Data Mining Project

Calogero Turco, Giovanni Cupitò, Davide Pirolo

January 8, 2024

1 Task 1 - Data Understanding and Data Preparation

1.1 Task 1.1 - Data Understanding and Correction

1.1.1 Incidents dataset

Analyzing the dataset incidents.csv, the first step was to remove duplicate records, which were 296. We noticed that the number of participants is 0 when the perpetrators have not been identified, and there are no other individuals involved (the number remains zero even in cases where there was a police-involved shooting). We found that when there is 1 participant it is likely a case of suicide (from incidents characteristics features).
In this dataset, we encountered incidents that had dates with years 2028, 2029, and 2030. Meanwhile the 2018 records were related to only early months. We proceeded by transforming the records with the year 2028 into records with the year 2018 as they are related to the later months of the year, suggesting that they make help in completing the year and nonetheless they are very few, so they do not make a real impact on the year data. As for the records with the years 2029 and 2030, we replaced the year with the mode (most common year) for the respective state and city if available; otherwise, we considered only the mode for the state.
We chose to not transform the records with the years 2029 and 2030 into 2019 and 2020 because there were very few records. We assumed that the distributions across the years should be similar. Upon closer examination, we noticed that one record for the year 2030 actually described an incident that occurred in 2015, and we corrected the year accordingly.
We removed the records that had the year 2013 because, although there were only a few of them (253), they deviated significantly from the distributions of the other years.
We fixed the missing values in the latitude and longitude (2D coordinates) columns by grouping them by state and city and taking the average of the values for the city and state in 3D coordinates, in this way average was calculated taking into account the spherical shape of the Earth, allowing to get a center point per pair and then converting back to 2D coordinated. In cases where the city was missing, we considered only the average for the state.
If the address field was null, we inserted the string 'Missing'. For the missing values in the 'congressional_district' column, we took the mode of it for each state and city. We note that the 'congressional district' can change in the years, but the mode in our case remains consistently accurate and is not a wrong value in the years; we will deal again with them further when merging with elections dataset. We did not modify the 'state_house_district' and 'senate_district' columns because they will not be used in future tasks.
Looking at the 'participant_gender1' column, we noticed that in only one record, we found the value 'Male, female,' which does not belong to either 'Male' or 'Female.' We replaced this value with 'Female' because, upon checking the notes, we found that the incident referred to a woman. However, we did not make any further changes to this column because we will remove it as it is not significant.
For the 'incident_characteristics1' and 'incident_characteristics2' attributes, we inserted the string 'Missing value' where the values were null. For the columns related to ages ('min_age_participants', 'max_age_participants', and 'avg_age_participants'), we converted values ranging from -100 to 0 by changing the sign, assuming it was a typographical error. Values greater than 100 or less than -100 were set to null values.
For the columns 'avg_age_participants,' 'min_age_participants,' and 'max_age_participants,' we looked

at the means and medians of these data both by state and by year. We noticed that the distribution of the mean and median does not change by year, but it does change by state. To preserve this distribution, we decided to replace missing values with the median calculated for each state. Additionally, to be more precise in the distributions, apart from grouping by state, we also grouped by 'incident_characteristics1' and 'incident_characteristics2' where available and replaced missing values with the median obtained from grouping by these values.
For the 'participant_age1' column, the value is randomly selected based on the value in the 'participant_age_group1' column, taking into account the previously calculated 'min_age_participants' and 'max_age_participants' values.
For the columns 'n_participants_child,' 'n_participants_teen,' and 'n_participants_adult,' we performed similar analyses as mentioned above. We looked at the means and medians by state and by year, and also examined the distributions. We observed that there was much more variation by state compared to by year. We transformed the negative values, in these features, into positive values and conducted data consistency checks.
We started with the assumption that the 'n_participants' column is correct and checked that the sum of values in the 'n_participants_child,' 'n_participants_teen,' and 'n_participants_adult' columns is equal to the total number of participants. This is a reasonable approach to ensure data consistency and accuracy.
In cases where this condition is not satisfied, we first inserted a null value and then replaced it by looking at the proportions by state while adhering to the constraint of the total number of participants. This approach helps maintain data integrity and consistency. Also for the columns 'n_males' and 'n_females' we saw the average and the median by state and year, preserving the variation per year, considering the correct number of participants, we replaced the incorrect values with the proportions calculated per year as done above.
For the columns 'n_killed,' 'n_unharmed,' 'n_arrested,' and 'n_injured,' **we noticed that they are mutually exclusive, so the sum of these four attributes should be equal to the total number of participants**. These values will be adjusted by first calculating the proportion by state and year for each attribute and then replacing inconsistent values with the calculated proportions, as done previously.

1.1.2 PovertyByStateYear Dataset

From the analysis of this dataset, we noticed that there are no duplicate records. In addition to the 50 states plus the federal district (District of Columbia), there is also a record for the average for all states for each year.
Visualizing the values graphically, we notice that the distribution resembles a Gaussian distribution. We also observed that there are no records for the year 2010 for the state of Wyoming, but there are two records for the year 2009. We took action by changing the second record to the year 2010. Furthermore, we noticed that for all states, the poverty value is missing for the year 2012. We addressed this by taking the average between the years 2011 and 2013, considering that poverty tends to exhibit a consistent increasing or decreasing trend.

1.1.3 YearStateDistrictHouse Dataset

From the analysis of this dataset, we notice that we do not have duplicate records. The elections are held every two years. We observe issues with the District of Columbia elections, which appear only for the year 2020. We have added the elections for the years '2014,' '2016,' and '2018', obtaining them from Wikipedia. We did not include the entire voting history because our main dataset covers the period from 2013 to 2018.
From an observation of the box plots of the data, we noticed outliers: for the state of 'Florida', the 2020 election values were negative. We took corrective action by searching for the correct election data through Wikipedia.
For the state of Minnesota, the winning party is reported multiple times as 'DEMOCRATIC-FARMER-LABOR,' which is an affiliate of the Democratic Party. We replaced this value with 'DEMOCRATIC' for convenience.
We noticed that for the state of 'Maine', there are outliers in the 2022 election data related to the total number of votes and the number of votes from participants. We decided not to modify these

data because our area of interest spans from 2014 to 2018.

1.1.4 Data Aggregation

To merge the datasets, we decided to omit some columns that we did not consider useful for our future evaluations. We removed the columns 'state_house_district,' 'state_senate_district,' 'address,' 'notes,' 'participant_age1,' 'participant_age_group1,' and 'participant_gender1.'
The first issue encountered concerns the congressional district, where in cases where a state has only one district, it is reported with the identifier '1' in the 'incidents' dataset and with the identifier '0' in the 'YearStateDistrictHouse' dataset. This occurs for the states: "Alaska," "Vermont," "North Dakota," "Wyoming," and "Delaware." We resolved this by indicating '0' as the identifier for the single congressional district in these states for both datasets. In the state of "Delaware," an identifier '4' appears for a congressional district, which we replaced with the value '0' since this state has only one congressional district.
The state of "West Virginia" has only 4 congressional districts. We found an incorrect value in the 'incidents' dataset for this state with identifier '6,' and we replaced it by taking the mode of congressional districts for this state in the 'incidents' dataset. We did the same for the state of "Oregon", where incorrect values related to congressional districts '9' and '10' appeared, which do not exist. Similarly, for the state of "Kentucky", where congressional district '8' does not exist.
Another issue was related to the name of the District of Columbia, which was spelled differently in the three datasets due to typographical errors. Regarding the merging strategy for combining the dataset of biennial elections with the dataset of incidents, we considered the records from the previous years for the years in which election data were missing.

1.1.5 Data correlation

As we could imagine, there are correlated data: those between the ages of participants (min, max, and avg participant age) with 'participant_age1,' the number of adult participants, and the number of teen participants; finally, the number of male participants with the total number of participants, we can see them in figure 1.
As was to be expected, we find that the data related to each other are: the winning candidate's votes and the total votes, as seen in figure 2.

1.2 Task 1.2 - Data Preparation

We deliberately decided to not remove outliers (mass murders, mass shootings), to capture the full range of data and make a comprehensive analysis of the underlying patterns and trends. As requested, we added new numerical features to get better clustering.

Percentage of males involved in accidents compared to the total number of males for the same state and in the same period
As a period we considered first the years and then the months. Below each description there is the code used to calculate it.

```
total_males_year = non_scaled_df.groupby([non_scaled_df['date'].dt.year, 'state'])
    ['n_males'].transform('sum')
df['male_percentage_year'] =
    round((non_scaled_df['n_males'] / total_males_year) *100,2)
```

Ratio of people unharmed in accidents to those unharmed during the same period
As a period we considered first the years and then the months.

```
total_unharmed_year = non_scaled_df.groupby([non_scaled_df['date']
    .dt.to_period("Y")])['n_unharmed'].transform('sum')
df['ratio_unharmed_over_period_year'] =
    non_scaled_df['n_unharmed']/total_unharmed_year
```

Trend of poverty
The indicator shows the poverty increase and decrease in each year for each state compared to the previous year.

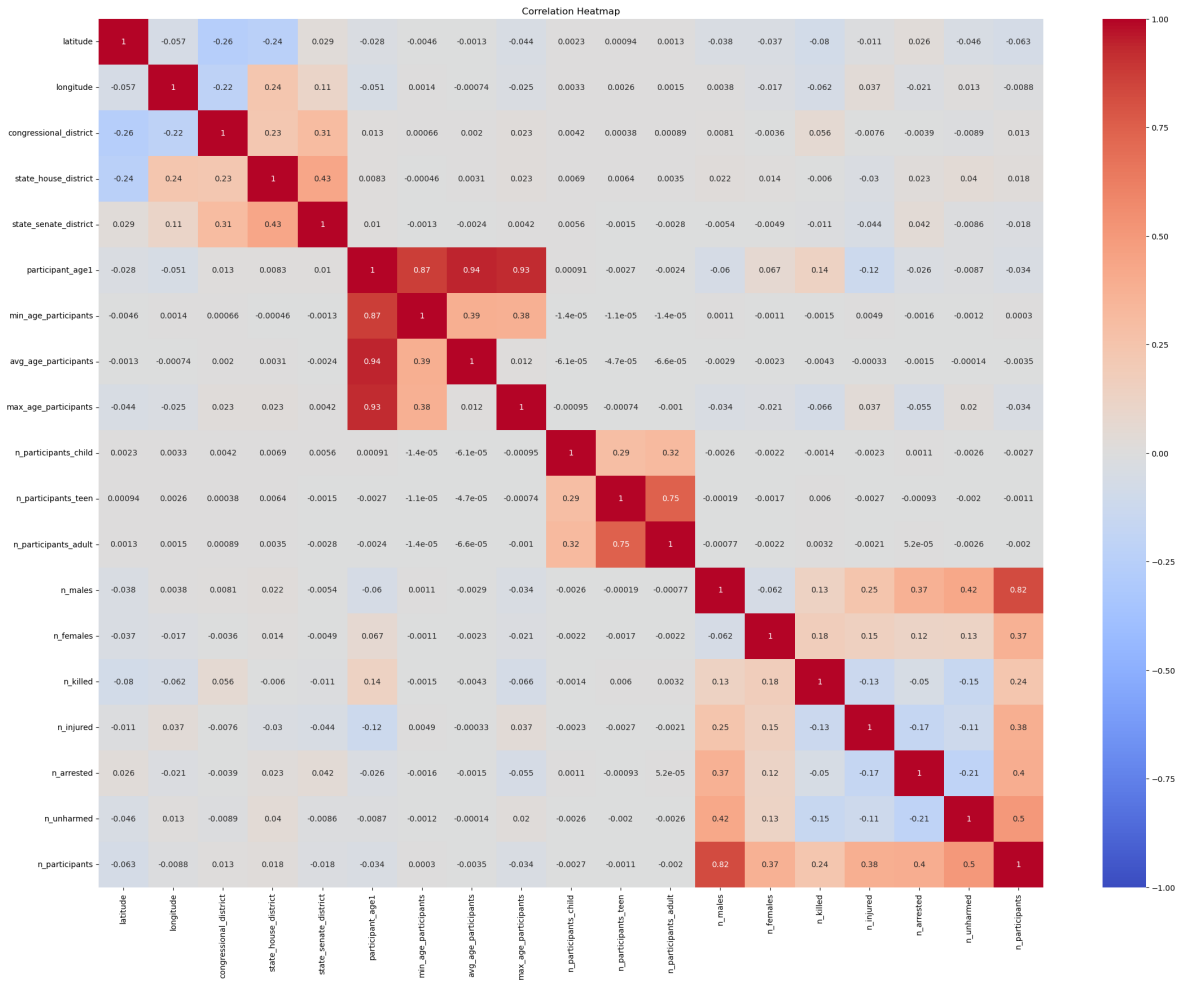


Figure 1: Dataset incidents correlation heat map. The closer the color to dark red/dark blue the more data is related

```
df_3 = df_3.sort_values(by=['state', 'year'])
df_3['percent_change'] = df_3.groupby('state')['povertyPercentage'].transform(
    lambda x: round((x - x.shift(1)) / x.shift(1) * 100,2)
```

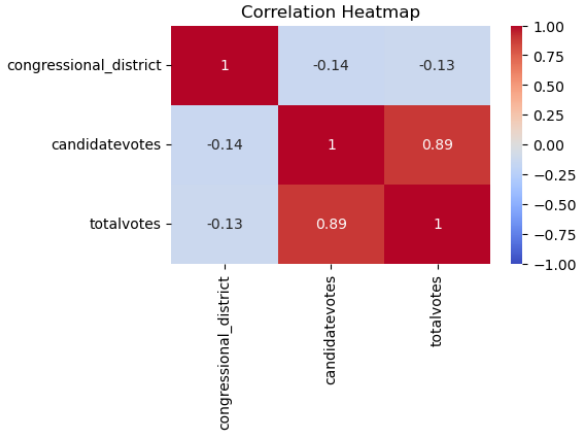


Figure 2: YearStateDistrictHouse dataset correlation heat map. The closer the color approaches dark red/dark blue the more data are related

```
)
```

As a variation of the indicator we multiplied the previous result with the number of participants.

Winning party
For each state, for each Congressional district, we considered the political party that won most of the elections.

```
df_2['state_district_modal_party'] = df_2.groupby(['state', 'congressional_district'])
['party'].transform(lambda x: x.mode().iloc[0])
```

Most successful party by state
We have calculated the mode in each state and Congressional district between several years, thus showing the most winning party for each state.

```
df_2['stronghold_state_party'] = df_2.groupby(['state', 'congressional_district'])
    ['party'].transform(lambda x: x.mode().iloc[0])
result = df_2.groupby(by=['state'])[['stronghold_state_party']].first()
```

State poverty compared to national average
For each year we compared the poverty of each state to the national average. In case it is greater we have added a label true, otherwise false.

```
df_3['above_national_mean_poverty'] = df_3.groupby(['year'])['povertyPercentage']
    .transform(
         lambda x: x > df_3[(df_3['state'] == 'United States')]['povertyPercentage']
            .iloc[0]
    )
```

Ratio between the number of arrested and participants compared to the average per year and state

```
df['ratio_participants_arrested'] =
    non_scaled_df['n_arrested'] / non_scaled_df['n_participants']
df['ratio_participants_arrested'].fillna(0,inplace=True)
year_state_ratio = df.groupby(['year', 'state'])[['ratio_participants_arrested']]
    .mean()
for index, row in df.iterrows():
  state = row['state']
  year = row['year']
  mean_value = year_state_ratio.loc[year, state]['ratio_participants_arrested']
  df.at[index, 'ratio_arrested_state_year'] =
    df.at[index,'ratio_participants_arrested']/mean_value
```

Ratio between the ratio of male or female participants to the number of participants and the average per state and month of the ratio

```
df['ratio_participants_males'] =
    non_scaled_df['n_males'] / non_scaled_df['n_participants']
df['ratio_participants_females'] =
    non_scaled_df['n_females'] / non_scaled_df['n_participants']
df['ratio_participants_males'].fillna(0,inplace=True)
df['ratio_participants_females'].fillna(0,inplace=True)
month_ratio_male =
    df.groupby([df['date'].dt.month, 'state'])[['ratio_participants_males']].mean()
month_ratio_female =
    df.groupby([df['date'].dt.month, 'state'])[['ratio_participants_females']].mean()
df['month_ratio_male'] = 0.0
df['month_ratio_female'] = 0.0
for index, row in df.iterrows():
```

```
  state = row['state']
  month = row['date'].month
  mean_value = month_ratio_male.loc[month, state]['ratio_participants_males']
  df.at[index, 'month_ratio_male'] =
    df.at[index,'ratio_participants_males']/mean_value
  mean_value = month_ratio_female.loc[month, state]['ratio_participants_females']
  df.at[index, 'month_ratio_female'] =
    df.at[index,'ratio_participants_females']/mean_value
```

Entropy
We calculated the entropy between the number of killed, injured, arrested and unharmed and also the number of adult children and teenagers.

$$H(x) = -\sum_{i=1}^{n} p_i log_2 p_i$$

Ratio of number of participants to number of participants per year or month

```
year_difference_participants = df.groupby([df['date'].dt.year,
'state', 'city_or_county'])[['n_participants']].mean()
 mean_value_month = month_difference_participants.
 loc[(month, state, city), 'n_participants']
    df.at[index, 'avg_month_killed'] =
    row['n_participants'] / mean_value_month if mean_value_month != 0 else 0
```

Product between number of participants and poverty rate

```
df['n_participants_poverty'] =
    non_scaled_df['n_participants'] * df['povertyPercentage']
```

Ratio of number of injured over number of injured

```
df['ratio_unharmed_injured'] =
    non_scaled_df['n_unharmed'] / non_scaled_df['n_injured']
```

Ratio of killed over unharmed

```
df['ratio_killed_unharmed'] =
    non_scaled_df['n_killed'] / non_scaled_df['n_unharmed']
```

Ratio of number of arrested over number of injured

```
df['ratio_arrested_injured'] =
    non_scaled_df['n_arrested'] / non_scaled_df['n_injured']
```

Ratio of killed over arrested

```
df['ratio_killed_arrested'] =
    non_scaled_df['n_killed'] / non_scaled_df['n_arrested']
```

Ratio between the number of votes of the winning candidate over the number of adult participants

```
df['winnerdivparticipants'] = df['candidatevotes'] /
non_scaled_df['n_participants_adult']
df['winnerdivparticipants'].replace([np.inf, -np.inf, np.nan], 0, inplace=True)
```

Count the number of accidents occurred in the state, before each accident
In this indicator we have calculated the number of accidents that occurred with respect to the date of each specific accident in the respective state.

```
df['historical_incident_freq'] = df.groupby('state')['date'].rank()
```

Counting of accident characteristics
We coded the concatenation of 'incident_characteristics1' and 'incident_characteristics2' with a hashing algorithm and counted each result obtained.

```
df['encoded_incident_characteristics'] = df['incident_characteristics1']
    .str.cat(df['incident_characteristics2'], sep='_').apply(deterministic_hash)
```

Severity of the accident
We calculated for each accident the severity of it, by dividing the sum of the number of arrested and the number of injured with the number of participants. Also to avoid division with zero, we added the number of participants.

```
df['severity_per_participant'] =
    (non_scaled_df['n_killed'] + non_scaled_df['n_injured']) /
    (non_scaled_df['n_participants']+1)
```

Ratio of age interval to state average and month
The age range was calculated by subtracting the maximum age from the minimum age.

```
age_diversity_state_month_means =
    non_scaled_df.groupby(['state', non_scaled_df['date'].dt.month])
    [['min_age_participants', 'max_age_participants']].apply(
        lambda x: (x['max_age_participants'] - x['min_age_participants']
    ).mean())

for index, row in non_scaled_df.iterrows():
    state = row['state']
    month = row['date'].month
    mean_value = age_diversity_state_month_means.loc[state, month]
    non_scaled_df.at[index, 'age_diversity_state_month'] =
        (row['max_age_participants'] - row['min_age_participants']) / mean_value
```

Boolean indicators
We have created these Boolean indicators that will not be used in clustering but will be useful for classification, but also to better understand the results in each cluster.

```
df['has_identified_participants'] = non_scaled_df['n_participants']>0
df['has_killed'] = non_scaled_df['n_killed'] > 0
df['has_underage'] = (non_scaled_df['n_participants_teen'] > 0) |
(non_scaled_df['n_participants_child'] > 0)
df['has_females'] = non_scaled_df['n_females'] > 0
```

The following have been realized by looking at the incident_characteristics1 and 2 features. Those indicators will serve us in characterizing the clusters created with the previous crafted indicators, they act as labels describing the clustering results.

- may_be_suicide with also a check in n_participants==0
- has_shooting
- is_non_shooting
- has_drug_involvement
- is_gang_related
- has_police_involved
- has_domestic_violence
- is_hate_crime

2 Task 2 - clustering

We started with almost all indicators previously defined, before doing features selection. The indicators chosen for clustering allow us to provide insights into socioeconomic disparities, criminal activities, and demographic trends within a region. They enable the analysis of how poverty levels correlate with crime rates, the impact of gender distribution on guns incidents over months, and the influence of political landscapes on criminal incidents. Additionally, these metrics allow for the examination of crime severity, outcomes of criminal incidents, and participant profiles in relation to varying socioeconomic backgrounds (poverty).

2.1 Discretization, normalization and outliers removal

To make the categorical data discrete we chose the LabelEncoder. The attributes interested are: 'state_district_modal_party', 'stronghold_state_party' e 'above_national_mean_poverty'. After that, we apply on data the QuantileTransform, that allow us to delete outliers as take into account the interquartile range, and give as an output uniform distribution data, which is stated to work well with clustering algorithms as K-Means. Then, through the MinMaxScaler we normalize data in an interval between 0 and +1.

2.2 Features selection

Through the application of the Pearson correlation formula, we chose to delete the features that have correlation greater than 0.7. The attributes that resulted high correlated with others and were removed are:'above_national_mean_poverty_encoded', 'stronghold_state_party_encoded', 'ratio_arrested_state_year', 'male_percentage_month', 'n_participants_poverty', 'avg_year_killed', 'n_participants_increase_decrease', 'ratio_unharmed_over_period_month'. We therefore are doing the clustering based on 21 features.

2.3 K-Means

2.3.1 Parameters

The first clustering algorithm that we implemented is the K-Means. To find the best number of clusters (K) we tried to minimize the SSE, (figure 3). To calculate it we used the MiniBatchKMeans as we read in the ScikitLearn documentation, that is suitable for large dataset. We calculate also the Silhouette

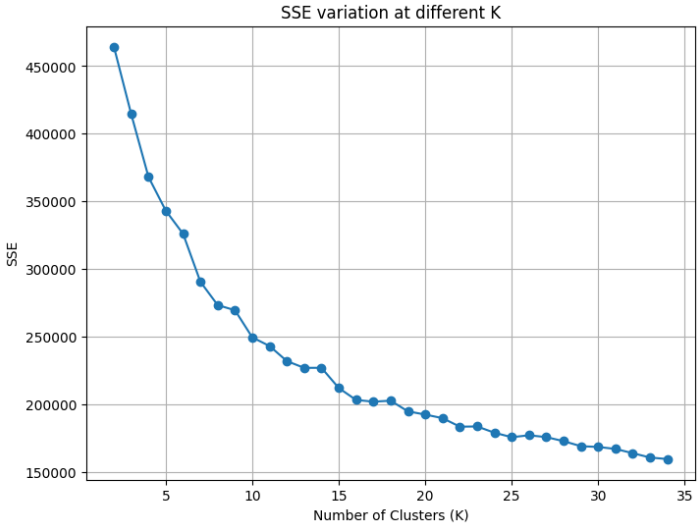
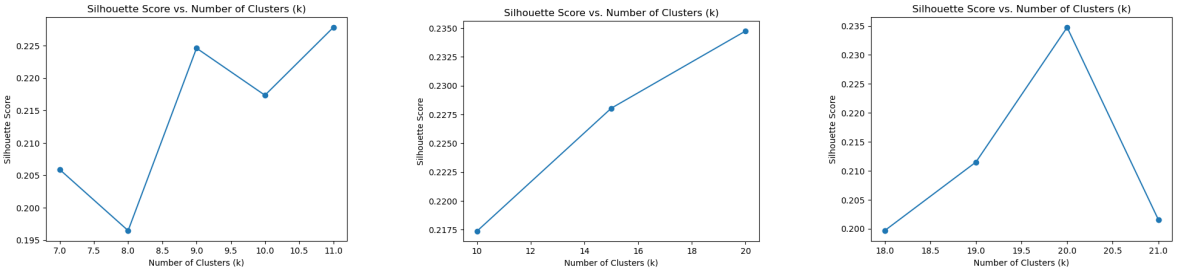


Figure 3: Show how SSE varies across the K

score to verify, with another measurement, the cluster goodness when K varies.
The graphs for the Silhouette score are three, can see them in figure 4, due the fact that generating a single chart would take too much time.



(a) $7 \leq k \leq 11$ (b) $10 \leq k \leq 20$ (c) $18 \leq k \leq 21$

Figure 4: Show how Silhouette score varies across K

We chose K equal to 11 and not more because even if the SSE decreases and the Silhouette Score has its peak with K = 20 by increasing K we may would reduce the understandability and make harder the analysis on clusters. The SSE value for K=11 is 237787.14 while the Silhouette score is 0.22.

The plotting of 2D and 3D PCA, along with t-SNE visualization can be seen in the

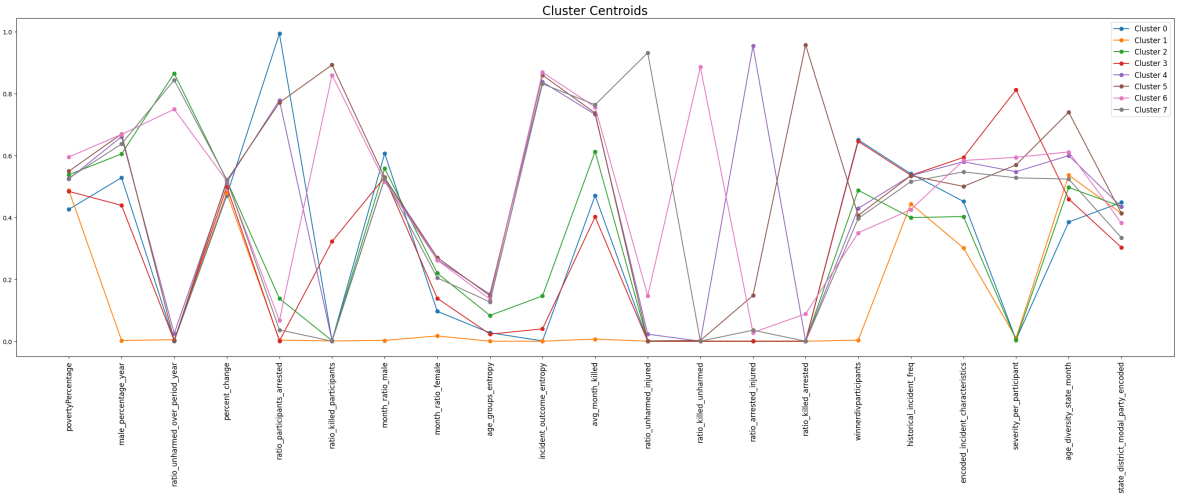


Figure 5: Show of centroids in the indicators.

notebook, for K-Means an all other techniques. We also present the histogram contour plot allowing to see the distribution of labels interactively. Please refer to the notebook for those figures. Note at section 3.3 of the notebook on clustering we show what are the features that influence the PCA components for the datasets considered (USA and state wide). Also *for the following description we refer to graphs that are not present here, due the lack of space. You can find them in our notebook file.*

2.3.2 Understanding the cluster data.

We notice that in the clusters the number of records is almost homogeneous, we can see that 1 has many more incidents than the other clusters. In the following there is the list of clusters with relative quantities:0: 16580, 1: 54713, 2: 24660, 3: 13303, 4: 10951, 5: 25488, 6: 15392, 7: 20679, 8: 18920, 9: 13868, 10: 24574

- cluster 2 uniquely presents cases with unidentified participants (*has_identified_participants*).
- clusters 4 and 6 are distinguished from other clusters as they only hold incidents with killed persons(*has_killed*). Cluster 1,5 hold a mixture of incidents with killed or not, while all other cluster have incidents without victims.
- the females are distributed proportionally in the various clusters(*has_females*), besides cluster 2 having no females.

- suicides are presents only in clusters 1,4,5 and 6 (*may_be_suicide*).
- cluster 1 has the highest number of incidents where mostly are shootings. Clusters 3,4,5,6,10 are mostly involving shooting incidents.(*has_shooting*).
- drug involvement (*has_drug_involvement*) is significantly represented in cluster 0 and 7, implying a correlation between drug factors and the incidents in those two clusters.
- police involvement (*has_police_involved*) is notably higher in clusters 0, 7, possibly reflecting a correlation to drugs.
- domestic violence (*has_domestic_violence*) is present in cluster 3,6 the most, we can see that cluster 2 has none, which seems correlated to the fact that it holds no incidents with females.

Then, we try to obtain more information relative to these indicators: stronghold state party, last election winning party, number of incidents among year and if a state poverty is above the national mean. For 0,5 and 9 clusters incidents have as dominant (stronghold for state) party the Republican, all other clusters have the Democrats. Almost all clusters have 2017 record dominance, in general also between the years the proportions of clusters are equal, exceptions are cluster 4 and 8 with mostly 2014 incidents.

With an interactive map, present in our notebook at section 3.4.2 we can see at which cluster belongs the majority of records in each state. We have also a geographical view for each year and for every incidents with the cluster label and the indicators values. We may see on the west coast cluster 1 and 4 to be predominant, while on the east coast lots of cluster 9 and 10 incidents for 2014. In 2017 label 1 seems predominant for all states.

2.3.3 Validity

As we can see in the heat map in figure 6, the clustering analysis is valid. The figure shows how on the diagonal the various points forming squares. For this view we sampled randomly 100 samples and sort them accordingly to the cluster label. The records belonging to the same cluster have a blue dark intersection as they are closer. We can see how the third cluster seems the better defined, as having its internal point with a really low distance denoted by the deep dark blue. Not all clusters internal distances are as cluster 3, but given that the blocks are generally blue (close points) we can say that the clustering result is acceptable.

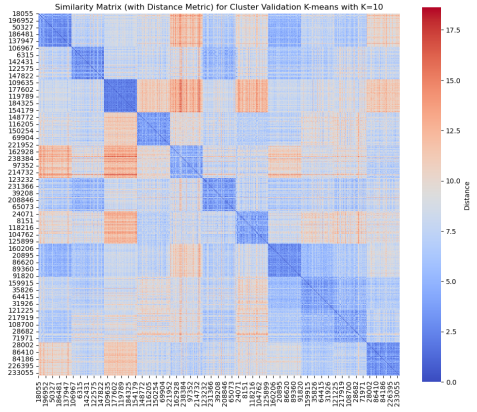


Figure 6: Heat map for K-Means validity.

2.4 DBSCAN

2.4.1 Parameters

We chose California because it is the state with the highest number of fatalities in the various accidents. To find the best EPS and best MINPTS, we used the graph that evaluates the distance to the 4-th nearest neighbor for each point sorted by its distance, and we note that the elbow is found in the range

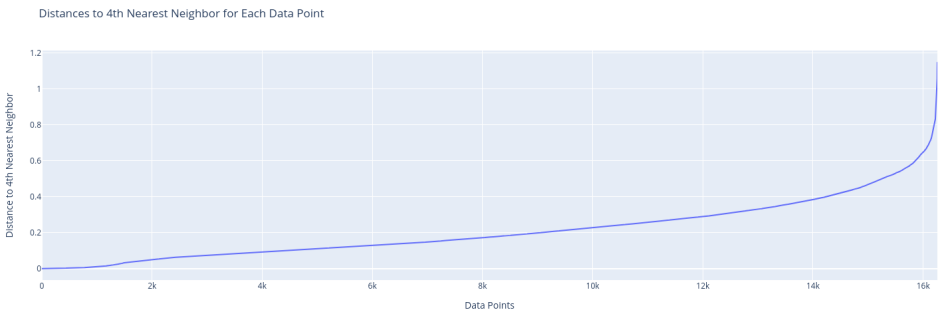


Figure 7: 4-th nearest neighbour

0.5 to 0.7 (figure 7). We have seen that the elbow for the 10-th nearest neighbor is always around 0.5 to 0.7 (figure 8). As rule of thumb, to choose the best parameters of DBSCAN we do a grid search

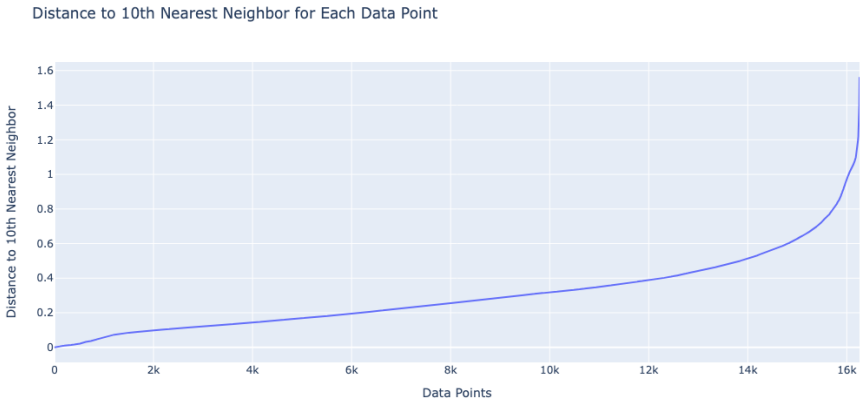


Figure 8: 10th nearest neighbour

through the value of EPS between 0.2 and 0.7 and consider as MINPTS between 3D and 4D, where D is the feature size of the dataset. To get the best parameters we go to calculate the Silhouette score and we got the best value 0.264 with EPS 0.7 and MINPTS 63. It seems that the search always falls on the highest EPS and the lowest minsamples value. The number of clusters obtained from this configuration is 25. Recall that -1 label values are used to label outliers.
2D and 3D PCA, t-SNE along with 2D Histogram Contour visualizations are ready available in the notebook.

2.4.2 Understanding the cluster data

As done for K-Means we also analyze Boolean identifiers for DBSCAN to understand what is happening within the various clusters. We have 25 clusters which makes it more difficult to detect patterns than K-Means with 8 clusters. In contrast to what happened with K-Means, here we find a strong unbalance of the data. We can see how incidents are distributed in clusters with the following list: 0: 1833, 1: 814, 2: 2130, 3: 786, 4: 948, 5: 719, 6: 2106, 7: 241, 8: 294, 9: 415, 10: 503, 11: 326, 12: 254, 13: 784, 14: 305, 15: 145, 16: 101, 17: 168, 18: 88, 19: 208, 20: 112, 21: 144, 22: 285, 23: 116, 24: 72. As was the case before, clusters that do not identify participants are a small minority (1 and 7). From a graphical analysis of the clusters we noticed that clusters 0, 8, 9, 13, 14, 16, 19, 20, 23 and 24 have records where there were deaths. In clusters 11, 15, 17, 18, 19, 20, 23 we noticed that there were deaths involving female individuals. In cluster 0 we find a significant number of suicide cases, followed by cluster 13.
In clusters 0 and 2 we find a high number of cases in which shooting occurred. Cluster 6 by the largest presence of non shooting incidents. We can also see that cluster 5 and 6 excel versus other clusters in the indicators of: drug involvement and criminal gang involvement. In particular cluster

6 holds the most school shootings incidents, incidents with police involvement, and domestic violence incidents. We found that in cluster 6 we have a high number of incidents related to 2017. Some cluster have predominantly 2014 incidents, the distribution of incidents by years is not homogeneous for each cluster. In cluster 6 we find that most of the incidents occurred in states below the poverty average. For all clusters the winning party is predominantly the Democrat for the congressional distrinct in which incidents occurred. But we note for clusters 5 7, 10, 13, 16, 21, 22,24 that the dominant party (stronghold) is Republican for all their incidents congressional districts. Again, we have reported geographic graphs, in this case related to California showing to which cluster the various incidents verified in this state belong. We note a large number of cluster 6 is predominant in 2017 in Los Angeles city, while in 2016 it seems to be cluster 0.

2.4.3 Validation

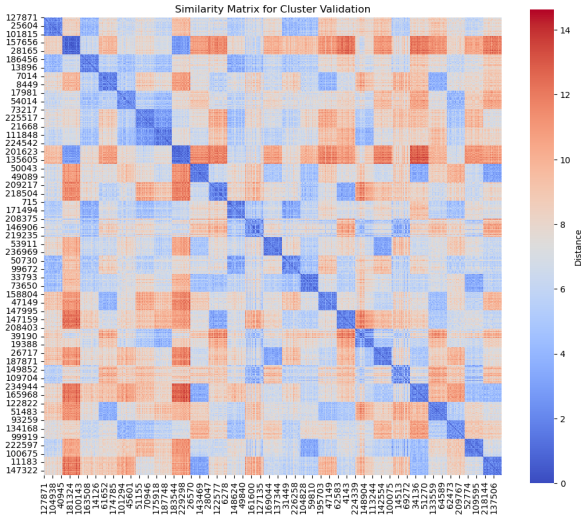
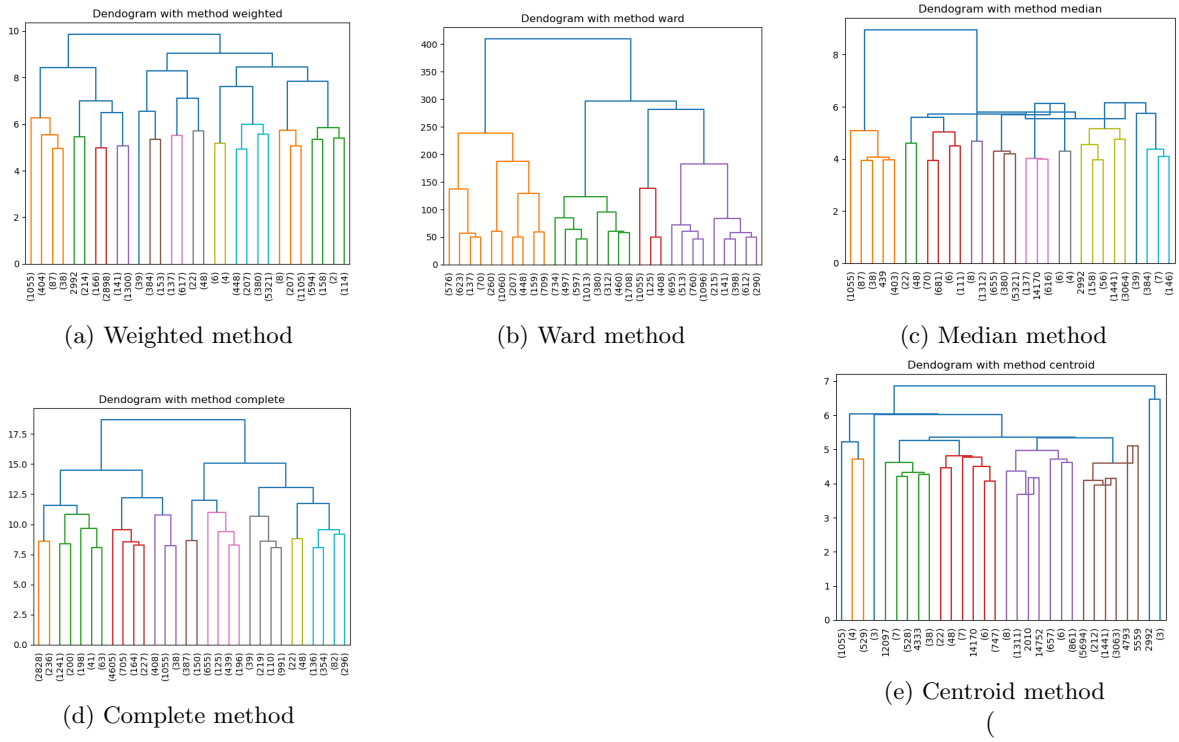


Figure 9: Heat map related to 100 randomly extracted samples from the dataset for DBSCAN validation.

It is possible to see that this clustering technique produced some valid clusters, since we find distinct zones on the heat map (figure 9). But we can see some problems, such as cluster 5 and 6 (starting to count from 0) not well separated, as their points are close. At the same time mostly all clusters are fine as most of their boxes are a deep blue.

2.5 Hierarchical clustering

Again, we use only the state of California for data clustering. We use the Canberra distance as a metric, which is particularly suitable for high dimensional data. The Dendrograms can be seen in figure 10. In the weighted method we find that the biggest jump is at start, this suggest that the initial clusters are quite separate, this feature we find in all the methods considered. We made the cutoff around 6, merging the nearest ones together. With the Ward method we have the biggest distance overall (as it uses the squared Euclidean distance), but with the most homogeneous jumps as it tends to create clusters with similar sizes, we cut the dendrogram around 250. The Median method the dendrogram exhibits a particular behaviour where the cluster at point '(39)' in the graph (on the right of the graph) is not assigned to any other clusters before the cutoff around 5, meaning that its points are very distant to the other clusters points. With the Complete method we cut the dendrogram around 11. The dendrogram appears well-structured, with clear delineations between clusters, which is characteristic of the complete linkage method Finally, with the centroid method we get the highest number of cluster merged, but also with the highest number of not joined clusters, the point '(1055)' is not joined with point '(O4)' before the cutoff, same with point '(3)' and point '2992', the cutoff is around 5.



(a) Weighted method

(b) Ward method

(c) Median method

(d) Complete method

(e) Centroid method

(

Figure 10: Dendrograms for various hierarchical clustering method

Method	k = 5	k = 7	k = 9
Ward	0.245	0.262	0.267
Single	0.008	-0.006	-0.015
Complete	0.232	0.240	0.274
Average	0.119	0.115	0.138

Table 1: Silhouette Score as k varies between methods

2.5.1 Parameters

Comparing the various methods by calculating the Silhouette Score for each AgglomerativeClustering execution using k = "number of clusters to find" equal to 5, 7 and 9 (Table 1). We decided to use the results with method Ward as it gave the best results for 2/3 n_clusters executions and number of clusters equal to 9 which yields the best Silhouette score for Ward. The absolute best Silhouette Score is instead given by complete with n_clusters = 9, not so far from the Ward result.

2.5.2 Understanding the cluster data

Clusters do not have equal distributed incidents with some having significant more: 0: 4553, 1: 1336, 2: 1479, 3: 3366, 4: 1320, 5: 938, 6: 1055, 7: 1287, 8: 924. Cluster 6 presents incidents with only non-identified participants and it is the only one having them. Cluster 5 presents incidents where there are only victims. In cluster 5 and 7 we find the majority of incidents to be due to suicides and in 5,7 8 we find most incidents to be due to shootings. While cluster 3 has a majority of non shooting incidents. The incidents with females are evenly distributed in the clusters with identified participants (so excluding 6), except cluster 7 having none.
Cluster 0 and 3 have a large majority of incidents occurring in states below the mean poverty percentage. Looking at the winning party over the years we can see that only cluster 7 involves more records in state were the stronghold is the Republican party. Again, we have reported geographic graphs, in this case relating to California showing to which cluster the various incidents that occurred in this state belong. We can see how cluster 0 and 3 alternate themselves in being predominant across the years. There are not many differences between cities across the same year.

2.5.3 Validity

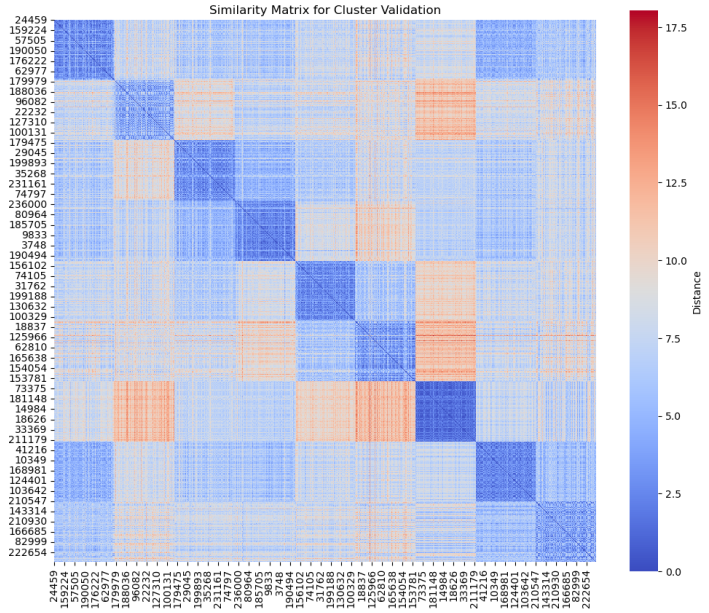


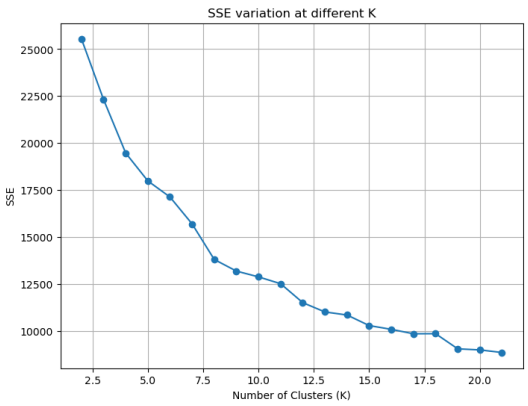
Figure 11: Heat map on distance related to 100 samples of hierarchical clustering (5 clusters) on California

We can see through the heat map (figure 11) that the various clusters are not well defined and some overlap occurs, especially in the cluster 2 and 3 (counting from 0). The most well define cluster is 6. In general this clustering result seems acceptable.

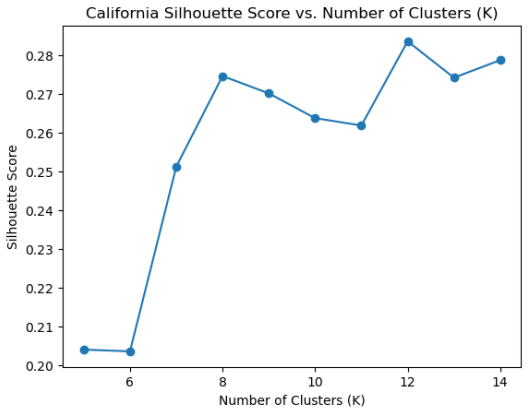
2.6 K-MEANS for California

2.6.1 Parameters

To have a more accurate comparisons in this section we performed clustering with K-MEANS on the state of California only, as done for the previous methods. Here we searched for the best K by calculating the SSE and Silhouette score. In the graphs 12a and 12b, there are the results. We choose as number of clusters (K) equal to 12, not so 11 as chosen for the whole dataset but close, having Silhouette 0.28 and SSE of 11511.05 with K=12.



(a) Sum of Squared Errors (SSE) for different numbers of clusters (k)



(b) Silhouette scores for different numbers of clusters (k)

Figure 12: Comparison of SSE and Silhouette scores for K-Means clustering on California dataset

2.6.2 Validity

In the figure 13 we can observe how the distance between the 100 extracted samples behaves in the heat map. It is observable that the clusters here are also well defined, so we can say that the validity is met. **Note that the structure of the matrix is very similar to the whole USA dataset one**

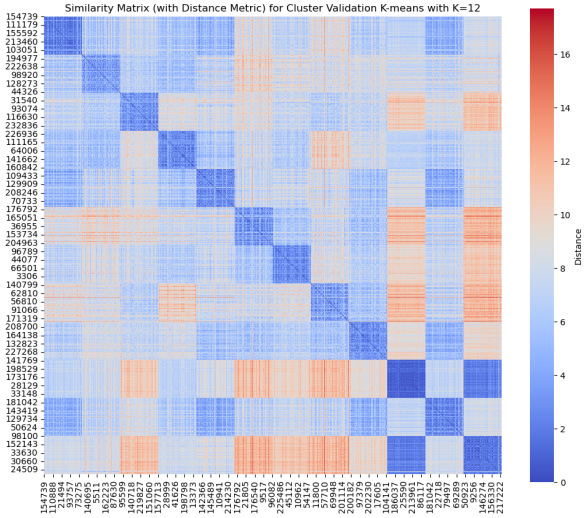


Figure 13: Heat map for K-MEANS for the state of California only.

2.7 Optional: MBSAS

2.7.1 Parameters

We have chosen as an optional clustering method the MBSAS method. This method is a sequential clustering method that does not require a predefined number of clusters. We apply MBSAS on the California and USA data. The only parameters that must be configured are the similarity threshold, which determines how close points should be to form a cluster, and the maximum number of clusters, which limits the total number of clusters formed.
To find the best parameters for the maximum number of clusters and the maximum distance that two points can have, a grid search was conducted.
The maximum distance that two points in a cluster can have is defined as the distance between point and it is indicated to be chosen according to the distribution of distances between the data points. It is indicated to choose a value around the histogram peak, in a way that in the cluster close points are kept, but not being to large of a distance that means merging points of different clusters. The histogram of distance can be seen in the notebook. We will do a grid search in the interval [1.2, 2.2]. On the USA dataset we had to do sampling to compute the pairwise distance, the resulting histogram was very similar to the California one so we omit it from this report.
The optimal parameters found were: max_clusters: 17 and threshold: 1.40, yielding a Silhouette score of 0.30 on California. The algorithm, in short, works by finding representative points in the data, by evaluating the points based on the similarity threshold. A point is chosen as a representative if no existing representative is within the threshold distance from it. The points by this implementation of MBSAS are considered in order of arrival, so in the order they are in the dataset. The algorithm computes the distances between all points and the representatives to create the initial clusters. At each iteration, it defines new representatives by finding the point that most centrally represents the points in its vicinity. The results for the USA data was almost identical using a max_clusters:17 and threshold:1.3 with a resulting Silhouette score of 0.311. *Again the PCA visualizations of the clustering results are on the notebook.* **We proceed to analyse the clustering result on the USA data.**

2.7.2 Understanding the cluster data on USA

We may see how some clusters have much more incidents assigned than other 0: 13809, 1: 20299, 2: 24418, 3: 30969, 4: 14734, 5: 11934, 6: 10632, 7: 13267, 8: 8745, 9: 16580, 10: 21105, 11: 5558, 12: 10946, 13: 5184, 14: 2976, 15: 3338, 16: 24634. There are not, as in K-Means analysis, indicators that are only true or only false for each clusters, giving less definition to our cluster results with respect to our labels for characterization. We can at best to see differences between clusters compute the ratio between true and false values for the indicators. Below in the text when we state "more incidents" we refer to incidents in proportion for a single cluster (True/False ratios on incidents of the cluster).

- **Identification and Gender:** cluster 5 and 6 have the lowest number of identified participants. Clusters 14 and 15 have more incidents with females.
- **Victims and Age:** cluster 10 has the most incidents with killed. Cluster 8 have the lowest victims. Also, cluster 8, 10 and 11 have the most incidents involving underage individuals.
- **Drugs, Gang Activity, Domestic Violence:** cluster 8 has the most drug-related incidents. Clusters 4 shows the higher incidents with gang activity and domestic violence.
- **Police and Violence:** cluster 14 seems to have the highest police involvement. Cluster 11 has more school shootings, while clusters 4 has fewest.

We can see in the notebook how cluster 3 is the most predominant for each state across the years. We notably have cluster 16 that is predominant for some states in some years.

2.7.3 Validity

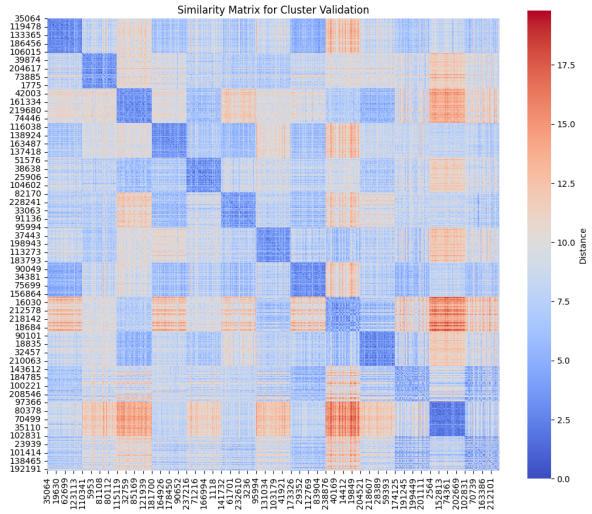


Figure 14: Heat map for the entire dataset using the MBSAS method.

We can see that clusters are generally somewhat defined, with some clusters not being very well defined and separated. (figure 14). The result are not great, with many not well separated clusters.

2.8 Comparison

We compared the various clustering methodologies, and the results obtained are shown in the tables 2 and 3. We see how for each algorithm the number of optimal clusters varies. However, with Silhouette score remains around the ranges of

$$0.24, 0.31$$

. We can also compared K-Means and MBSAS algorithms on the entire dataset. We did not run hierarchical clustering and DBSCAN since on the entire dataset as it was not required and it would be very computationally expensive. We tried DBSCAN on the whole dataset and it took too long to

Algorithm	K	Silhouette
K-Means	12	0.292
MBSAS	17	0.287
DBSCAN	26	0.261
Hierarchical	9	0.266

Table 2: Comparison on the California dataset only.

Algorithm	K	Silhouette
K-Means	11	0.240
MBSAS	17	0.311

Table 3: Comparison on entire dataset

run. Although, all the algorithms did not score well on the Silhouette score, where good clustering usually reaches a threshold of 0.5, we can say that all the algorithms are okay for an analysis on real data. There are some overlaps for all clustering techniques obtained clusters as we can see from the PCA visualizations.
We highlight some problems for DBSCAN as there one overlaps for two clusters, as seen on the distance (opposite of similarity) matrix. Additionally, the DBSCAN optimal solution that we found has 25 clusters, which is terrible for understandability by small teams (of 3 computer scientists).
On the contrary, we appreciate how the Hierarchical clustering while not giving a result much worse than others in terms of Silhoutte score, it needs less clusters (nine) than other methods.
We can also see that the results in terms of number of clusters and Silhouette change a lot for K-Means between the state of California and the entire U.S.We have more clusters for California than the USA, suggesting that the analysis on USA is broader and less specific for K-Means. MBSAS has very close Silhouette scores for the USA dataset and the California dataset, making it more stable than K-Means. K-Means has a much worse performance in terms of silhouette score on the USA dataset.
The last fact we want to emphasize, is the closeness in the optimal number of clusters found for K-Means and MBSAS (12 and 17), which are also the best in terms of Silhouette Score for California. We find a certain consistency in these numbers, indicating that there is an underlying structure in the data.
It must be noted that K-Means while not having a not "exceptional" Silhouette score as MBSAS on the USA dataset, it shows much more diverse clusters with respect to the indicators used to characterize them, while the MBSAS clustering results had less well defined cluster with respect to those, having the indicators values more balanced between clusters. In conclusion, while achieving a better Silhouette score with MBSAS, in our analysis with the target labels defined, we can say more with K-Means, without having to check the internal ratios of true/false inside the clusters. This may imply that not always going with the Clustering having the best Silhouette score is a good solution, but further check on the Similarity Matrix, graphically and by checking labels distribution inside the clusters are needed.

3 Task 3 - Classification

We are going to use the previously defined, but not yet used, Boolean indicators, we removed the incident characteristic features 1 and 2. We removed the label indicating the number of killed, we modified the dates by indicating only the month and day in order to better generalize to new data. To this end, we exclude the features: 'male_percentage_year','historical_incident_freq','ratio_unharmed_over_period_year', and 'percent_change'.
We performed at the discretization of ages into four categories, respectively: minor (up to 21), young_adult (up to 40), adult (up to 55), and senior (over 55). Applying one-hot-encoding to the features: party and stronghold_state_party, we merged multiple states into regions (Northeast, Midwest, South, and West) and coded with one-hot-encoding, because by coding individual states we would have gotten too many new columns.
Applying Random Forest we notice that the incidence of max and min average_participants features is zero, we proceeded to eliminate them.
Before proceeding with classification we proceeded to divide the dataset into two subgroups, train and test, respectively.

3.1 Decision Tree

The decision tree was found by searching for the best parameters using a grid search. This resulted as the best parameters 'criterion': 'entropy', 'max_depth': 15, 'min_samples_leaf': 3, 'min_samples_split': 4, 'splitter': 'best'. With an accuracy of 0.99. But having 15 levels, it is really difficult to understand the rule tree. We wanted to constrain the number of levels to 3, as to have less difficulty in understanding. We see that the accuracy of the model still remains very good, equals to 0.93. We can see that the

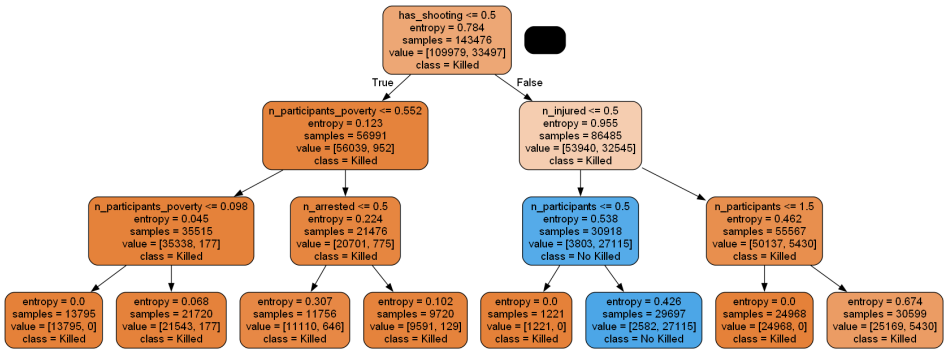


Figure 15: Derived rules tree

most important rule for the classification decision is the has_shooting indicator. We also observe that is classification by 'no killed' appears only in one leaf. From the confusion matrix it can be seen that

	Train Set
Accuracy	0.937
Precision	0.937
Recall	0.937
F1 Score	0.936
Support	0.913

Table 4: Performance metrics on train set

	Precision	Recall	F1 Score	Support
Killed	0.94	0.98	0.96	73320
No killed	0.88	0.81	0.86	22332
Accuracy			0.94	95652
macro avg	0.93	0.89	0.91	95652
weighted avg	0.94	0.94	0.94	95652

Table 5: Performance metric of test set

the number of false negatives is 4233 while the number of false positives is 1722. The results are good, in fact, almost all the records are correctly classified but the false negatives are more than the false positives.

3.2 Neural network

We used a neural network approach, applied the one hot encoding on the following Boolean variables: 'has_females', 'above_national_mean_poverty', 'has_shooting', 'has_drug_involvement', 'is_gang_related',

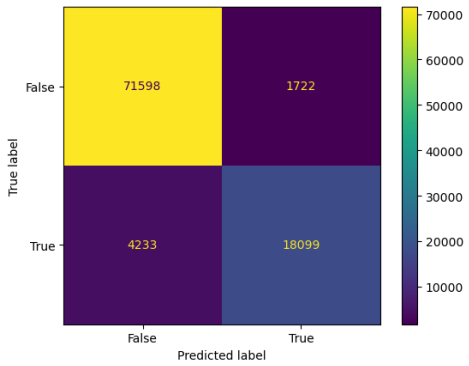


Figure 16: Confusion matrix over entire decision tree dataset

	Precision	Recall	F1 Score	Support
Killed	1.000	1.000	1.000	73320
No Killed	1.000	1.000	1.000	22332
Accuracy			1.000	95652
Macro Avg	1.000	1.000	1.000	95652
Weighted Avg	1.000	1.000	1.000	95652

Table 6: Obtained scores of Neural Networks

	Precision	Recall	F1 Score	Support
Killed	0.99	1.00	1.00	73320
No Killed	0.99	0.98	0.99	22332
Accuracy			0.99	95652
Macro Avg	0.99	0.99	0.99	95652
Weighted Avg	0.99	0.99	0.99	95652

Table 7: Scores obtained by ADABoost

'is_school_shooting', 'has_police_involved', 'has_domestic_violence', 'is_hate_crime', because the classification approach with neural networks prefers data encoded in this way. We used as a model a neural network with a hidden layer. We used a Sigmoidal function as activation function, Adamax as the optimizer, mean_square_error as the loss measure, and accuracy as the metric. We trained our model among 80 epochs and with batch size of 512. We obtained an accuracy of 100 percent already after 18 epochs, a loss value equal to loss:1.3394e-08 and val_loss equal to 1.3137e-08. We verified that the model was not overfitting by checking the weights of the various units, noting that the units were not saturated.

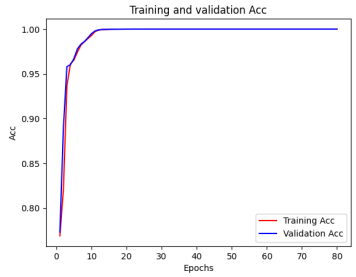


Figure 17: Training and Validation Accuracy

3.3 Adaboost

Since we found too good results, we decided to implement ADABoost as a boosting technique that allows a very weak classifier, which suffers from underfitting, to be able to adjust its parameters. The choice of the depth of the decision tree is set to 2, knowing that it is too small a parameter compared to the one observed previously. We then do a grid search and obtain that the best parameters for the decision tree are: 'estimator_min_samples_leaf': 500, 'estimator_min_samples_split': 500, 'learning_rate': 0.5, 'n_estimators': 70, 'random_state': 42, obtaining an accuracy of 0.9928 We can see how the accuracy jumps directly to a value close to 100% immediately after the first iteration of AdaBoost

3.4 Rule based classifier

We could not use the usual design set of 60% of the total because the running time was too long, so we reduced our dataset. To search for the best hyper-parameters, we performed a grid search with various configurations. We find the best configuration with the following parameters: 'dl_allowance': 128, 'k': 1, 'n_discretize_bins': 15, 'prune_size': 0.3 So we get an accuracy of 0.984, a precision of 0.980 and a recall 0.952. We can find the classification rules in our notebook. We can see that the most

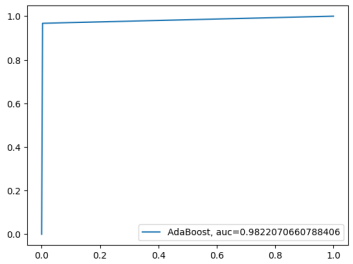


Figure 18: AdaBoost accuracy

frequently the rules defined involve features used by the Decision Tree based classifier. We list some interesting rules found, we observe that various attributes, including political party that won in the congressional district where the incident occurred, economic status, gender, and police involvement, significantly influence the classification of incidents involving deaths.

- has_shooting = True ˆ n_participants=>3.0 ˆ n_unharmed = 0.0 ˆ n_injured = 1.0 ˆ has_females = True ˆ has_police_involved=False ˆ above_national_mean_poverty = False]
- has_shooting = True ˆ n_injured = 0.0 ˆ n_participants = 2.0-3.0 ˆ region_South = True ˆ party_DEMOCRAT = True
- day = 0 ˆ povertyPercentage<0.21 ˆ has_shooting = True ˆ n_injured = 0.0 ˆ party_DEMOCRAT = False

3.5 End of Classification

We can say that as we saw from the rule, if there is a value has_shooting then there is a killing. Which seems as a natural consequence. In general we noticed more false positive than false negative. We can accept this bias because if there is a gun related incident its highly probable that there will be some victim. Also considering the high correlations between n_injured, n_unharmed, n_arrested and n_participants with n_killed we can justify why the models have this so high accuracy. We noticed even some surprising features that where used for the classification, for instances the decision_tree using poverty as can be seen in image 15.

4 Task 5 - Time Series Analysis

For this task we have defined as a score for each city, being *the number of incidents with drugs involved in a week*. We filtered out out cities with less than 30% of shooting incidents weekly with respect to the number of weeks in the dataset. We do not look at the number of drug related incidents, just that there were incidents recorded for that city. We obtained:

- 352 time series, 1 per city
- 208 points in the time series, 1 per week

4.1 clustering and motif/anomalies extraction

Before applying whole clustering on the uncompressed data series, we create a scaled version of our time series, using TimeSeriesScalerMeanVariance, setting the mean to 0 and the standard deviation to 1 for our time series. We limited the clusters to at most 6 in order to extract motifs and anomalies on those and make an in depth analysis, adding more would make the interpretability too much hard.

4.1.1 Whole clustering

We decided to use the Dynamic Time Warping metric as a distance, rather than the euclidean one. As it behaves better with Time Series, while it has the problem of being more computationally expensive.

We used TimeSeriesKMeans plotting the SSE for k from 2 to 15. We found the elbow point between K equal from 4 to 6. In this range we plotted the silhouette score and we settled for K=6. We obtained a silhouette score of 0.24, which indicates some degree of structure in the data but it is not great.
We can look at the cluster centers and the means of the clusters time series. Consider that one year is roughly 50 weeks. There is an interactive version of the those graphs at section 5.3.1.1 of the notebook on Time Series. Looking at the Mean Time Series in image 19. We see the 6 distinguished clusters

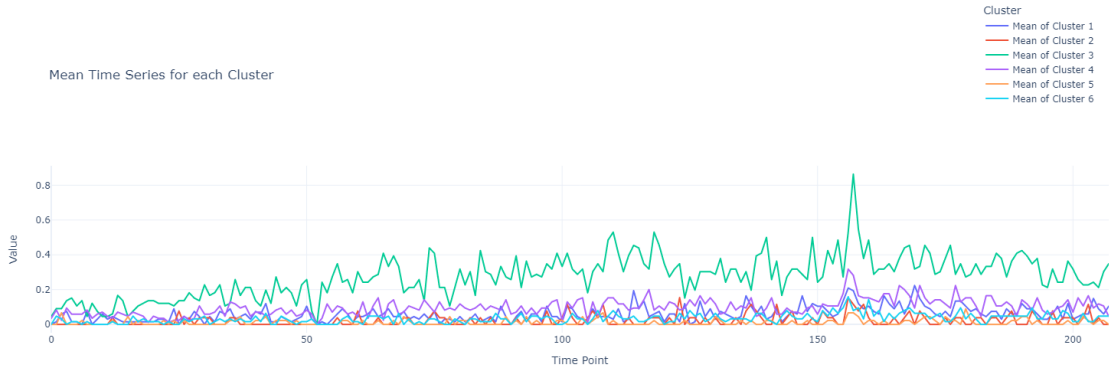


Figure 19: TimeSeriesKMeans result with K=6

and notice how cluster 6 has a bigger amplitude than all others, with bigger peaks, it suggests that the cities with the most drug related incidents weekly were clustered there. While it seems that cluster 2 and 5 have weeks where they are flat and others with some peaks. Clusters have different number of cities, with cluster 5 having 85, the most, while cluster 2 has only 26 cities. At section 5.3.1.1 we have an interactive USA plot that shows the cities colored with their cluster label, allowing to spot the cities of cluster 6. We may highlight that the city of Los Angeles in cluster 6. New York is in cluster 1. We can notice how the cities bordering with Mexico are almost all in cluster 6.

4.1.1.1 Motifs and Anomalies The plot of the Matrix Profile can be seen in the Notebook, while we show here for each mean time series of the clusters the motifs and anomalies. We computed the matrixprofile windows 8 weeks and 16 weeks. This allows to search bi-monthly pattern with short-term responses with the former and get get seasonal patterns (4 seasons of 4 months). We can see the result for our clustering in 20. We notice that with w=16 we capture much more motifs. There are cases where motifs repeat themselves, as can be seen in cluster 4 from point 24 to 48. With w=16 the motif seems well defined for all clusters, while for w=8, cluster 1 seems to have only 1 well defined motif and then cluster 3 and 4 have some repeating motif through the years. We found anomalies by searching discords in the matrix profile, with an exclusion zone of 2 points, this setting helps ensure that we do not mistakenly identify closely subsequent weeks as separate anomalies and seems appropriate given the datasets relatively small size of just over 200 points.
Looking at the anomalies in figure 21 We can see how for instance how in cluster 6, near the start of 2017 there was a peak followed by a big drop in drug related incidents (red anomaly), this could have been cause by a specific event or policy change that had a substantial impact on drug involved gun-related incidents.

4.1.2 Feature-based clustering

We created for each city time series a row with columns representing their dataseries, having Average, Stdev, Var, Median, 10th-25th-50th-75th-90th Percentile, Interquartile Range, Coefficient of Variation, Skewness measure and Kurtosis. We got a good silhouette score above 0.8 with K = 4 by using the K-Means algorithm over the dataset made from the features. We merged cluster 3 and 2 because the latter had only 1 city, we saw from the similarity matrix they were the more similar and silhouette score reduced a bit, but still above 0.8. It must be noticed how the now three cluster are greatly unbalanced. We have 258 cities in cluster 1, 83 in cluster 2 (2 merged with 3) and 11 in cluster 3 (cluster 4 before the post processing). Looking at the Mean Time Series in image 22 we can see how

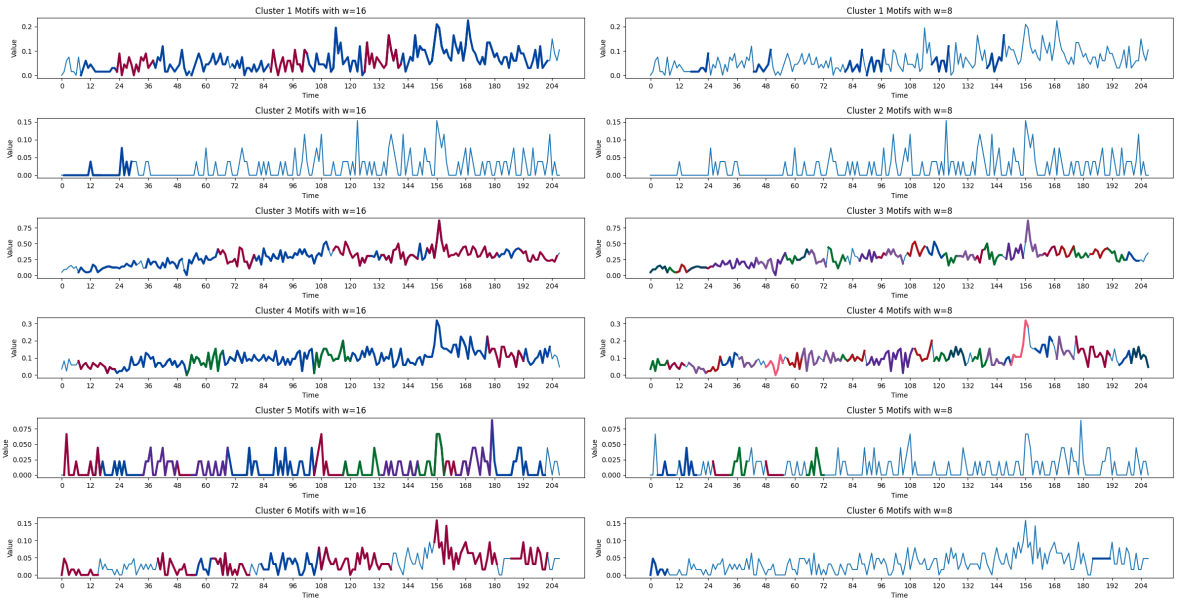


Figure 20: Time Series clustering Motifs Spanning 2013 to 2017 with 5 Distinct clusters. Window sizes of 16 (4 months) and 8 (2 months). The x-axis labels denote quarterly intervals of 12 weeks each

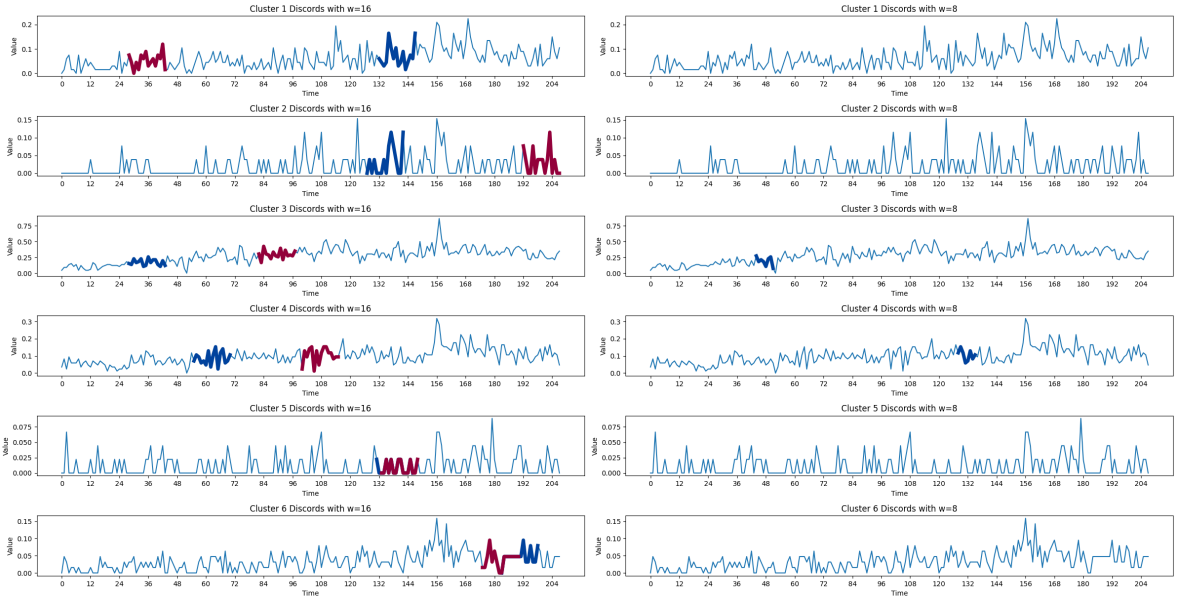


Figure 21: Time Series clustering Anomalies seen with Matrix profile discords Spanning 2013 to 2017 with 5 Distinct clusters. Window sizes of 16 (4 months) and 8 (2 months). The x-axis labels denote quarterly intervals of 12 weeks each.

the cities in cluster 3 had big increase over the years of drug incidents, while they had the least in 2013. We may note how, here Los Angeles and New York are on the same cluster, differently than the whole clustering results. We can see cluster 3 from the interactive map having cities only on the East Coast, such as San Antonio and Baltimore.

4.1.2.1 Motifs and Anomalies We did the analysis similarly as the whole clustering. We can see the motifs in figure 20. For w=16 we can see how cluster 1 has two motifs that alternate between the years. Cluster 3 has a motif that repeats for all years, seeming to have a stable situation. It may be possible given this seasonality even to try and forecast future drug related incidents, but this may

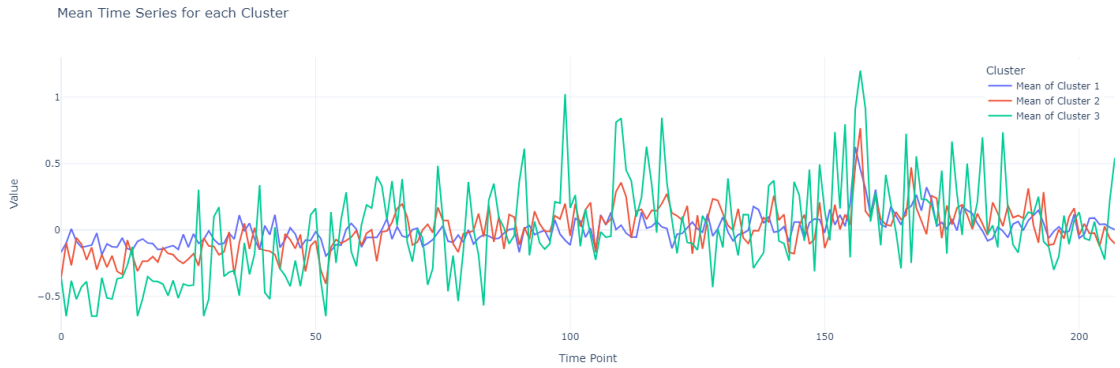


Figure 22: K-Means on time series features result with 3 clusters.

hold only as long as the socioeconomic situation is kept the same.

For anomalies, we observe in Figure 23 that there are more anomalies with $w = 8$. Specifically, for

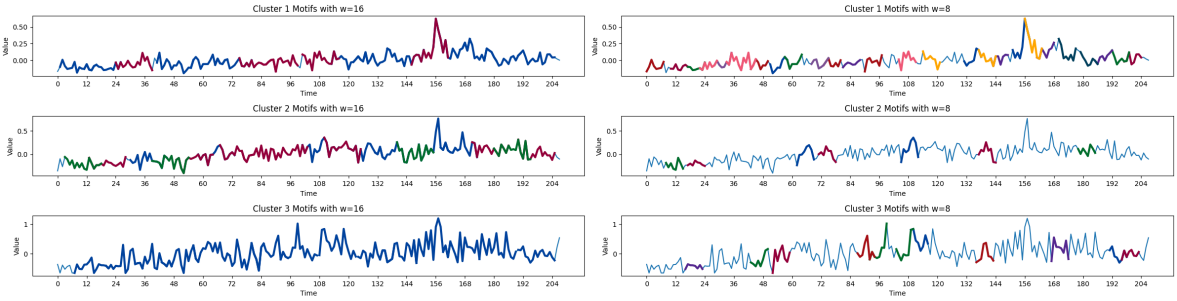


Figure 23: Time Series features clustering Motifs Spanning 2013 to 2017 with 3 time series features clusters. Window sizes of 16 (4 months) and 8 (2 months). The x-axis labels denote quarterly intervals of 12 weeks each.

cluster 1 with $w = 8$, numerous anomalies appear from point 84 to point 132. This suggests that in 2015 and 2016, some event or policy may have impacted the cities within this cluster, which comprise the majority of the cities under consideration. Focusing on point 100, corresponding to around November 2015, we note a spike in drug-related incidents. This may be speculatively linked to the ruling by the United States Sentencing Commission on October 30, allowing nonviolent drug offenders to begin being released from prison, with the first wave totaling nearly 6,000 individuals, as stated in Wikipedia.

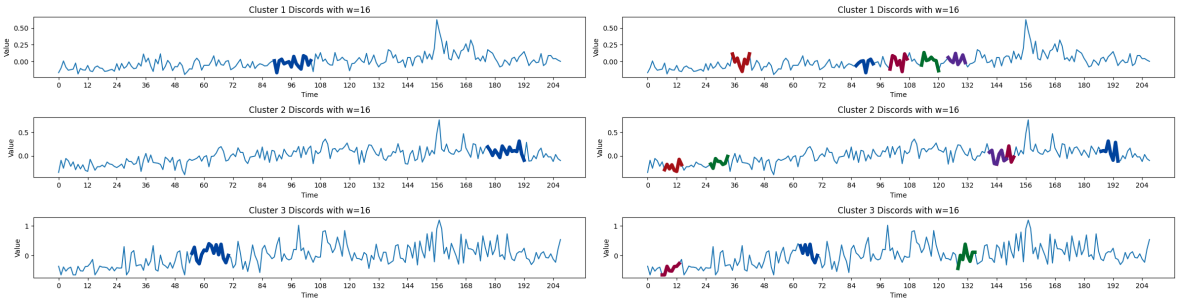


Figure 24: Time Series features clustering seen with matrix profile discords Spanning 2013 to 2017 with 3 time series features clusters. Window sizes of 16 (4 months) and 8 (2 months). The x-axis labels denote quarterly intervals of 12 weeks each.

4.2 Compression-based clustering

We used Normalized Compression Distance to infer the similarity of two distance based on how well they are compress together. The distance is computed by encoding the series with UTF-8 characters, constructing a matrix of distances based on this and then, on top of this matrix we applied DBSCAN. It allows to identify clusters of similar time series based on the distances computed. We obtained a silhouette score of 0.44, but with the problem that we got three clusters, where cluster 1 had 350 cities, while cluster 1 and 2 and a city each. We also attempted to compress the data points of the time series using PiecewiseAggregateApproximation, creating time series of 64 points instead of the original 208. Using TimeSeriesKMeans with the dtw distance metric on top of those compressed time series, we obtained with K=6 a Silhouette score of 0.06, very bad. We concluded that, having 208 points for each city, those compression techniques are not useful for our analysis, as the points are already few and it appears that the process of compressing the time series may have led to a loss of distinctive features that are crucial for identifying clusters within the time series.

5 Shapelets classification

In doing this task we faced many difficulties. If we filter the dataset for the clustering with the 30% of less eventful cities being removed, we did not get any city without killed in the years of the dataset. If keeping the whole dataset cities, we had the problem that too many cities with 0 events would be kept. So we tried to filter out the least we could, filtering the 1% of cities with least incidents. The classification is challenging as we have an unbalanced classification, reported in the table 8. We expect the model to have difficulties with No Killed labels, as there are too few instances. To better train the

has_killed	Count
Killed	4509
No Killed	1005

Table 8

model, we resorted using the SMOTE data augmentation, oversampling the minority class over the training set, obtained by applying a 70-30 split. We must highlight that the SMOTE augmentation may not be the best oversampling solution for time series as we lose the intricacy of real world data with the inclusion of made up records. We used the ShapeletModel from tslearn.shapelets using the parameters:

- l=0.25: to make the model look for time series long at least 51 points (the 25% of 205 points), so at least a year.
- r=4 to allow the model to try different shapelets sizes bigger than 51.

Our model was able to obtain an accuracy on the training set of 0.56, while the accuracy on the test set is 0.44. We have seen that the precision is of 0.82 for the Killed label on the test set, while it is 0.18 on the No Killed label. Considering that Killed recall is 0.41, we can say that the model is highlty biased in giving as outtput Killed. The shapelets obtained can be seen in figure 25.

5.1 Alternatives undersampling Killed and using score: n_injured

By only undersampling in the training set the Killed feature to have equal instances for both classes without using SMOTE, we still got a model that does a random guess, with Training accuracy 0.5 and test accuracy 0.48. Here again the precision for Killed (0.55) is higher than No Killed (0.43), witth a Killed recall of 0.45, showing that this model is biased to output Killed. Since it may be possible that the incidents involving drugs do not correlate well with whether a person has been killed in a city, we attempt a different approach. With 'n_injured' for each city and each week, as we know that this feature is highly correlated with n_killed. In this case, we achieved a Training set accuracy of 0.61 but a Test set accuracy of 0.52. The real improvements are marginal. The precision for 'No Killed' is 0.17, while for 'Killed' is 0.81, which has seen also an improvement in recall at 0.54.

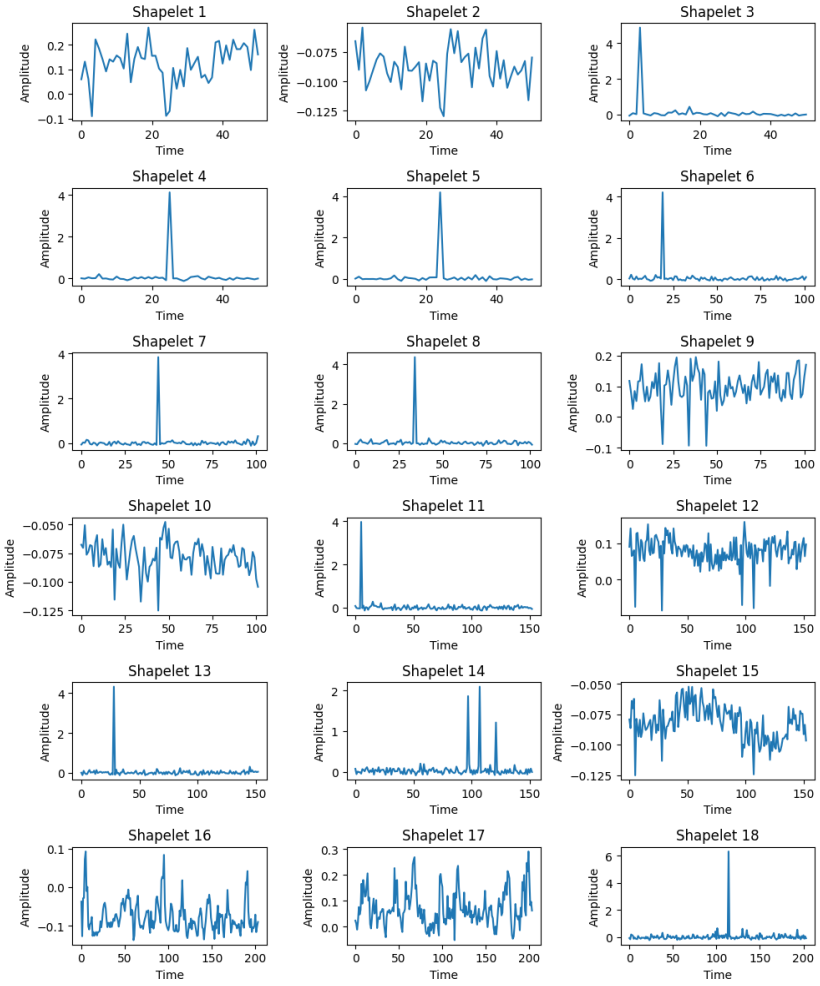


Figure 25: Shapelets obtained by ShapletModel with l=0.25 and r=4 over drug related gun-incidents time series of 207 points (weeks) per city (5514 cities)

5.2 Conclusions on Shapelets classification

We conclude that determining if there have been shooting victims in a city over the dataset's five-year span is a challenging task. Generally, cities without victims are few, and for those, gun-related incidents are also less frequent, making it particularly difficult to learn the 'No Killed' case. Given the totality of the five years and the free circulation of guns in the USA, it is perhaps understandable why classification models are biased towards the 'Killed' label. This tendency of preferring the 'Killed' label was observed not just here, but also in the classification section of task 3. It is reasonable to conclude that with the many gun-related incidents happening in the USA, a city might experience at least one death over a five-year period.
A city with no deaths would mean that has few gun related incidents and we may notice how unfortunately the model was unable to make this assumption. Furthermore we have tried to train a model without SMOTE/under-sampling and it resulted in a model outputting only the Killed label. In conclusion without the bias of having more cities with killings than without (or viceversa), the classification with shapelets will output a random guess, with said bias it only output that there were killed, making this task unfeasible.

6 Conclusions

We have found many correlations and patterns. We have found clusters with groupings that put together incidents with different characteristics. **We can say that multiple patterns definitely**

exists and have exposed in this report, but they amount to an overview. Further specific studies hare required, possibly in accordance with domain experts (criminologists or sociologists).