London Crime: Data Understanding report

Maddalena Amendola Giuliano Cornacchia Mario Leonardo Salinas

Contents

1 Dataset descritption

The original dataset consists of 13.490.604 records, each record is described by 7 variables:

- The `lsoa_code` is a code that identifies the 4835 different LSOAs (Lower Layer Super Output Area), namely a geographic area inside London.
- The `borough` is a nominal variable that identifies one of the 33 boroughs present in London, each of them contains a variable number of LSOAs.
- The `major_category` is a categorical attribute that specifies the category of the crime committed; there are 9 major categories: `Theft and Handling`, `Violence Against the Person`, `Criminal Damage`, `Robbery`, `Burglary`, `Other Notifiable Offences`, `Drugs`, `Sexual Offences`, `Fraud or Forgery`.
- The `minor_category` is a categorical attribute that specifies precisely the crime committed; there are 32 minor categories.
- The `year` is a numerical attribute that indicates the year when the crime was committed; year varies in [2008,2016].
- The `month` is a numerical attribute that defines in which month the crime was committed.
- The `value` is a numerical attribute that indicates how many crimes of a certain type in a certain period in a LSOA were committed.

Reviewing the data, there aren't null values or missing ones. Furthermore we choose to keep all columns in order to be able to analyze from different perspectives the crime distribution; but comes out that approximately the 70% of the records has a value equal to zero.

Considering a generic line of the dataset, for each lsoa_code, minor_category, month and year the value attribute specifies how many crimes of that kind were committed, even if there isnt any crime that was committed.

Since a `value` = 0 gives no contribution for the computation of the standard statistical measures, the dataset was filtered deleting all the rows where the value was equal to 0. This operation leads to an important memory space saving: we started with about 13M records and, after the filtering, we ended up with 3M non-zero records (the 23% of the real size). This will also allowed a faster of computations.

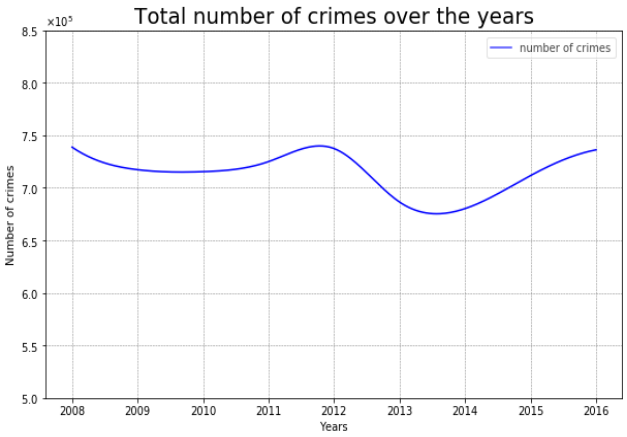
2 Exploratory Analysis

2.1 Features Distributions

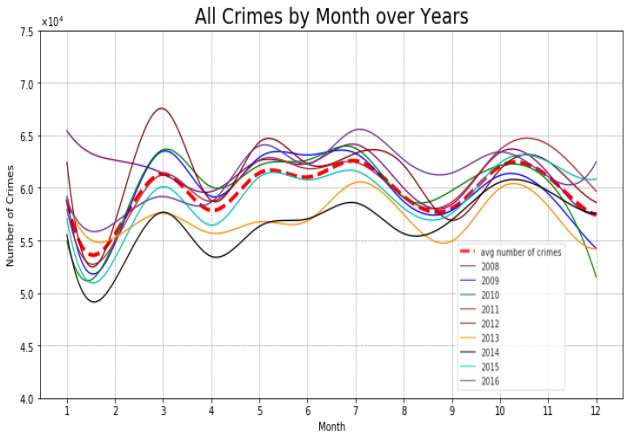
We start our analysis by having a look at the distribution of the number of crimes over the years and months.

The line graph 1a shows that the number of crimes is equally distributed between the years with the peaks in 2008, 2012 and 2016; while in 2013 and 2014 there were fewer crimes. In particular, the maximum number of crimes is in 2008 with a total of 738k crimes, while the lowest value is in 2014 with 680k crimes. The trend can be defined as constant except for small fluctuations of $\pm 0.1\%$.

In figure 1b it is possible to compare the monthly trend of each crime in each "year" with the average trend. The observations from figure 1a are confirmed, the years 2013 and 2014 are below the average while 2008, 2012(with a critical peak in March), 2016 are above. The total number of crimes varies each month compared to its average at most by 5k / 10k. Finally, both the average number of crimes line and the ones referring to the years show some common behavior:there are peaks in March, May, July and October, while February is the month with less crimes. Up to now we know that the number of crimes is



(a) Total number of crimes during the years.



(b) Total number of crimes over months.

almost constant over the years and months less than small fluctuations; the next step is to analyze how the crimes are distributed over the years in terms of major category.

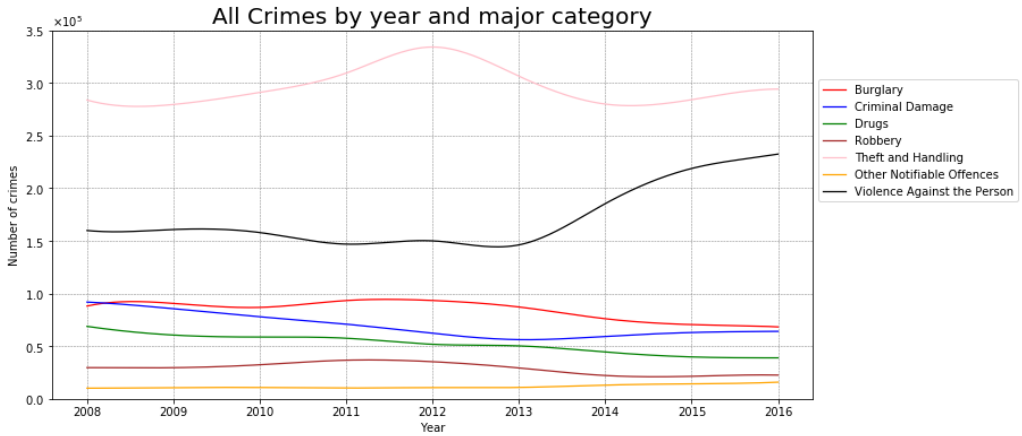


Figure 1: Total number of crimes during the years.

We decided not to show the lines related to Fraud and Forgery and Sexual Offences since the dataset contains only 2008 data for these categories. However we can observe that lines related to different crimes almost never cross each other; this means that the fraction of felony with a specific category over the total number of crimes is constant in time. Other Notifiable Offences, Robbery, Criminal Damage, Burglary don't show particular fluctuations, while Drugs crime are decreasing linearly. Theft and Handling has a peak in the year 2012; maybe due to the XXX Olympic Games hosted by the capital. More concerning is the Violence Against the Person rise since 2013; its value doubles in two years.

We can now visualize the proportions between the major crime's categories without distinguishing by year, since we've already seen that these fractions remain constant during the years.

The donut chart in figure 2 shows that the two most frequent major categories are Theft and Handling and Violence Against the Person, which respectively represents the 41% and 24% of the total crimes; Burglary, Criminal Damage and Drugs are the second most frequent crimes with percentages between, while the other categories have percentages ≤ 4, thus really rare.

We have also analyzed what are the most influential minor category in each major category and discovered that each major category has at most two dominant minor categories.

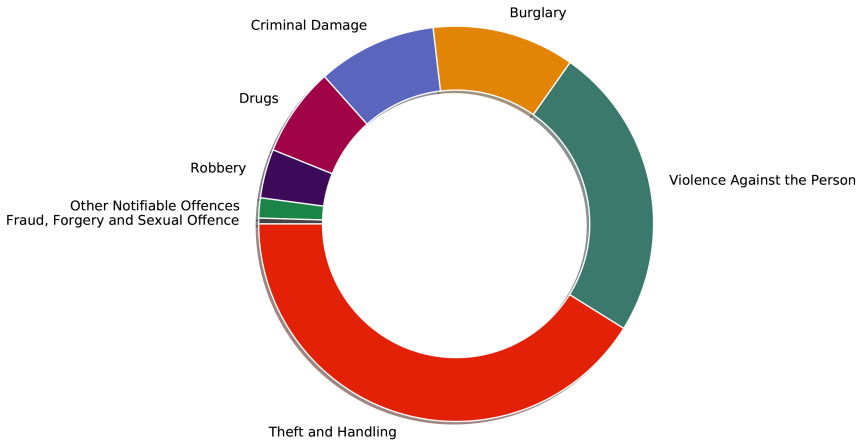


Figure 2: Major crime fractions over the total.

2.2 Geolocalized Features Distribution

We already looked at data from a statistical perspective, but since we're dealing with geographic entities such as LSOAs and boroughs a further geolocalized crime inspection is needed. In order to do so we enriched the filtered dataset with geographical attributes, such as the shapes of boroughs and LSOA.

To proceed with a time-series-like approach, we split the geo-dataset by year. At this point we have 9 datasets, one for each year between $[2008, 2016]$, sharing the same schema and semantics.

We also computed a quantitative attribute `crime/month`, equals to $\frac{value}{12}$. This measure is more understandable since its referred to a smaller time interval and so it gives a stronger impression.

Since here the purpose is to better understand the trends highlighted in the exploratory analysis section, we choose to reproduce the same plots in this geolocalized analysis. The quantitative information that we choose to represent is the crime/month attribute. We modeled and visualized this dimension at LSOA-level with the help of a color-map, while we still show boroughs edges.

In each choropleth map we also show:

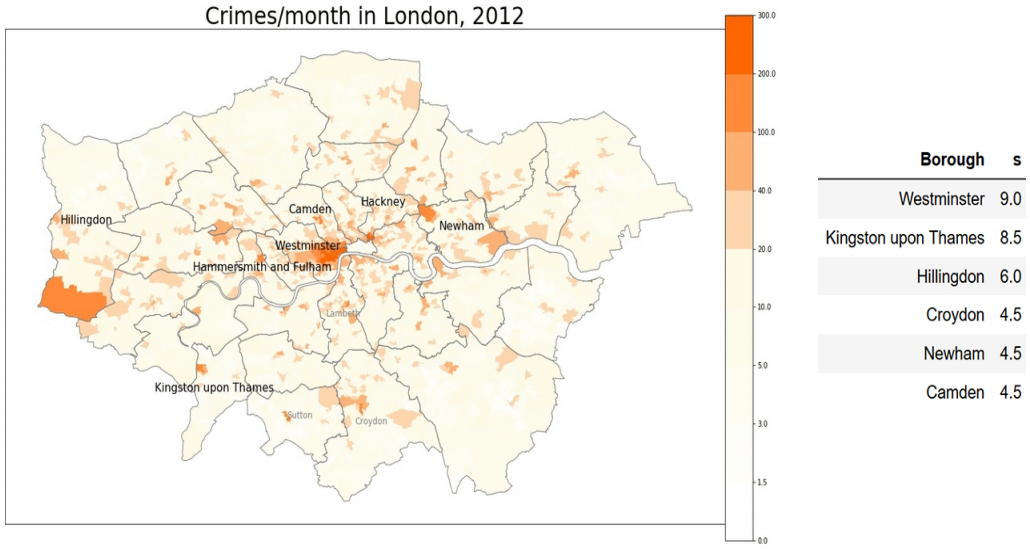
- in *black* the names of the borough with at least one LSOA that falls in the two highest ranges of the color-map. We will call such LSOA an **hotspot**;
- in *gray* the names of the boroughs with at least one LSOA that falls in the third highest ranges of the color-map; We will call such LSOA a **concerning** one;

This additional information will also provide us an implicit borough classification, indeed we can now compute a **colormap-induced-score** s and rank boroughs in each plot. For each borough b, the score is computed as:

$$s(b) = n_b + \frac{n_g}{2}, \quad n_b, n_g \in [0, 9]$$

where n_b and n_g are respectively the number of times b appears in black and gray in the map. To assign a lower weight to gray-appearance-events we divided n_g by 2. This score will provide us a way to better understand the crime spatial distribution. We found that there's a **spatial locality** in crime, meaning that dangerous boroughs and hotspots don't change too much over the years. This behavior has been observed not only for the general crime value, but also for all major categories. As an example Figure 3 shows the map related to the total number of crimes in London in 2012.

We notice that hotspots inside dangerous boroughs are usually isolated, indeed in a dangerous borough there is at most one critical LSOA, and this LSOA often confines only with not-critical ones - an example is Kingston upon Thames; Westminister and, in general the LSOAs above the Thames tend to be the most dangerous zones.



Borough	s
Westminster	9.0
Kingston upon Thames	8.5
Hillingdon	6.0
Croydon	4.5
Newham	4.5
Camden	4.5

Figure 3: All crimes distribution over London, 2012.

Finally the central white LSOA corresponds to the City of London area and there are no crimes associated with this zone.

The table on the right of the map shows the boroughs ranked by our score s computed for each year. The first three dangerous boroughs, resulting from the table, have a score really close to the maximum. This means that they have been the most dangerous ones almost every year.

We repeated the very same analysis for each major category c_i, computing every time the related score s_{c_i}. Once we have collected all the scores, for each borough we can now define a **weighed score**:

$$\forall b \in \{Boroughs\}, ws(b) = \sum_{c \in MC} s_c(b) \times w_c$$

where $ws(b) \in [0, 9]$, MC is the set of major categories of crime considered, $s_c(b)$ is the score of the borough b in the category c and w_c is the weight assigned to the crime c. The weights are the percentage of crimes with major category equals to c over the total crime number.

This score represent a first, naive, **danger indicator**: a score close to 9 means that inside that borough, since 2008, there was at least one hotspot for each crime category. The table below shows a normalized version of ws.

Borough	*ws*
Westminster	0.98
Kingston upon Thames	0.77
Camden	0.75
Hillingdon	0.63
Croydon	0.56
Hammersmith and Fulham	0.53
Newham	0.52

Looking at the boroughs names we have that:

- 3 boroughs that are in the city centre and on the northern bank of the Thams: Westminster, Camden, Hammersmith and Fulham; this confirms the hypotized dangerousness the involved zones
- Hillingdon, even if appears with an high score, shows almost always only one hotspot, that coincides with the LSOA that contains the airport
- all the other boroughs in the list are like Hillingdon, and this high crime density in just few LSOA is certainly a phenomenon that deserves further analysis.

3 Correlations

Also in the correlation analysis we choose to study the problem with a time-series-like approach by means of the previously splitted datasets. The main problem in this phase was the lack of numerical quantitative attributes in the original dataset; indeed we only have value. The first thing to do is to reshape the datasets in order to have more independent variables.

3.1 Dataset Reshaping

The first idea was to create one numerical attribute for each major category that contains the sum of the values of that kind of crime in the current year - since we're dealing with a dataset split by year - for a specific LSOA. Thus these new datasets will contain 4835 rows, namely the number of LSOA, and 9 new attributes, one for each major category.Furthermore we extend our dataset with economic and demographic indicators and points of interest (Subways, Train Stations, Taxis, Airports, Bus Stations, Parkings, Monuments, Hospitals, Police Stations, Stadiums).

3.2 Correlations Analysis

We first compute the correlation between crimes both at LSOA and borough level.

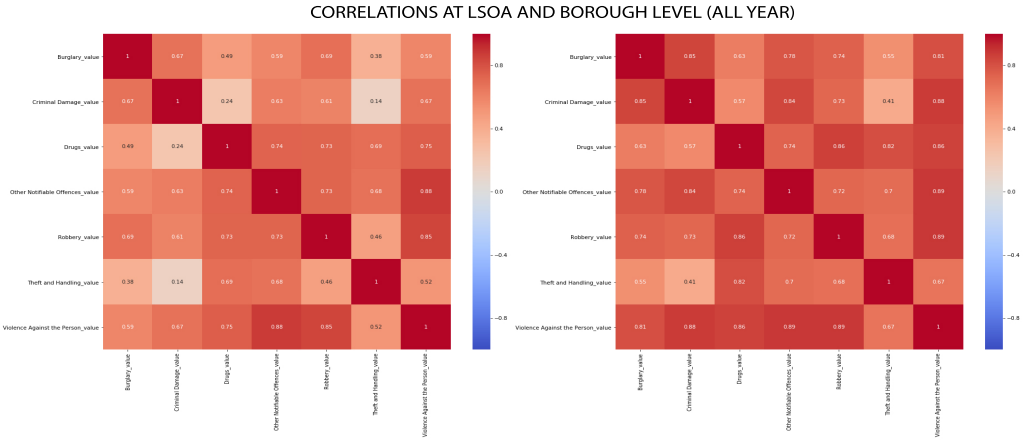


Figure 4: On the left the LSOA-level crime's correlation matrix, on the right the borough-level one.

Since the mutual correlations between different categories of crimes are strong both at LSOA and borough level we can say that if a borough has an high number of crimes, and thus can be considered dangerous with our score, it's likely to have an high number of crimes in all categories; this observation its related to the score defined in the previous section because the crime levels seem to be higher in some recurring zones as the northern bank of the Thames and the city center in general, resulting in very high scores in the final rank table.

Then we analyzed the correlation among all the attributes of the enriched dataset at LSOA level, but the only strongly correlated subset of attributes was again the one of crime values. All the new features

seemed not correlated with the values of the crimes, the only notable correlation is between the number of Airports in an LSOA and Other Notifiable Offences. We now try to see the same correlations at borough level. This time the correlation matrix showed better results. We analyze the most significant ones by looking how the correlation varies over the years.

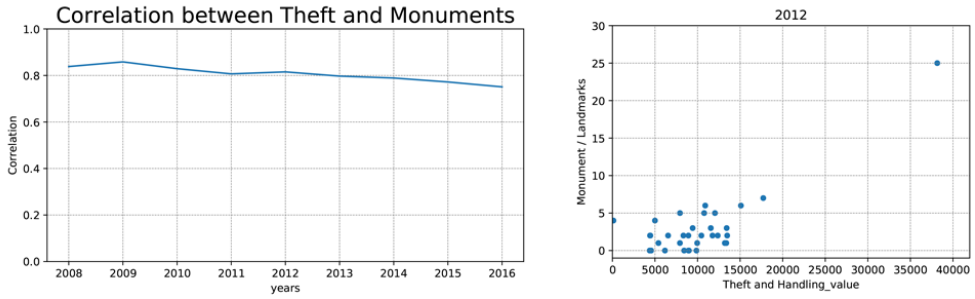


Figure 5: Correlation between Theft and Handling and Monuments, with scatter plot

The line-plot 4 confirms our first impression: the theft crime is focused in the center of London; the (Touristic) Center can be individuated by the position of the monuments and assuming that a Borough is as central as the number of monuments that falls in that Borough we can conclude that places near monuments have an high risk of theft and handling crimes (for example because they are crowded by thousand of tourists). We also noticed that this correlations is decreasing linearly during years.

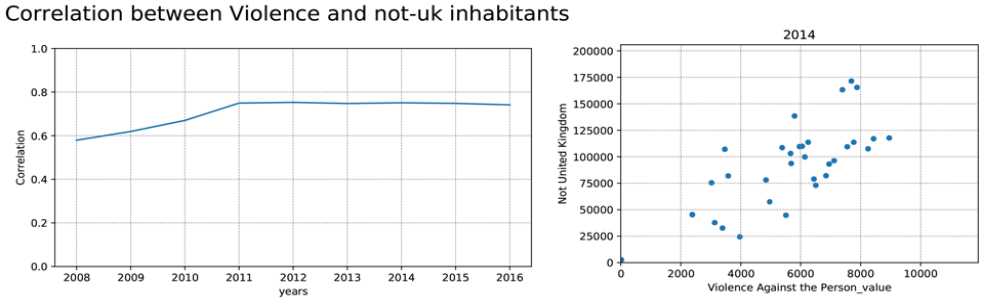


Figure 6: Correlation between Violence and non UK inhabitants, with scatter plot

In the plot above we found strong correlations between a Violence and not UK inhabitants

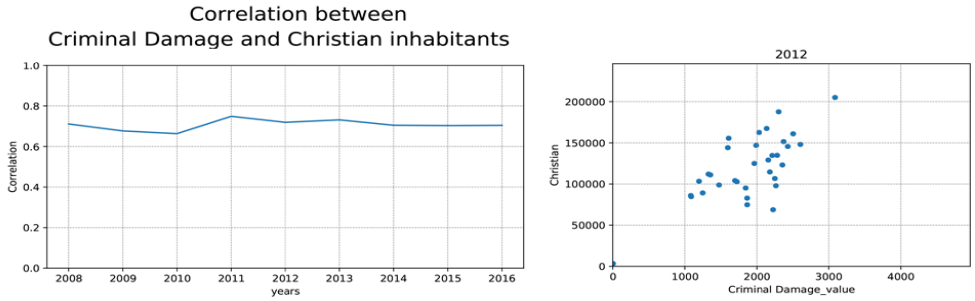


Figure 7: Correlation between Criminal Damage and non Christian inhabitants, with scatter plot

Finally these last plots gives us the opportunity to highlight that even if a crime is happened in a place it doesn't mean that their population is the active part, but can also be the victim. So in this case we can't know if the Christian/not UK population plays an active or passive role in this crime category.

4 Analytic Project Proposal

In this section we propose three analytic projects based upon our analysis:

1. Model that predicts the number and the kind of crimes that will occur in a specific LSOA given the month and year.
2. Model that predicts the number and the kind of crimes that will occur in a specific BOROUGH given the month and year.
3. Given the available resources of London Police provide a *smart* way to arrange them in order to reduce the number of crimes according to the prediction of our model.

In each case we will consider this problem as a time-series and use relative clustering algorithms and modeling techniques.

After the data understanding presentation, we discussed points 1 and 2 of the list and, together with the PhD researcher Luca Pappalardo, we decided to merge them in one unique parametric model. In this way the user will be able tho choose the level of prediction (borough or LSOA).

The last proposal remains still valid and can be seen as an extension of our predictive model.

4.1 Clustering

We decide to perform clustering at **LSOA** level to find groups of *similar* **LSOAs**.

4.1.1 Dataset Reshaping

In order to perform clustering we reshaped the dataset grouping by LSOA code and we sum all the crime values.

4.1.2 Attributes selection

Looking at the correlation matrix (fig 4) of the reshaped dataset we can state that crimes have a strong correlation with each other. In a clustering analysis it's better to select attributes that aren't strongly correlated, because variables that are strongly correlated represent the same concept, and if we insert them in the model this concept is represented twice in the data, having in this way more weight than the other variables.

After this analysis we tried different combinations of attributes among the weak correlated ones using from 3 to 5 features; the best combination was the one using Burglary, Drugs and Other Notifiable Offences values. After the feature selection we perform clustering using three methods: DB-scan, Hierarchical clustering and K-means.

4.1.3 DB-scan

The *density-based* clustering method needs two parameters: ϵ that is the maximum distance between two samples for be considered in the same neighborhood and *min-pts* that indicates the minimum number of points in a neighborhood of a point to be considered a core point. For estimating this two parameters we used the *k-distance plot* of our dataset; we compute K-nn for each points and we analyze the density distribution of the data; once we choosed a *min-pts*, we fix k to that value. Then we choosed as ϵ the k-distance corresponding to the area of the k-distance plot (fig. 8) with a low slope.
We selected $\epsilon = 0.05$ and *min-pts* $= 200$. The result consists in an high-silhouette clustering, but we only obtain two clusters, one containing *noise* points; although we obtained an high silhouette clustering we discarded it because give us no significant information.

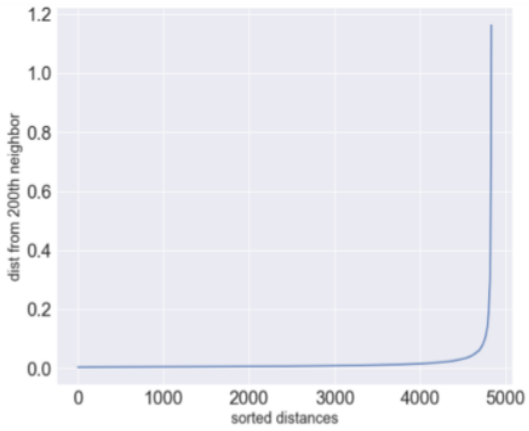


Figure 8: K-nn plot

4.1.4 Hierarchical Clustering

Then we try to see if the data points could be grouped in a hierarchical way. We computed the proximity matrix between all the points using the euclidean distance before performing the clustering with the *complete-linkage* criteria. After this phase we analyzed the dendrogram (fig. 9) and we decided to cut at height 27, because above this height the distance to merge two cluster becomes too high obtaining a clustering that gave us four clusters; the silhouette score is 0.90 but the clusters are very *unbalanced* w.r.t. the cardinality.

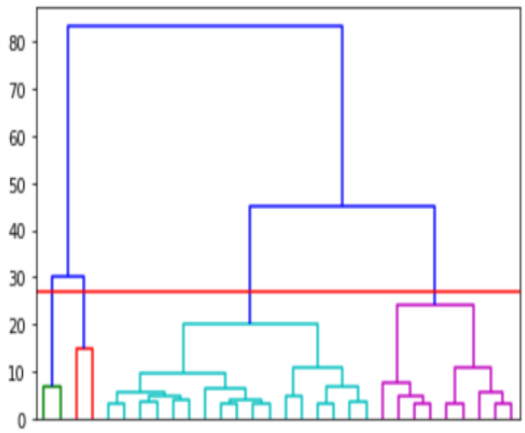


Figure 9: Dendrogram

4.1.5 K-means

Finally we perform a K-means on the data points; since the number of clusters k is a parameter of the method we first tried to estimates this value. After running K-means with $k \in [2, 20]$ and collecting the SSE score we analyzed the SSE-plot. As we can see from the figure below, using the *elbow-method* we identified the range of the optimal k as $[4, 6]$ so after performing the K-means with the k in the ranges we found that the optimal k was 4.
The execution of the algorithm gave us a silhouette of 0.45 and a SSE of 7.08. Even if we have a lower silhouette then in the hierarchical clustering we choose the k-means execution as the best since we had *better* clusters.

As we can see in the fig. 11, the clusters obtained represents a set of LSOA divided according to their *dangerousness*.

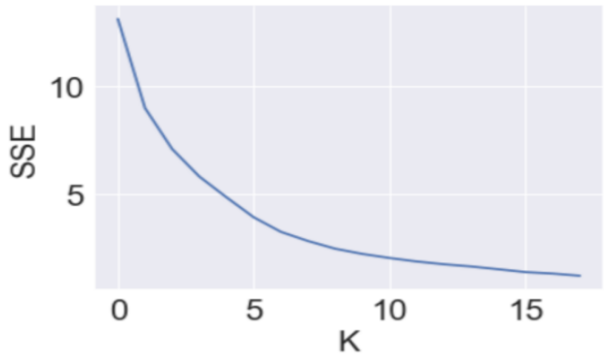


Figure 10: SSE plot

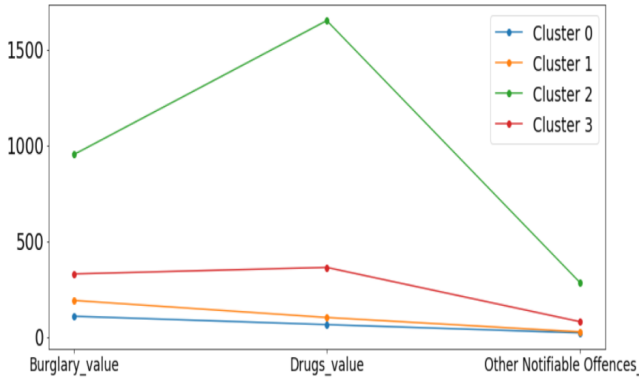


Figure 11: SSE plot

For each cluster we compute the mean value of the different type of crimes and we made a bar chart (fig 12); we can see that LSOAs are divided according to their mean value of crimes, for example, cluster 2 contains 16 LSOAs with the higher value: this LSOAs belong to the most dangerous boroughs like Camden, Croydon, Hillingdon, Islington, Westminster and Hackney. Then we have in order cluster 1, cluster 3 and cluster 0 with decreasing values. In the right-most figure below we can see how the data are distributed in a three-dimensional space, each dimension is one of the clustering attribute, the data points are really dense.

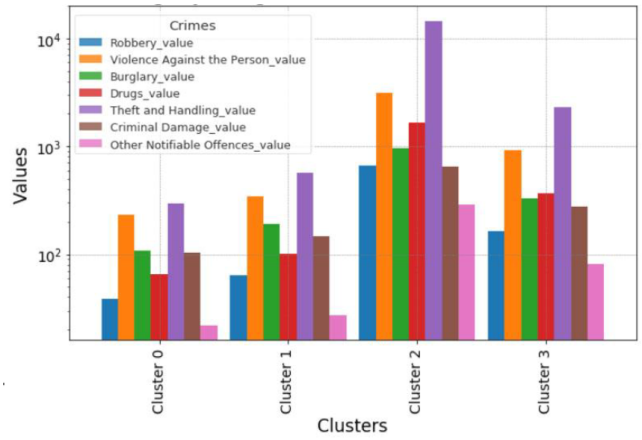


Figure 12: Barplot

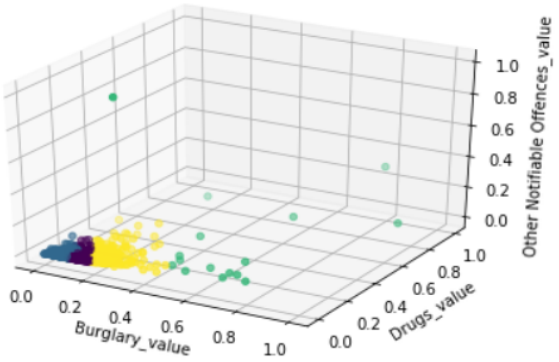


Figure 13: Data in a 3D space

4.2 Analytic Proposal Implementation

As said before, the goal of our analytic proposal is to output a parametric model capable of predicting the number of crimes at LSOA or borough level.

4.2.1 Data preparation

We choose a time-series approach and before proceeding we need some data preparation steps. We build a dataset where there is a row for each LSOA and month - intended as the couple year-month; there is

also a numerical attribute for each major category of crime. Out of this new dataset we can extract the time series for each borogh and LSOA in order to analyze it.

4.2.2 Stationarity test

We decided to start with the ARIMA regressor, with autoregression parameters configuration (1,0,0). It requires the time series to be stationary; we used Dickey-Fuller test and plot rolling average and standard deviation to investigate time series stationarity.

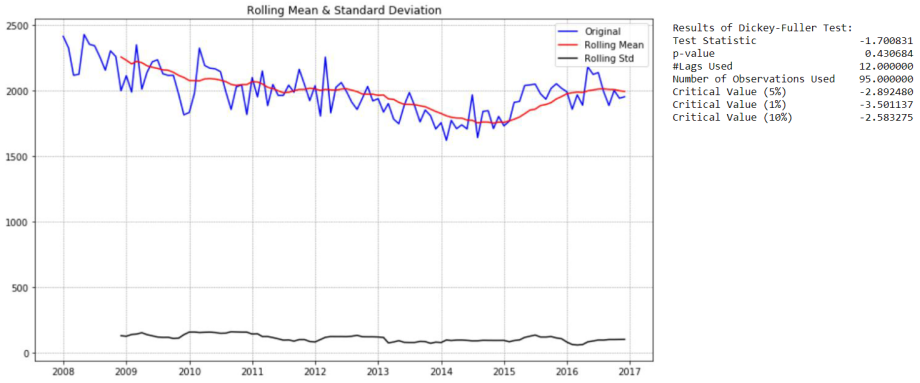


Figure 14: Stationarity test

A time series is stationary according to the Dickey-Fuller test if the test statistic is lower than one of the confidence interval values. The majority of boroughs exhibits a behaviour similar to the one plotted above: even if the series is not strictly stationary we will assume them so because of the constant standard deviation trend.

4.2.3 Score Matrix

Since it is not possible to have one unique model for all boroughs, we would have a specific model for each borough. We need a rapid way to visualize and compare scores, this method consists in the visualization of a **Score Matrix**. The Score Matrix S is an $nb \times nm$ matrix, where nb is the number of boroughs and nm is the number of year-month couples. The generic element $s_{i,j}$ is the score of the model for borough i trained in $[start\_date, j-1]$ and tested in $[j, j+2]$. We built three different matrices, one for each score:

- Mean Absolute Error (**MAE**)
- Mean Absolute Percentage Error (**MAPE**)
- Pearson's Correlation

An example of a score matrix is given in Figure 5. This matrix is a way to visualize the cross validation performances across all boroughs. An accurate and stable model will result in a regular and darker score matrix. Looking at a generic row, a consecutive block of dark cells means that the model predicts very well that period, so ideally we can spot the **most-predictable boroughs**.

4.2.4 Threshold-based Score

We define a score for evaluating the *stability* of the learning and forecasting process w.r.t. the train size. The score is defined as the **maximum** number of **consecutive** values in a Score Matrix row that are **above or below** (depend on the score used) a certain **threshold**.

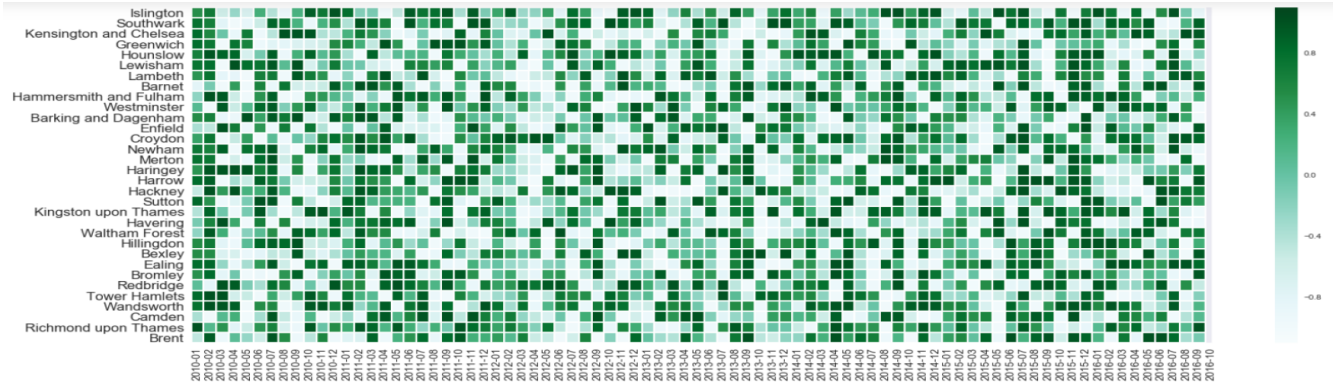


Figure 15: Score Matrix for the correlation

We compute the score for the Score Matrix, obtaining the following results (top 5):

Borough	Score
Lewisham	7
Havering	5
Croydon	5
Hammersmith and Fulham	5
Islington	4

4.2.5 Lewisham analysis

As we can see from the table above the borough of Lewisham has the best score; we wanted to analyze this borough during the period where we have the block of **seven** consecutive cells whose correlation is above 0.7. We selected the cell for Lewisham borough with j=2010-04-01; according to the definition of the Score Matrix we have to train the model in the range [2009-01-01, 2010-03-01] and test in the range [2010-04-01, 2010-06-01] (three months). As we can see in the figure 16, looking at the red line, the forecasting is able to capture the trend (correlation of 0.9) but it's shifted by a constant.

4.2.6 ARIMA(1,0,0) as a baseline

	MAE	MAPE (%)
ARIMA (1,0,0)	53.86	2.4
Dummy AVG	87	4.1
Dummy Last	148	6.3
Dummy Lasts	234	11.9

Figure 16: Comparison of the regressors, the dashed grey line divides train and test

In order to use the ARIMA(1,0,0) model as a baseline we compared the obtained results with three dummy regressors:

- Dummy **AVG**: return the average of the training set
- Dummy **LAST**: return the last value of the training set
- Dummy **LASTS**: return the n-last values of the training set, where n is equal to the size of test set

As expected the ARIMA model gives us the best results among the selected regressors, so the auto-regressor model is our baseline, in fact, in the next steps we will try to improve the accuracy of our model.

5 Final Directions

The goal of our analytic proposal is to output a parametric model capable of predicting the number of crimes in an LSOA or borough. We choose a time-series approach. We have proposed an auto-regression model (ARIMA) that predicts the number of crimes in a borough as our analytical solution and analyzed its performances compared to dummy baselines. ARIMA outperformed dummy regressors but there was little space for improvements with such models because of the high unpredictability of the crime phenomenon.
After discussing the further directions of the project with the professors we decided to **change the granularity** of our analysis:

- **prediction granularity**: instead of predicting the number of crimes we will consider **ranges** of values at **borough** level thus transforming the regression problem into a classification one.
- **learning/data granularity**: hoping that locality will increase the stability of the time series, we will predict the number of crimes at **LSOA** level.

In this way instead of having a single parametric model for both LSOA and borough we'll have two different tasks. We decided to solve two separate problems at different granularities because boroughs' time series have values that vary in wider range than the LSOA's ones, resulting in more unstable series.

6 Range prediction

We decide to change approach switching from a value prediction to a range prediction, basically from regression to a multi-classification problem.

First of all we had to transform the values in ranges/classes through a *discretization*. We decided to have **five** classes of risk, from Class 1 (safe) to Class 5 (dangerous) so we performed two techniques of binning: natural and equal frequency binning.

In this section we will show the results of range prediction on Westminster, that is the most dangerous and unpredictable borough we've seen so far.

Natural binning

In the natural binning the bins have equal width but, as we can see in Figure 17 the distribution is *skewed*, so we try another binning.

Equal frequency binning

Since we want a balanced distribution of the data, to better train the model with an equal amount of data for each class, we decided to perform an equal frequency binning; Figure 17 shows that each bin contains the same number of values.

The value-range of each class is in the form $[L, S)$, from it we can compute a representative value, taking the average as $\frac{L+S}{2}$ and dividing it for the number of days in a month, in this way we obtain an estimation of the number of crimes for day for each class. For a generic Class i:

$$crimes/day_i = \frac{\frac{L_i+S_i}{2}}{30}$$

The following table shows the representative value for each class.

Class	Crimes/day
Class 1	120.42
Class 2	130.86
Class 3	137.67
Class 4	146.05
Class 5	166.00

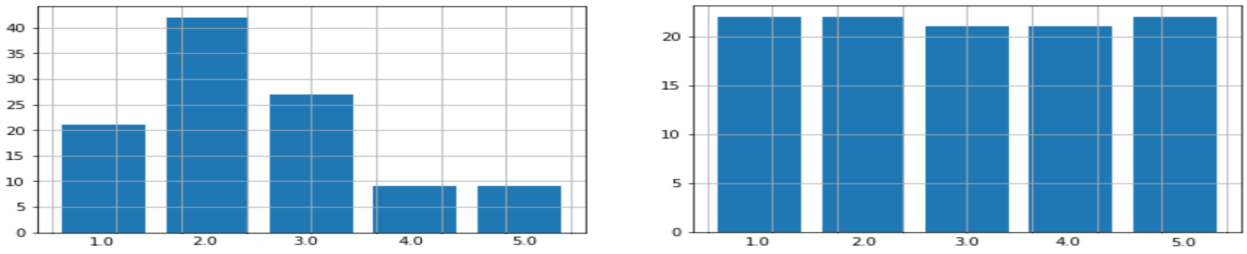


Figure 17: Natural (left) and Equal (right) binning histograms

For estimating some baselines we used also this time dummy models. The dummy classifier used are:

- Dummy **AVG**: return the average class of the training set.
- Dummy **RAND**: return a random value in the range $[1, 5]$.

The score used for the evaluation is the **MAE** (Mean Absolute Error) that gives an estimation of the error as number of class of difference.

6.1 Random Forest

The first classification method we used is *random forest*; we tried several configurations, the best-one with respect to the **MAE** had the parameters in table 1.
The parameter *prev_obs* controls the learning *granularity*, each sample have a set of attributes, in our case we have a variable number of attributes, namely the number of crimes in the previous months, and just one output variable, the next month's value. The table 2 shows samples and targets for a toy-time-series $[1, 2, 3, 4, 5, 6]$ with $prev\_obs = 3$:
Then in table 3 we compared the results of the random forest model with the dummy regressors cross-validated over the data of the borough of Westminster.

Parameter	Value
prev_obs	2
n_estimators	16
bootstrap	*False*
max_depth	*None*
min_samples_leaf	5

Table 1: Random Forest parameters

Sample	Target
[1,2,3]	[4]
[2,3,4]	[5]
[3,4,5]	[6]

Table 2: prev_obs example

Model	avg(MAE)	std dev
Dummy AVG	1.373	0.416
Dummy RAND	1.750	0.251
Random Forest	1.132	0.564

Table 3: Results comparison

6.1.1 Interpretation

Since $prev\_obs = 2$ we have a couple of features: $(Class(M_2), Class(M_1))$, where M_1 is the month at distance 1 from the month to predict (the previous month), similarly M_2 is the month at distance 2 and $Class(M_x)$ returns the Class for the month x. We analyzed the *feature importance* for the random forest model: $Class(M_1)$ and $Class(M_2)$ have an importance of, respectively, 0.604 and 0.395; this means that the class of the previous month (M_1) is ≈ 1.5 times more important than the class of M_2.

Now we analyze one of the 16 trees in the model; as we can see from Figure 18 the root node splits on $Class(M_1) \leq 3.5$, if *true*, assuming we want to classify M_{pred}, then $Class(M_{pred}) \in \{1, 2\}$ that are the "safe" classes. The difference from a Class 1 and a Class 2 is made by $Class(M_2)$:

- $Class(M_2) \leq 1.5$ then Class 1
- $Class(M_2) = 2$ then Class 2
- $Class(M_2) > 2.5 \ \wedge Class(M_1) \leq 2.5$ then Class 1
- $Class(M_2) > 2.5 \ \wedge Class(M_1) < 2.5$ then Class 2

Else, if $Class(M_1) \leq 3.5$ is *false*, $Class(M_{pred}) \in \{3, 5\}$ that are the "dangerous" classes.
The difference from a Class 3 and a Class 5 is made by

- $Class(M_1) = 5$ then Class 5
- $Class(M_1) \leq 4.5 \ \wedge Class(M_2) \leq 2.5$ then Class 3
- $Class(M_1) \leq 4.5 \ \wedge Class(M_2) > 2.5$ then Class 5

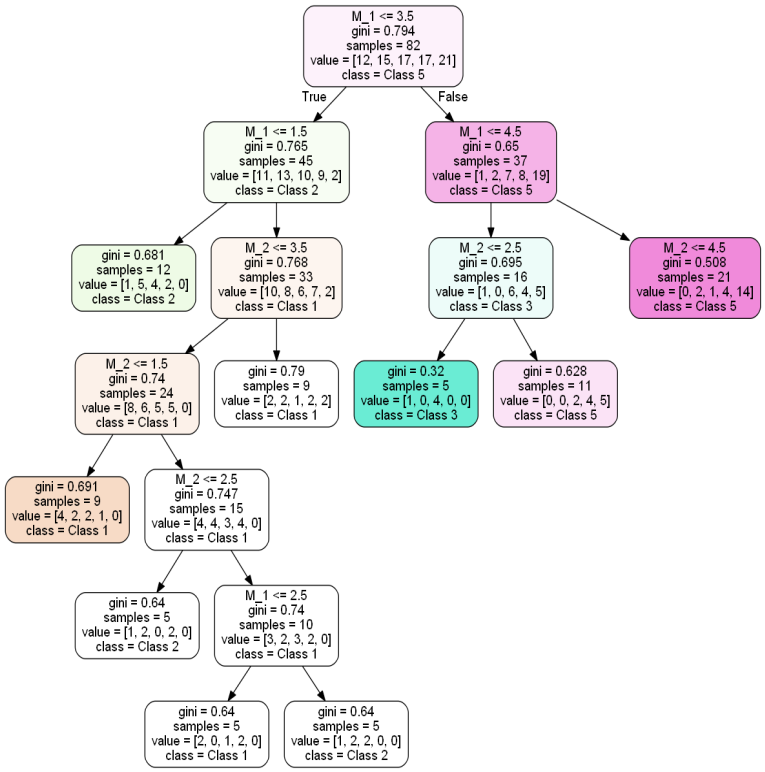


Figure 18: Decision Tree visualization

Parameters	Value
prev_obs	2
n_estimators	10
learning_rate	0.17
max_depth	4
min_samples_leaf	2
subsample	0.8

Table 4: Gradient Boost best parameters table.

6.2 Gradient Boosting

We tried also the *gradient boosting* method, also this time several configurations were evaluated; the best-one with respect to the **MAE** had the parameters shown in Table 4.
Than we compared the results of the gradient boosting model with the dummy regressors cross-validated over the data of the borough of Westminster, the results are shown in Table 5.

Model	avg(MAE)	std dev
Dummy AVG	1.373	0.416
Dummy RAND	1.750	0.251
Gradient Boosting	0.926	0.384

Table 5: Gradient Boost cross validation results.

6.2.1 Interpretation

Also this time the *optimal prev_obs* is 2, so we analyze the feature importance of $(Class(M_2), Class(M_1))$; this time the importance is more balanced, in fact is, respectively of 0.466 and 0.533.

6.3 Results

In this section we compare the dummy and the presented methods, plotting also an example where we predict four months, one at time. We train the models from 2008 to late 2014 and we predict February, March, April and May of 2015. The following table compares the scores collected with the cross-validation.

	Cross validation		**Test (1, 2, 1, 3)**	
Models	**avg(MAE)**	**std dev**	**predicted**	**MAE(test)**
Dummy AVG	1.373	0.416	(3, 3, 3, 3)	1.25
Dummy RND	1.750	0.251	(2, 1, 3, 2)	1.25
Random Forest	1.132	0.564	(1, 2, 1, 2)	0.25
Gradient Boosting	**0.926**	0.384	(1, 2, 1, 2)	0.25

Table 6: Results of the range prediction models

As we can see the random forest and the gradient boosting models give an accurate estimation, considering that they are trained to predict only one month in the future: the predicted class differs at most by 1.133 from the real one. The best model in this case is the Gradient Boosting since it has a greater accuracy (lowest value of MAE) and is more stable (lower standard deviation than Random Forest.

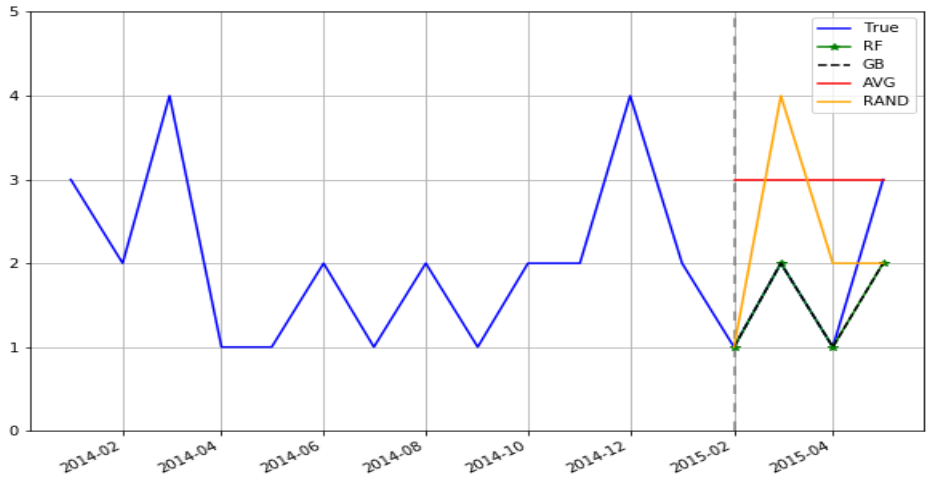


Figure 19: Test on February, March, April and May 2015.

7 LSOA prediction

In the data understanding phase we spotted LSOAs in London where the crime value is always high all over the year (e.g. Westminster, Camden etc). We then performed clustering in order to understand if

and how LSOAs can be grouped and characterized in some meaning full way; using as clustering attributes the number of crimes within a specific category.
Among all major crime categories used, three of them separate LSOA in dangerousness classes, precisely Burglary, Drugs and Other Notifiable Offences values.
After looking at the figure 13 we decided to focus our analysis on the cluster of dangerous LSOAs.

We will propose a model for a generic LSOA in cluster 2 and assume that this model approximate well also the behaviour of LSOA in the same cluster. Then we will validate our assumption looking at how the selected model works and performs. LSOAs in cluster 2 belong to the previously-discovered dangerous boroughs: Camden, Croydon, Hillingdon, Islington, Westminister and Hackney. We search for the model that best fits the first LSOA in the list, that belongs to Camden.

7.1 Model Selection and Interpretation

We decided to follow the analysis used in a paper[1] presented in class using three regressors: Random Forest, Gradient Boost and ExtraTrees. Each model was evaluated using the time series cross-validation on normalized data according to four scores: **EV** (Explained Variance), **MAE** (Mean Absolute Error), **R2** and **Person Correlation**; the following table summarizes the results.

Model	EV	MAE	R2	Corr
Random Forest	0.13	**0.113**	**-0.255**	**0.47**
Extra Trees	0.16	0.119	-0.338	0.4
Gradient Boosting	**0.17**	0.134	-0.47	0.46
Dummy_AVG	0	0.14	-0.87	//

Table 7: Regression scores comparison

All the best configurations outperformed the baseline, and among them, the random forest reaches the best performances in terms of all scores but one, so we choose to analyze it.

7.2 Random Forest Regressor

The configuration for the random forest models consists in an ensemble algorithm with **five** regressors with an unbounded depth and uses input with size nine, so $prev\_obs = 9$; the generic input is in the form $(Value(M_{prev\_obs})...Value(M_1))$.

7.2.1 Feature Importance

The bar plot in Figure 20 represents the feature importance for the random forest regressor. The most important measurements for predicting the total number of crimes that will occur in the next month are once again the one collected in the previous two months and the one collected eight months before.

Furthermore, in Figure 21 we have the autocorrelation plot of the analyzed series. This time series has positive and significant *autocorrelation* at lags 1 and 2, so we can expect some recurring behaviour in small temporal ranges.

7.3 Assumption Evaluation

Now we will evaluate our assumption that was that a model trained on a critic LSOA approximates well also the behavoir of LSOAs in the same cluster.

[1] Mining large-scale human mobility data for long-term crime prediction:
https://epjdatascience.springeropen.com/articles/10.1140/epjds/s13688-018-0150-z

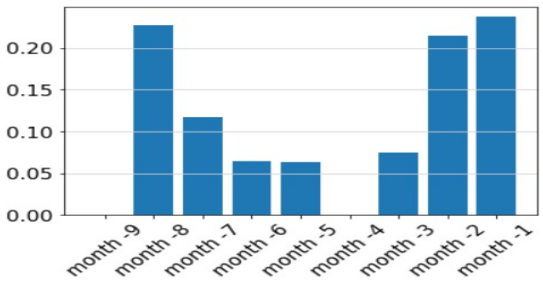


Figure 20: Feature Importance

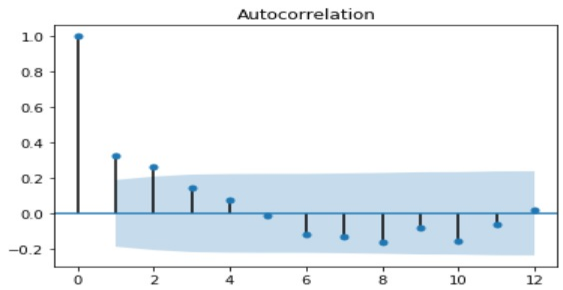


Figure 21: Autocorrelation plot

We use the random forest model with the parameter estimated in the previous step and we perform a Cross Validation where: For each step i of the Cross validation we build a Random Forest regressor RF_i that is trained in the current fold of the LSOA of reference, than we test RF_i on the others LSOAs and we collect the results in terms of EV, MAE, R2 and Correlation.

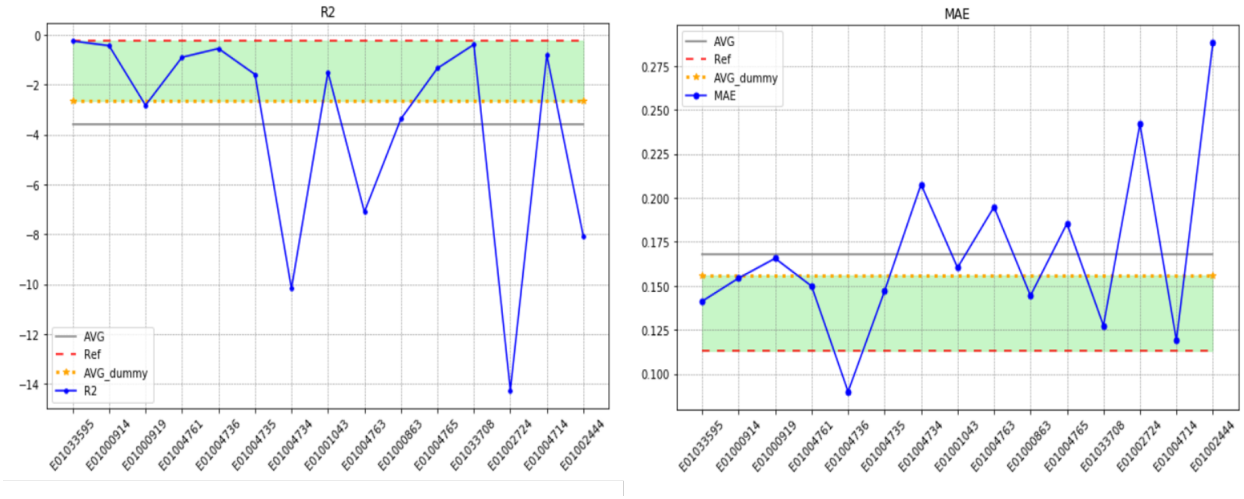


Figure 22: Score evaluation within the same cluster, the green rectangle between the two dashed line represent our *goodness interval*. R2 plot on the left, higher the better. MAE plot on the right, lower the better.

Figure 22 represents the first step towards our assumption evaluation: on the x-axis we have the LSOA belonging to the same cluster as the reference one; on the y-axis there's the score of the random forest tested on the specific LSOA. The green rectangle between the two dashed line represent our *goodness interval*: the red line is the score of the model on the reference LSOA, while the yellow one is the dummy regressor score; hence each point that falls in this range of values represents a good score for us because it's better than the dummy but still lower than the reference one.

As we can see in both plots the majority of LSOA has a score withing the goodness interval, and in some case the model behaves better than the reference-LSOA execution. So these plots confirmed that the model is quite stable with the respect to elements of the same cluster.

7.3.1 Null Model

As further evaluation of our assumption, we use two null models: we first try to randomize the data and then we also tried to train the model as before, on the reference LSOA, and test it on LSOAs from other clusters. In both cases we expect a critical degradation of performances.

Shuffle null model

We first tried to shuffle the values of the reference time series and cross validate the model with such randomly-shuffled values.

Model	EV	MAE	R2	Corr
RF on reference LSOA	0.13	0.113	-0.255	0.47
RF on shuffled data	-0.66	0.14	-1.004	-0.25

Table 8: Score comparison of null model (shuffled data).

In Table 8 we show how randomizing the data led to a critical performance degradation.

Different cluster null model

We propose another null model in which we collect the performance of the RF trained on the reference LSOA and tested on LSOA belonging the another cluster, that is the cluster with least dangerous LSOAS. In this way we will see whether the proposed model behaves better on the reference LSOA's cluster, thus validating our assumption.

The values of the LSOA crimes of this cluster are much lower than the reference LSOA. Therefore to perform this evaluation we normalized the data with the same scaler used to normalize the data of the reference LSOA.

Model	EV	MAE	R2	Corr
RF on reference LSOA	0.13	0.113	-0.255	0.47
RF on different cluster data	0	0.635	-240	//

Table 9: Score comparison of different-cluster null model.

As expected the results obtained are very poor, so the model is more suitable for the prediction of crime values of similar LSOA, in our case belonging to the same cluster.

7.4 LSTM

After regression with the classic techniques, we tried a neural network approach. In particular we decided to use LSTM neural network as it is very suitable for predicting time series. The approach used is the same: to build the model on a dangerous LSOA in order to predict crimes in the LSOA belonging to the same cluster. We set parameters for the neural network and we decided to change only the number of observations useful to make the prediction to the next step. For each execution, we calculated the relative scores.

	2	3	4	5	6	7	8	9	10	11	12
EV	**0.226**	0.218	0.242	0.21	0.228	0.143	0.168	0.177	**0.23**	0.201	0.136
MAE	**0.127**	0.129	0.127	0.13	0.128	0.126	0.126	0.123	**0.119**	0.122	0.127
R2	**0.209**	0.204	0.227	0.2	0.226	0.139	0.167	0.174	**0.229**	0.201	0.115
CORR	**0.609**	0.55	0.563	0.512	0.529	0.379	0.412	0.425	**0.518**	0.47	0.373

Table 10: Scores of the LSTM model varying prev_obs parameter.

In the table above we can notice that the number of observations to be made to have better performances is 2 or 10. These results coincide with the previous analysis:

1. During the phase of data understanding, studying the progress of crimes over the years, we have seen that their trend is repeated every 2 months;

2. With the classic regression techniques we have obtained that the best prediction is carried out by the Random Forest which requires many observations, more precisely 9.

Now we show an example of prediction and a table that compares LSTM results with Random Forest best regression.

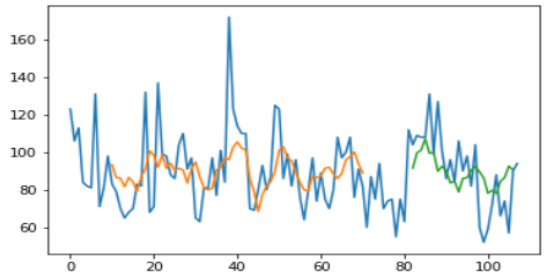


Figure 23: Example of LSTM prediction: in blue the true values, in orange the prediction on train and in green the prediction on test

Model	EV	MAE	R2	CORR
Random Forest	0.134	0.113	-0.225	0.475
LSTM	0.23	0.119	0.229	0.518

Figure 24: Scores comparison.

8 Conclusions

Our investigation of London Crime data led to interesting discoveries. Looking at how the number of cimes varies over the year between 2008 and 2016 we found that the trend is costant but for a suspicios peak in 2012. A further analysis showed that the peak was due to a critical increase in Theft and Handling felonies. We thoght that the presence of the Olympic games gathered a lot of tourists, that is a category often victim of such type of crimes.

Instead, looking at how the number of crimes varies over the month of a year we noticed that the city tends to be safer on February and more dangerous on March, but apart from those small fluctations the trend is quite regular.

Another important discovery is that awithin a year the different types of crimes mantain the same proportions: the more frequent are always theft, violence and criminal damages. Among these most frequent crimes, the Violence-related ones are the most concerning, critically increasing, doubling in the last three years. On the contrary Drugs crimes are suffering an almost linear descent during the past years, maybe due to the installation of new survelliance cameras all over the city. unfortunately the location data of this cameras is very difficult to find because those data often belong to private property or shops.

Furthermore we observed interesting strong correlation between the number of monuments in a borough and the number of theft felonies. This could be explained by the greater number of tourist near the city attractions that can be an incentive for pickpockets. Nevertheless the strongest correlations have been observed among the different type of crimes both at borough and LSOA level. This made us think that in the capital there's some *crime locality*, in the sense that the different type of crimes tend to behave similarly (grow and decrease in the same way) within a specific zone.

This supposition was further confirmed by chloropleth maps where, whatever the crime category, LSOA with high values of crimes are mostly near the upper bank of the Thames and the city centre, surrounded by LSOA with low values of crime; and this situation repeats over the years.

A clustering analysis over the values of crime in the LSOAs allowed us both to separate zones of London according to their dangerousness and to spot the most dangerous ones.

Combining all those results with the distribution analysis of the crime value over the LSOAs in Figure 23

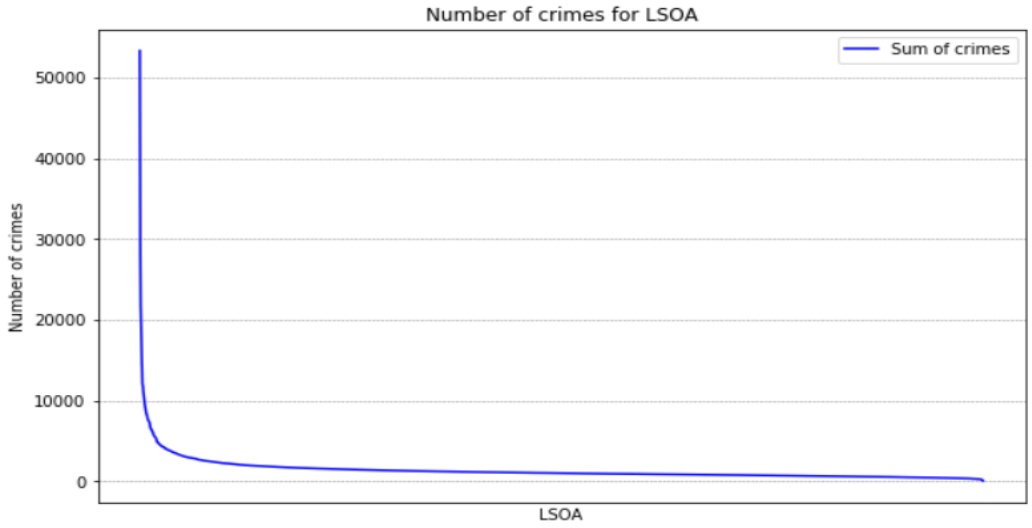


Figure 25: Crime values distribution over LSOAs

we have been able to understand that the high-crime-risk zones are rare and isolated, in fact the plot above depicts a power-law situation, so we'll have a lot of LSOA with low crime values and a few very dangerous.

The final output of our analysis is a couple of models: one that predicts in which class of danger will be a borough in the next month, the other that predicts the number of crimes that will happen in an LSOA in the next month.

A possible usage scenario is one in which the range predictor is used in order to have a general idea of the class of risk of each borough. Once we found the most dangerous one we can use the LSOA regressor to have a preciser idea of what will happen and how to deploy the available resources in a smart way, given that high risk zones are rare and isolated.