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

Candidate:

Paterniti Barbino Niko
Mat:638257
Nicassio Gaetano
Mat:658073
Moramarco Roberto Claudio
Mat:666140

ANNO ACCADEMICO 2023/2024

Indice

Capitolo 1

Time Series Analysis

For the time series analysis we considered only incidents that happened in the years [2014,2015,2016,2017] and we extracted a time series for each city, computing for each week of the 4 years a score with the following formula:

$$Severityscore = n\_arrested + n\_injured + n\_killed$$

$$Score = Severityscore \times (2 + \frac{n\_killed}{n\_participants}) - n\_unharmed \times 0.25$$

As we can see the final score used to represent the time series is based on a severity score indicating the severity of the incident, giving more importance to the killed rate w.r.t the number of participants in the incident. We also decided to utilize the n_unharmed information since usually if a participant is unharmed there is less probability of someone getting killed so we gave it a negative weight of 0.25.

We initially considered only cities with a number of weeks with incidents greater than 15% but since we found out that there were still a lot of time serieses with many NaN values (which we filled with a score of 0), we decided to increase the threshold to 25%.

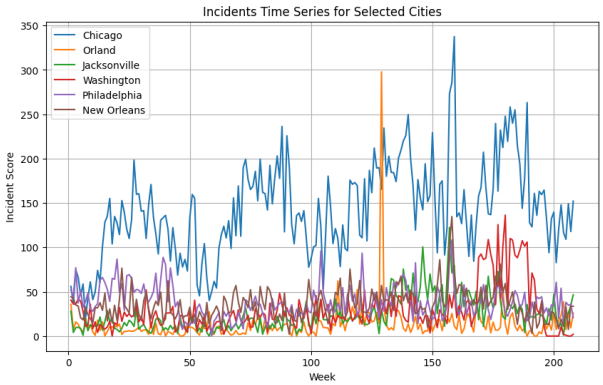

Figura 1.1: Example Time Series

1.1 Trend detection and removal

1.1.1 Detrending

We used the Augmented Dickey-Fuller, a statistical test to get the non-stationary time series.The Augmented Dickey-Fuller (ADF) test is a statistical assessment employed to determine the stationarity of a time series dataset. Stationarity indicates that the statistical properties of the time series remain constant over time. In our case if the p-value is higher than 0.05, then the time series is non-stationary. We applied the test to each time series in the dataset and recorded the indices of thetime series found to be non-stationary. No time series have been identified as non-stationary.

1.1.2 Denoising

To find the best sliding window for the noise we used the mean of the sums of absolute differences (SAD) between the original timeseries and the smoothed one, then we resorted to the Knee method to choose the best window that "doesn't smooth too much" the original timeseries, thus selecting the smallest window size value after the Knee.Using the Knee method we chose 5 as sliding window, as it can be seen in the following figure, we show an example of a denoised timeseries.

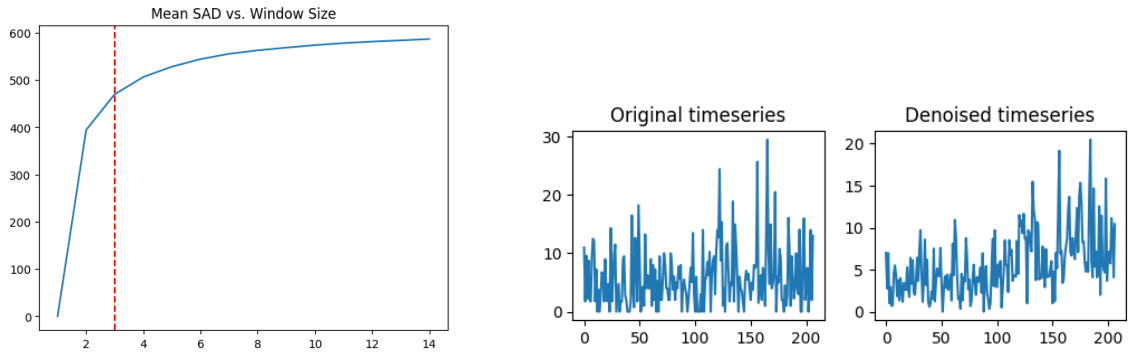

Figura 1.2: Example Time Series

1.1.3 Scaling

As for scaling the timeseries we resorted to the TimeSeriesScalerMeanVariance which specifically normalizes time series data using a mean and variance approach. It subtracts the mean and divides by the standard deviation for each time series independently.

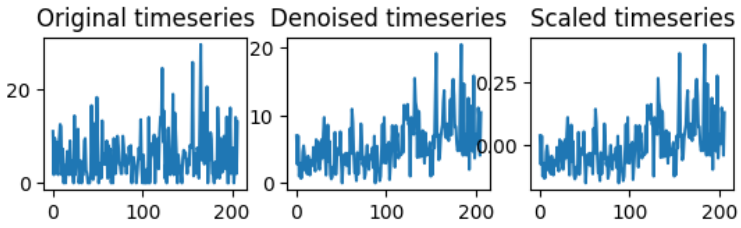

Figura 1.3: Denoising

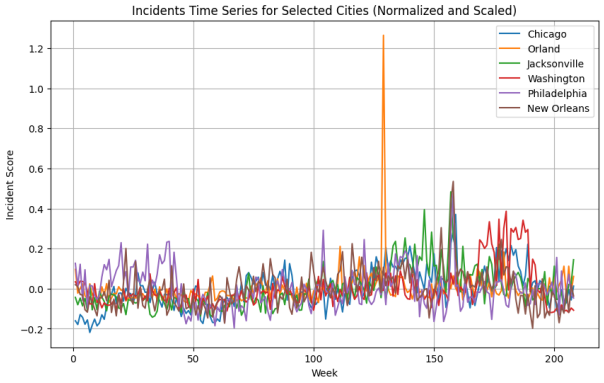

Figura 1.4: Example of Normalized timeseries

1.2 Clustering

The goal of this task is grouping and analyzing similar cities through the use of the created time series based on the defined score. We clusterized the timeseries using 2 different methods: Shape based and Feature based

1.2.1 Shape based

In order to select the best k for the K-means algorithm, we initially used the Euclidian metric, and then Dynamic Time Warping variant to see if the latter would lead to better results since, even if it is computationally very expensive, it has several advantages:

- Flexible and can handle time series that have different lengths or variations in speed.
- Robust to noise and small temporal distortions in time series data.
- Effective in capturing phase shifts or temporal misalignments in time series
- can handle time series with irregular sampling rates.
- Effective in identifying similar patterns within time series,

, The results showed that the results are balanced w.r.t the cluster size:

- Euclidian metric (K = 6): [36,83,57,91,96,40]
- DTW (K = 6): [65,57,135,47,48,51]

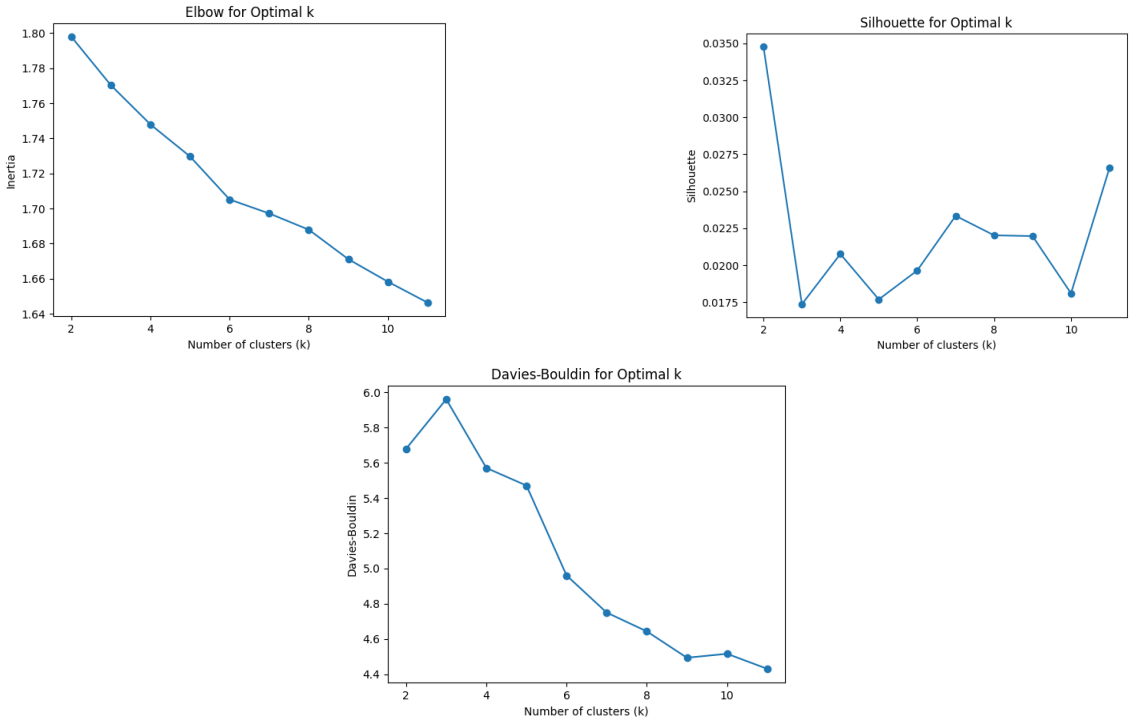

Figura 1.5: Metrics to identify best K

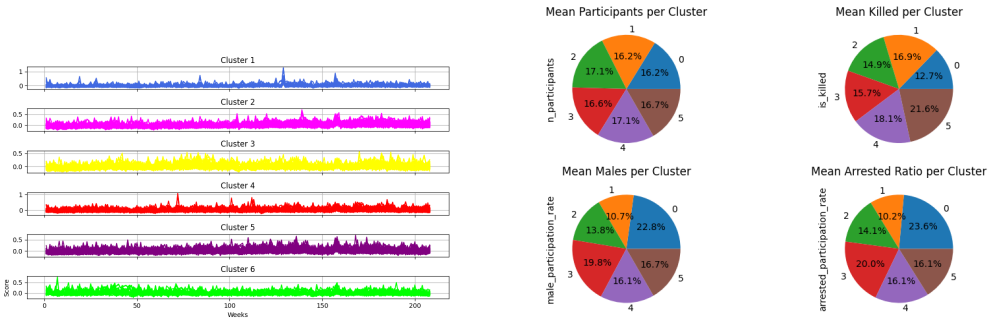

Figura 1.6: K-Means results

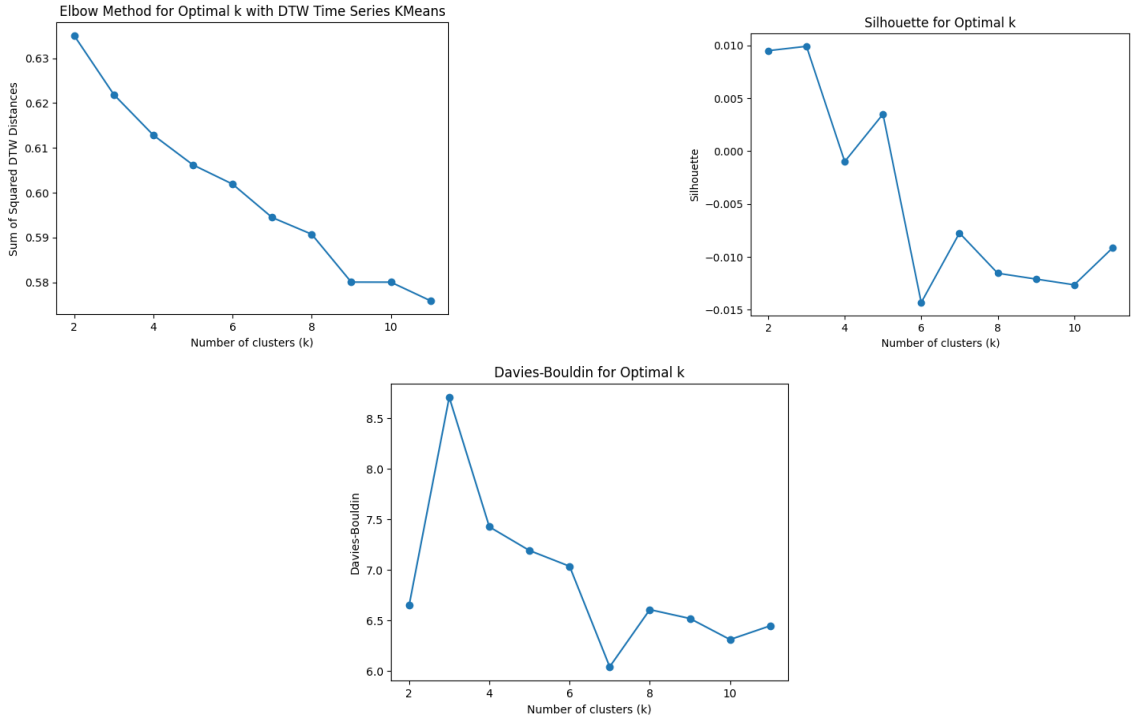

Figura 1.7: Metrics to identify best K

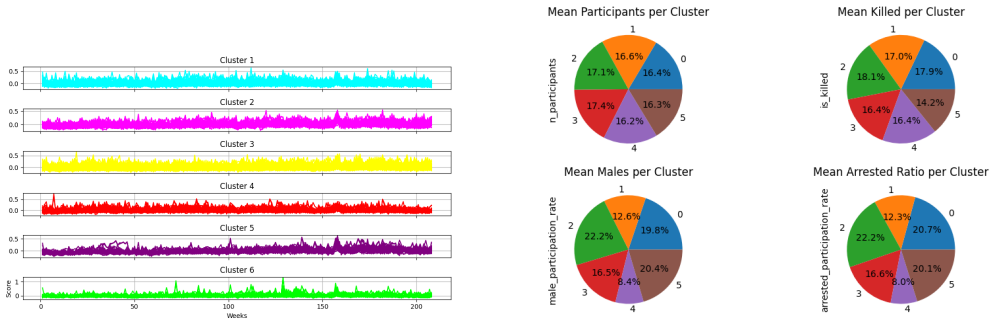

Figura 1.8: DTW results

1.2.2 Feature based

K-means

For the Feature based clusters we defined a set of features to extract from the timeseries to then perform the standard clustering using tabular data. We extracted from each time series: *Average, Standard deviation, Variance, Median, 10th percentile, 25th percentile, 50th percentile, 75th percentile, 90th percentile, interquartile range and the coefficient of variation.*
We selected k = 4 clusters with size respectively [35,3,4,361] demonstrating how the results are imbalanced in terms of cluster size having in fact one major cluster and three small ones.

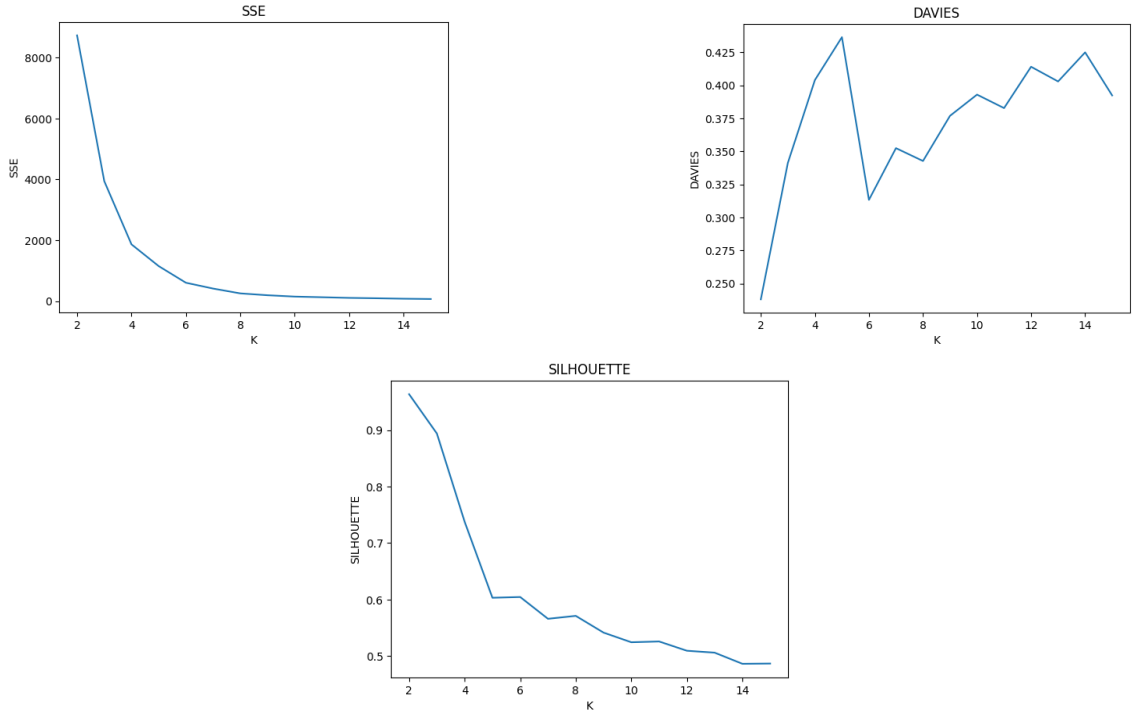

Figura 1.9: Metrics to identify best K

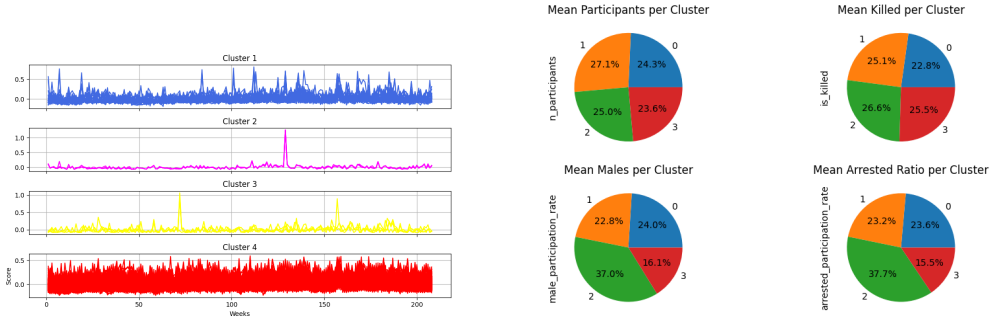

Figura 1.10: Feature Based results

Hierachical

For the Hierarchical clustering we tested all linkage methods and with both the standard scaler and minmax scaler ending up using the "Ward" method.

```
hierarchical_clustering_test(X_std, n_clusters=2,
    linkage_method='ward')
hierarchical_clustering_test(X_minmax, n_clusters=2,
    linkage_method='ward')
hierarchical_clustering_test(X_minmax, n_clusters=3,
    linkage_method='ward')
```

Listing 1.1: Clustering test with different configurations

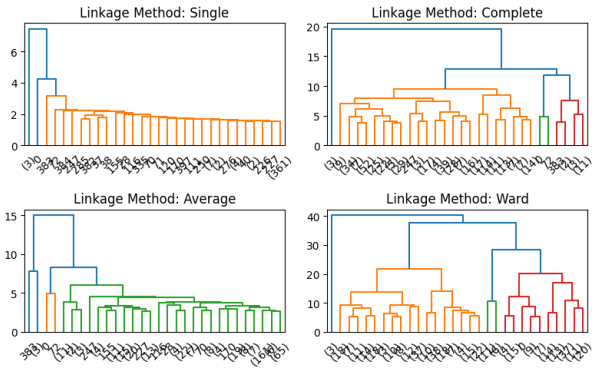

Figura 1.11: Hierarchical Clustering

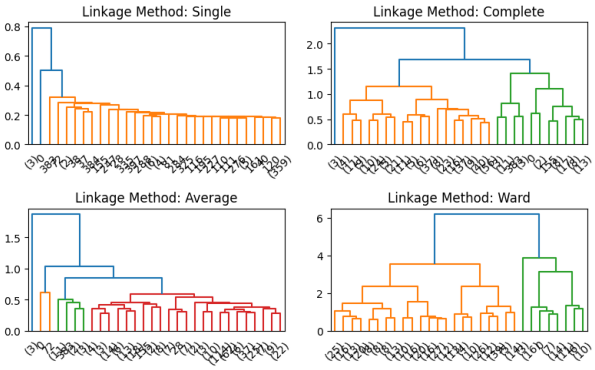

Figura 1.12: Hierarchical Clustering

1.2.3 Final cluster

Of the methods used for clustering, the one which produced the best results was the Shape-Based K-Means with Euclidean metric as it separated the 6 clusters in a balanced way, the Shape-Based K-Means with DTW metrics also showed comparable results but due to it being computationally expensive we decided to select the first as the best. So we decided to analyze the clusters by using a geoplot providing a map to look for interesting insights, we found that:

- The mean values for each cluster are smoothed over a common average
- The bigger cities are split over multiples counties and part of their data might have been filtered out.

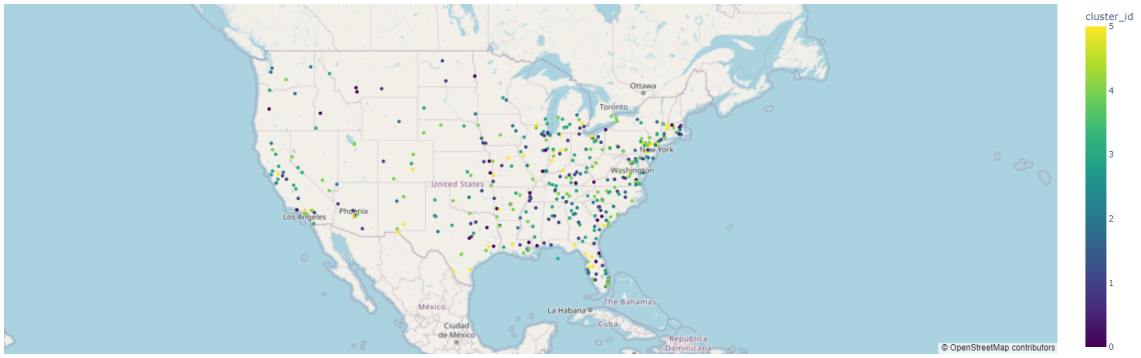

Figura 1.13: GeoPlot Clusters

1.3 Motifs and Anomalies

In order to find motifs(repeating patterns within a time series) and anomalies(regions where the time series significantly deviates from typical behavior) in each cluster we selected the time series most similar to the shape of the cluster centroid, resulting in a candidate for each cluster.

We then constructed the matrix profile(a vector that contains information about the similarity between subsequences of a given time series) by computing the Euclidean distance between each subsequence and its nearest neighbor within a window of 4 week(a month),

By analyzing the matrix profile, we identified the following motifs and anomalies:

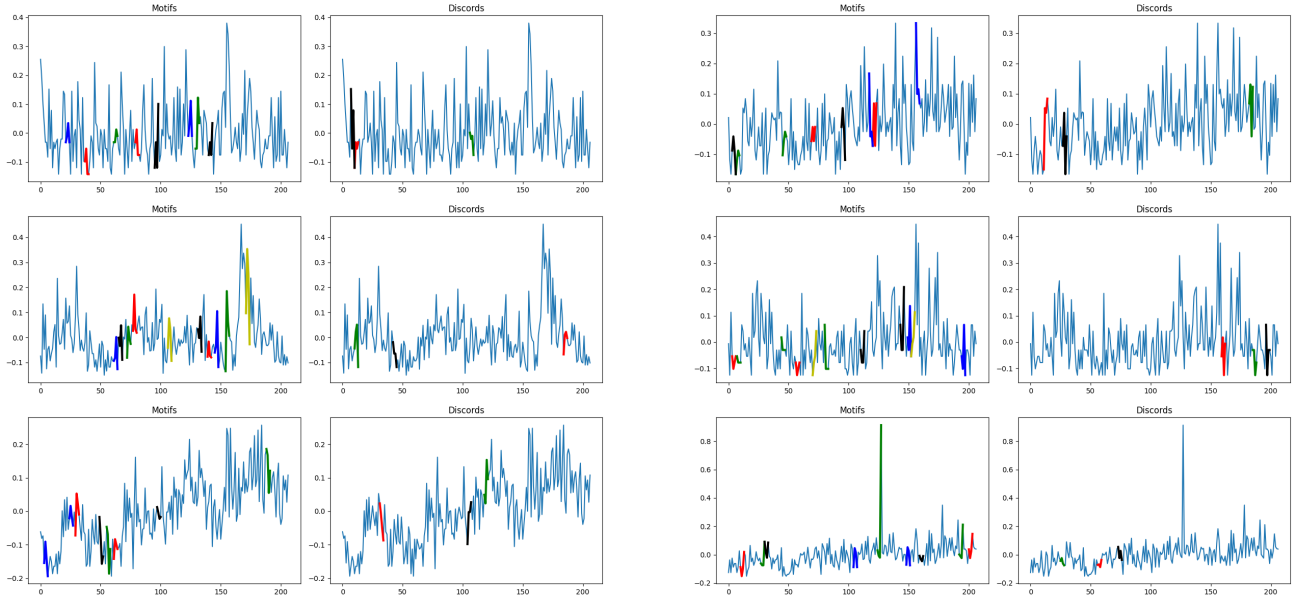

Figura 1.14: Motifs and Anomalies

1.4 Shapelets

For this task the goal was to extract the shapelet according to the class of the binary variable 'isKilled'.
Since the the score used so far was based on the binary variable 'isKilled' which is True if n_killed > 0. The first thing we did was calculating a new score, this time based on a 'youth involvement index'

$$YouthInvolvement = teen\_participation\_rate + child\_participation\_rate))$$

$$Score = 2 \times YouthInvolvement + [n\_participants - (YouthInvolvement)]$$

As we can see, we gave more importance to the involvement of young people. In order to find the number and size of shapelets within the time series we used Grabocka heuristics returning 4 as optimal size and 6 shapelets. In our analysis, we employed the LearningShapelets method from the tslearn library to train a Shapelet model on time series data. The dataset was partitioned into training and testing sets using an 80%-20% train-test split. Following the training process, the model demonstrated an accuracy of 0.67 on the test set. The results are shown in the following plots:

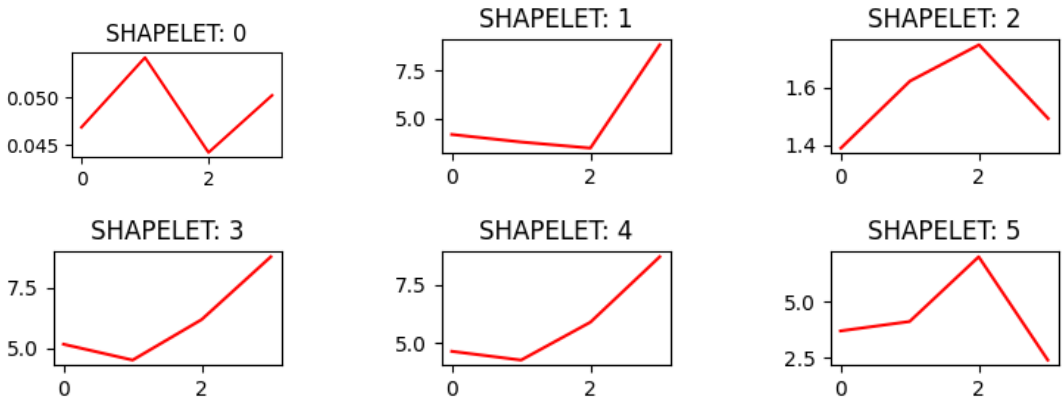

Figura 1.15: Shapelets

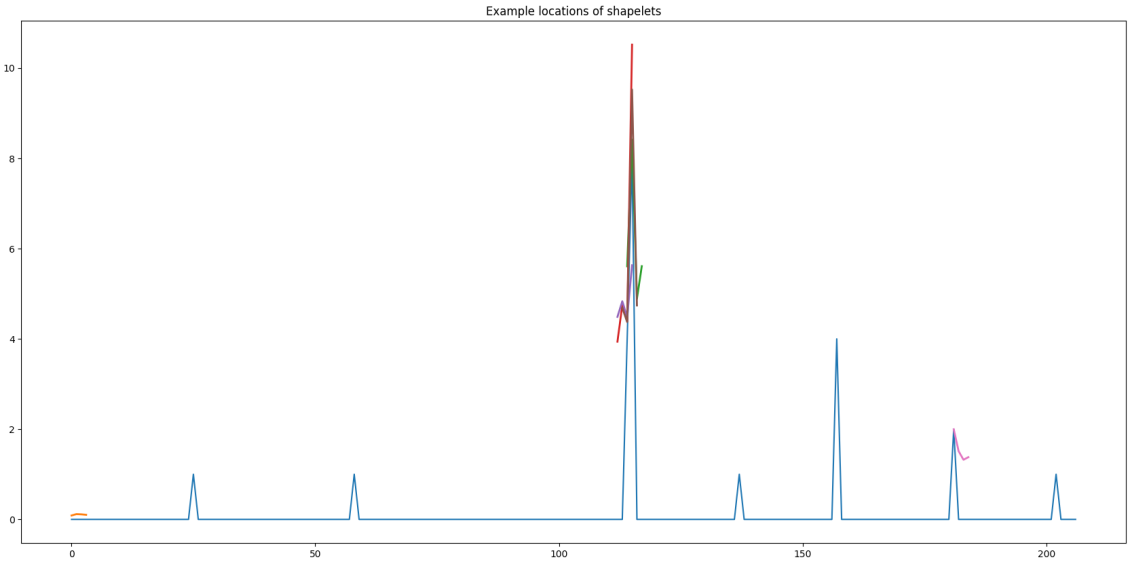

Figura 1.16: Example of identified shapelets