scores are very high compared to the others. The result is that Chicago is an entire cluster. The following plots are different representation of the clusters behaviors that we found.

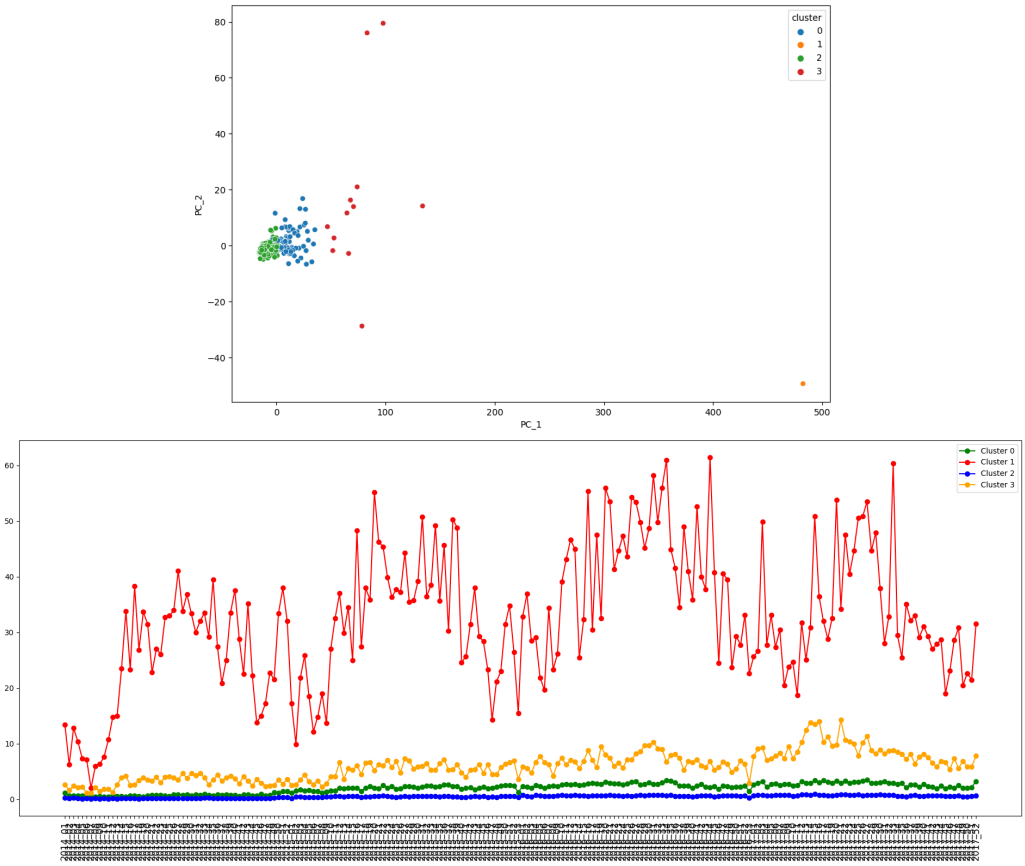
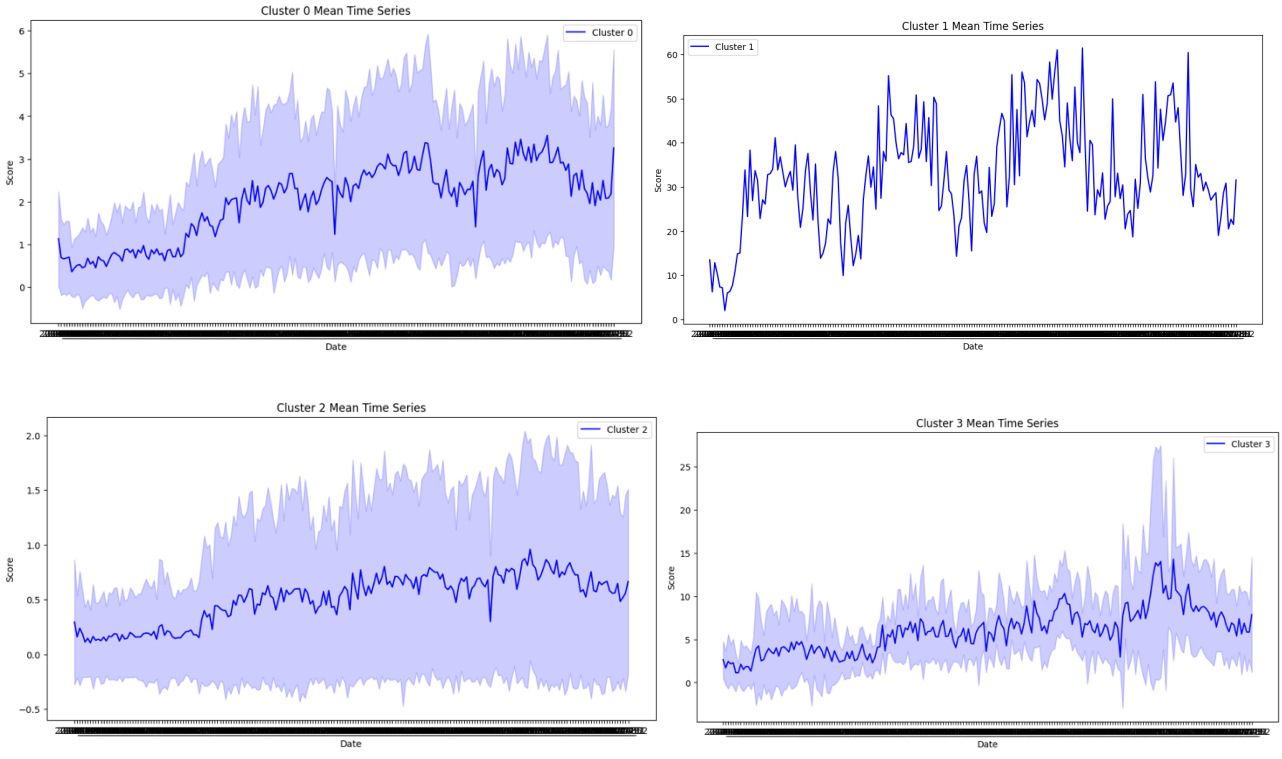


Figure 25 - Scatter plot clustering

The cities are separated in four different clusters in accordance to their severity score trends. For the sake of visualization, we plot here the mean of each cluster to show the general behavior of cities withing a cluster.



5.2. Motifs/Anomalies extraction

Motifs are repeated patterns in the time series. We discovered motifs by computing the Matrix Profile: a data structure that annotates time series by using a sliding window to compare the pairwise distance among the subsequences.

For our purposes we decided to extract motifs and anomalies w.r.t the city with the highest severity scores (Chicago).

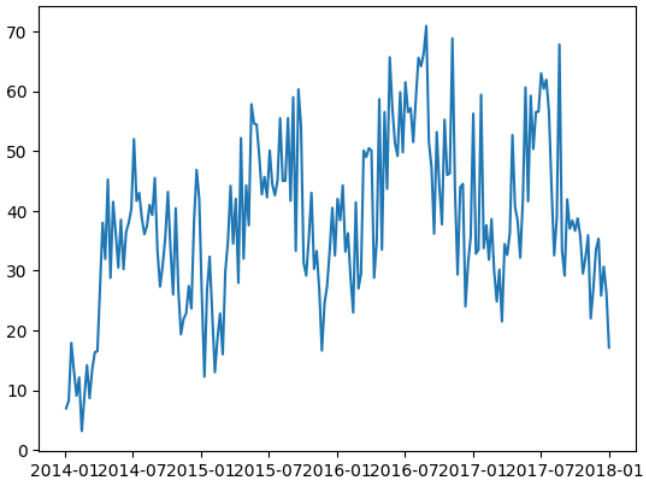


Figure 26 - Initial timeseries of Chicago

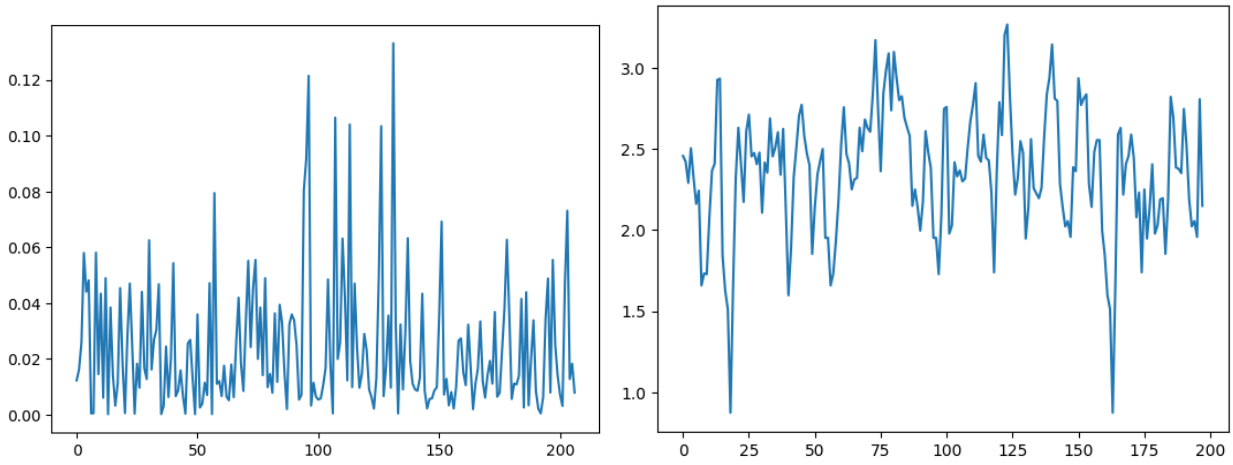


Figure 27 - STOMP resulting plot (window size=12 [left] and size=3 [right])

As we can see from the motifs graph, we extracted different motifs which are recurring patterns that appear frequently in a time series. These patterns represent important events or behaviors that recurs during time. We found 5 different motifs.

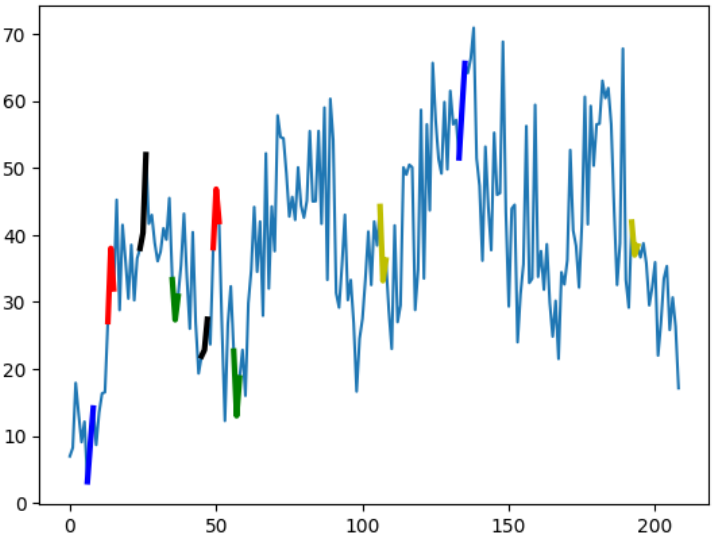
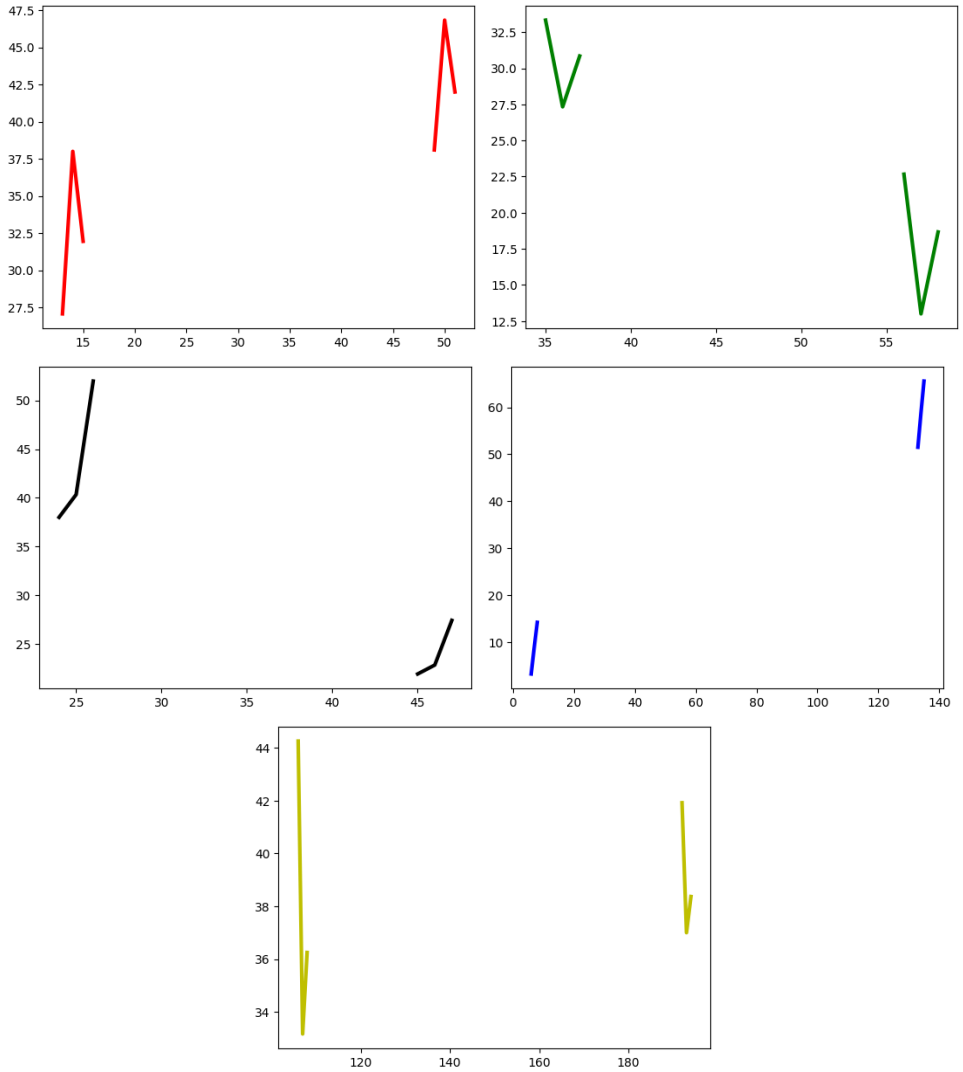


Figure 28 - Motifs plot

In the following plots, we highlight the single motif identified



We also analyzed the timeseries with the aim of discovering anomalies, also known as outliers which are datapoints that deviate significantly from the normal behavior of the timeseries. In the following plot we highlight what we found.

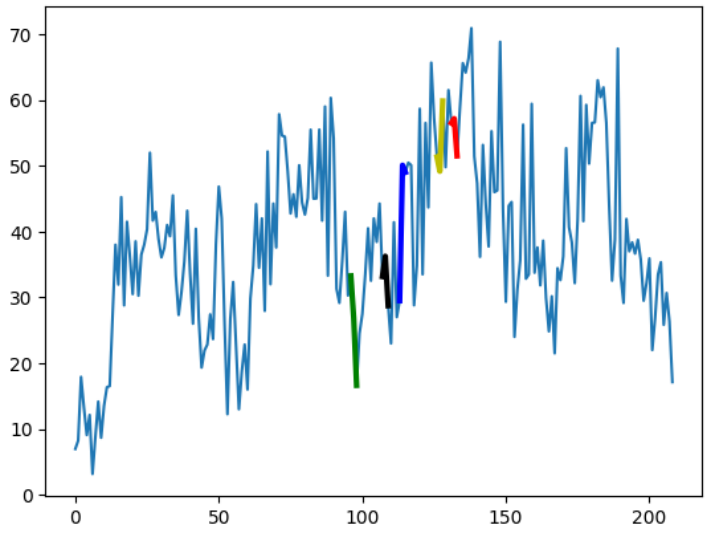


Figure 29 - Anomalies in the Chicago timeseries

5.3. Shapelet

For what we concern the shapelet identification, our aim was to identify discriminative subsequences to capture the essence of a particular pattern in the timeseries.
Due to the incoming deadline, we were not able to present significant results still the work done can be found in the notebook.