Predicting the occurrence of Violent Crime in London on a Postcode basis

Ian Dsouza

Group size: 2

Table of individual contribution by each member of my group based on my subjective opinion:

Student	Effort
Nathalie Richards (190564836)	50%
Ian Dsouza (190349211)	50%

Abstract

This paper contains predictions of occurrences of Violent Crime in London by postcode. Regression is utilized and a Bayesian Network model is run. The reasons behind using regression as the machine learning tool are discussed along with the data management and exploration. Interesting links between associative factors such as job density and the ratio of house prices to earning are discussed with causation seen when running the Bayesian Network. A literature review of related works is presented within the introduction and then later referred to. In addition, the applications of this model; such as in intelligent policing, are put forward.

Introduction

For as long as there have been laws, there has been crime. Hundreds of studies have looked at the causes and links between crime and other factors. The majority of these studies compare cities or states or countries, a few comparing boroughs or areas within cities [1][2].

In this paper, we exclusively focus on one type of crime, categorized in the Metropolitan police database as 'Violent or Sexual Assault'. Focusing on Violent Crime allowed us to maintain a more representative dataset since reducing the types of crime, enabled collection of crime records from a larger time span. In addition, it helped avoid examining overtly classicist crimes (I.e white-collar crimes such as fraud or drug-related petty theft)[3]. Violent Crime including homicide suffers least from underreporting [4] since corpses are harder to ignore than loss of property. This allows increased confidence in the integrity of our dataset.

This study compares crime on a postcode basis, examining violent and sexual crime and its prevalence by property in London, predicting the occurrences of Violent Crime in a one-kilometre radius of any London Property. Violent Crime is perhaps unique in that it can be isolated to a distinct latitude and longitude, whereas crimes such as fraud are often harder to pinpoint. It examines factors contributing to violent crime on a granular level, using large government and police datasets to delve further into contributing factors and to see what happens borough wise when these contributing factors seem to remain constant. Consequently, this study has the unique advantage of studying crime inclusively policed by one policing body, creating neutrality in policing not previously studied.

Prior to data collection, the factors cited to contribute to the violent crime were extensively researched using a combination of journals, papers and government websites; The FBI lists factors contributing to violent crime to include; population density, economic conditions, including median income and educational systems [5]. Several other factors were cited such as climate and availability of weapons that were not applicable in our study since we are examining one city with uniform weapon laws. Further research papers suggest other factors such as the average educational attainment[6], degree of urbanization and the age composition of the population. It seems increased education is strongly negatively correlated with the crime even when controlling for race and wealth measures. [3]

Age was a factor cited in multiple papers. The type of crime committed is strongly associated with age. Participation in crimes such as forgery, fraud, and embezzlement should peak at later ages. Conversely, older, more educated adults should commit fewer unskilled crimes. [7] Another study found that a high proportion of young males in the total population may promote more violent acts [3], however, it was argued that the influence of age structure on homicide becomes less evident as other risk factors for violence gain prominence[8].

Two of the primary factors examined; education and property price are cited as correlated in many reputable studies. A UK government study of property prices over three years -2013,2014,2015 found that house prices near the best schools were higher than in the surrounding areas. [9] There is a clear link between the price paid for a home and access to good schools. Other studies emerging as a consequence of the 'No child left behind' Act in the US, cite the same association. [10][11] These studies were particularly interesting since they discussed some of the biggest factors associated with Violent Crime, using no mention of crime at all, whilst other studies cite links between Violent Crime and just one of these factors, such as property [1]. Instead of just property price, our study factors in the ratio

between house price and salary which we discovered correlated more strongly with Violent Crime.

Economic factors such as job-density and business in the local area were included. A study by the University of Sheffield [12] stated that a decrease in viable economic prospects increased the incentive to engage in crime, however, since Violent Crime often has no prospect of Capital gain we were interested to find that this relationship still held true. This was found in an earlier paper which examined Violent Crime in the US, showing poverty and income are powerful predictors of homicide and Violent Crime even when standardized by age. [14]

The data for this paper was collected and combined to produce a dataset containing all the major contributing factors cited, as well as additional data that was only attainable by studying by postcode; such as distance from a school and number of violent crimes within one kilometre. Mental factors were considered in order to obtain greater insight into the impact of crime and also the potential links between the physical causes and the human effect on the population.

Data

Data retrieval

An effort was made to find data sources that were large and relatively unbiased. It was important that data was from a reputable source that had enough data points to draw sound conclusions.

The majority of datasets used were obtained from the UK government datasets. These are detailed below:

'pp_monthly.csv', a dataset of all houses sold in the UK over the last 20 years, 86 038 data points. Variables include; Borough, Price, Type, Postcode, Date.

'london-borough-profiles.csv', a dataset giving scores and ratings by London Borough. Variables include; average Age, Unemployment rate, Happiness and Anxiety.

'median-earnings-to-price-ratio.csv', a dataset which shows the median earnings to house prices ratio by borough.

'gcse-results.csv', a dataset detailing the number of high (A*-B) GCSE results by borough.

'newSchools.csv', a dataset detailing the top 100 schools in London and their postcodes.

Shape file and file path of London Borough, obtained from the London Datastore.

The data from the government database was supplemented with a dataset containing the latitudes and longitudes of every postcode in the UK, 'ukpostocodes.csv'.

Our Final dataset was obtained from the UK police dataset.

‘mayCrime.csv’, a dataset detailing all crimes reported in London in May 2019, 95 240 crimes. Variables include; type of crime, date, outcome and the latitude and longitude at which the crime occurred. This dataset was cut to focus on Violent Crime.

Data representation

All datasets were represented as pandas data-frames in our project, this allowed us to utilize a lot of in-built functions and provide consistency.

For Data Exploration we used:

Matplotlib - For basic graphs such as scatter graphs.

Seaborn- To build on top of matplotlib, providing a richer environment. Seaborn includes a plot function for rapid exploration of multiple variables.

GeoPandas- In order to work with geospatial data in python easier. GeoPandas extends the datatypes used by pandas to allow spatial operations on geometric types. Geometric operations are performed by shapely.

Data cleaning

The majority of analysis time was spent on data cleaning to make the most of a large amount of data available and explore the data in a unique way.

In every data-frame the following basic procedure was followed:

(1) Columns with consistent variable names such as ‘borough’ were set to lowercase in all data-frames to ensure data was merged correctly.

(2) Duplicates were removed using the built-in panda’s function.

(3) Confirmation of no rows with NULL values (rare due to high-quality dataset).

(4) Data types were modified, in most cases from type ‘object’ to ‘integer’. An example being the ‘date’ variable. This was converted to int of the number of years since the year 2000. Data prior to 2000 was discarded in order to remain consistent across datasets.

The working data-frame, df, was derived from ‘pp-monthly.csv’. It was cut to only include houses in London by dropping houses, not in a London Borough.

House type ‘O’ was discarded since in some cases this included warehouses or car parking spaces, not accommodation. Abnormal prices (deemed to be under 100 000) were also discarded.
Latitudes and longitudes for each postcode in the housing dataframe were obtained using an inner join with the’ uk_postocode dataset’, latitudes and longitudes of the top 100 London schools were obtained in the same way.

The function ‘calculateDistance’, returns the distance in kilometres between two different points using latitudes and longitudes. The distance of each house from a top 100 school was calculated by looping through this function. In order to use the Violent Crime occurrences, a

loop was run through every property and every crime, calculating the distance between each. Every crime within 1 km of the property was counted . Both sets of values were added to the main data-frame.

A dilemma was faced over whether to include the general borough information on our main data-frame due to concerns that merging the happiness, anxiety borough ratings to our individual housing information create a ‘Borough flag’ which would enable the Borough to be immediately determined. We were unsure about how this would affect the integrity of our data. It was included to provide interesting additional borough metrics that had not yet been explored, such as links between schooling and violent crime and the link between house prices and anxiety.

Data exploration and visualization

Since our project is based on analysis of crimes in London we decided that the best way to find meaningful and underlying trends was by creating a map borough level and the shapefile was obtained from The London Datastore.

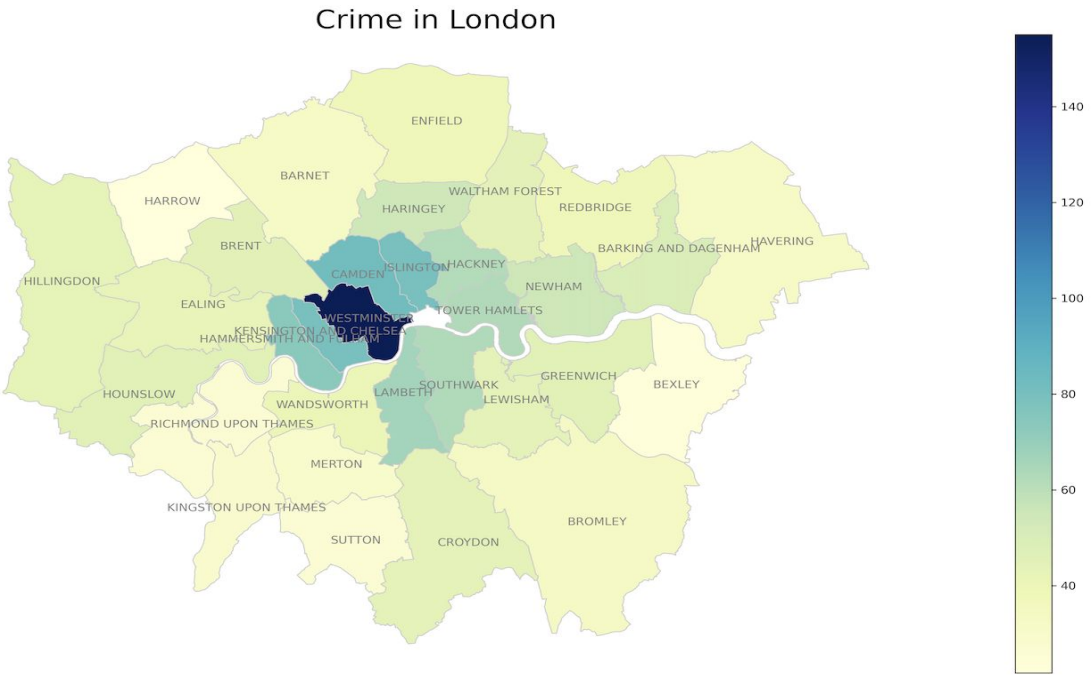


Figure [1]: Choropleth map of crime in London boroughs

A choropleth map was used to plot the crimes which visually helped us understand that crime in London is not spread throughout all boroughs but it was concentrated only in a few areas as seen in figure 1.

Figure [2]: Choropleth of Happiness by London borough

In figure 2, it's evident that areas with a high crime have relatively low happiness. Due to a large number of features in our dataset we were able to find more relations with crime, its distribution and its correlation with other features.

Method

Method selection

The dataset consists of variables with continuous numerical values, in addition to the target variable. Hence a regression model was the obvious choice for this supervised learning problem. In regression analysis, a number of predictor variables and a continuous response variable are used, a relationship between those variables is determined to allow the prediction of an outcome.

The Scikit-Learn linear regression model was used which employs ordinary least squares Linear Regression. Linear Regression fits a linear model with coefficients w = (w1, …, wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted.[13]

Other regression models were also considered [13] in the event that the standard linear regression model produced problems when testing, such as:

-**sklearn.linear_model.Ridge**, which imposes a penalty on the size of the coefficient with l2 regularization by altering the cost function. This helps reduce model complexity.

- **sklearn.linear_model.Lasso**, which estimates sparse coefficients with l1 regularization. Again, this helps in reducing overfitting and can aid feature selection.

After consideration, these models were discounted in the interest of avoiding using an over-complex model, since they are most useful with a larger number of variables or features.

Considerations and Feature selection

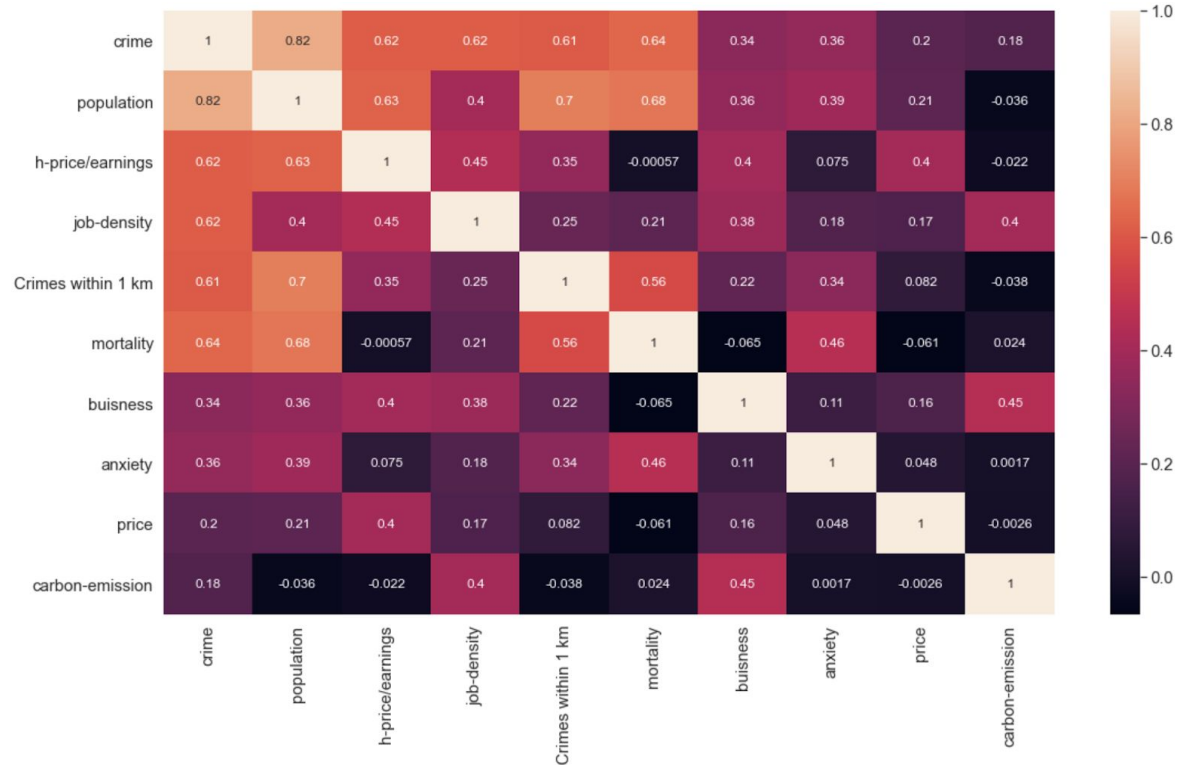
Prior to training the model, several potential problems were considered. Features were selected based on correlation.

Our dataset was reduced to 10 features, enabling the use of Seaborn 'Pairplots' for visual inspection of any relationships between variables. Graphs along the left diagonal represent the distribution of each feature, whilst graphs on-off diagonals show the relationship between variables.

Figure[3]: Seaborn pair-plots of 10 most associative features

Similarly, pandas Corr() was utilized to find the correlation between each variable in the matrix and plot this using Seaborn's 'Heatmap' function, specifying the labels and the

Heatmap colour range.



Figure[4]: Correlation heat map of features produced using Pandas

The combination of these tools was useful for identifying important features in the model quickly. Using the Heatmap we can see from the top row, that the population, h-price/earnings, job-density are positively correlated with crime.

Under-fitting is a potential danger resulting in the model being unable to learn from training data. This can arise through lack of information to accurately model real life and was combated primarily in our data collection stage. Data was collected to incorporate domain expert opinion on factors linking to crime and ensure these factors were present in our dataset, hence eight datasets from different government sources were combined in order to increase scope and reduce bias.

Overfitting is another potential problem in machine learning models, producing a model that performs well on trained data, but that generalizes poorly to any new data. This was also minimised in the data collection stage, the use of large datasets minimises the possibility the model could learn noise in the data, or learn to identify specific inputs rather than factors predictive for the desired output.

In order to ensure the model coped well with new data, it was trained on a percentage of the dataset. The dataset was split using the train_test_split utility in Scikit-Learn. A high proportion of the data, 75% was used for training and a lower proportion for testing. This was randomly selected from the data in order that the training data didn't contain characteristics

specific to one part of the data.

Modelling and validation

As with all prediction models in Scikit-Learn , the model prediction follows three basic steps:

(1) Instantiate class object : estimator = svm.SVC()
(2) Fit training data: estimator.fit(x, y)
(3) Perform prediction: estimator.predict(x)

Use of the Scikit-Learn regression model allowed access to a large number of tested metrics and scoring mechanisms to quantify the quality of predictions.
From these, we selected :

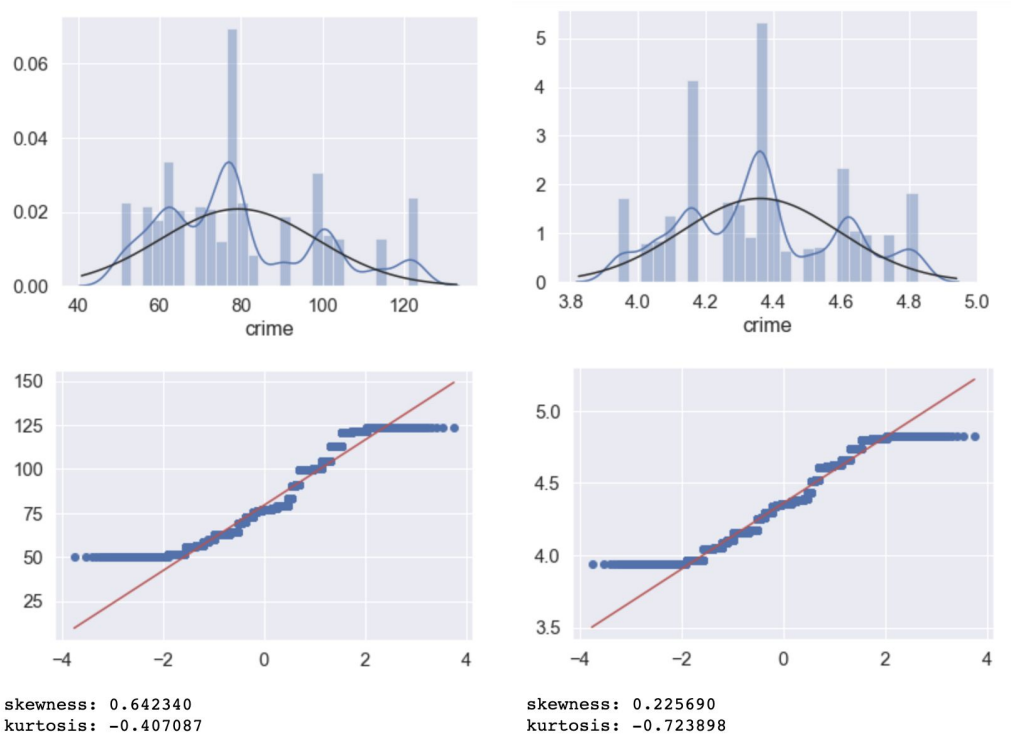
metrics.mean_squared_error, which was used to calculate the root mean squared error, RMSE, in order to measure the average deviation of estimates from observed values and explicitly indicate how much our predictions deviate from the true value. RMSE gives a relatively high weight to large errors which is particularly useful in this study since when discussing Violent Crime prediction, large errors are particularly undesirable.

metrics.r2_score, which computes the coefficient of determination, usually denoted as R^2. It represents the proportion of variance (of y) that has been explained by the independent variables in the model. It provides an indication of goodness of fit and therefore a measure of how well-unseen samples are likely to be predicted by the model. The best possible score is 1.0.

These scoring metrics complement each other well-giving access to intuitive and absolute indications of model accuracy.

Results

Due to the work and research prior to finalizing our top 10 features, we achieved an R-squared value for our model that was a perfect 1. This indicates the percentage of the variance in the dependent variable (crime) that the independent variables (features) explained collectively. The RMSE value was observed as 2.3 and 3.2 for our training and test data respectively. In order to reduce this further, the distribution of the ‘Crime’ and normal probability graph was plotted in order to identify substantive departures from normality. This included identifying outliers, skewness and kurtosis.



Figure[5]: Distribution of Crime with normality and Q–Q (quantile-quantile) plot graph

Logarithmic transformation was performed to align crime values more perfectly with the diagonal line representing normal distribution in the above graph, this decreased skewness and greatly improved the RMSE value to 1.1 and 1.19 for training and test data.

```
In [255]: df_result
          df_result.T
          # predicted 4.5 and  true value is 4.5
```

Out[255]:

	0	1
0	4.519612	4.519612
1	4.048301	4.048301
2	4.298645	4.298645
3	4.157319	4.157319
4	4.384524	4.384524
...	...	...
2044	4.048301	4.048301
2045	4.175925	4.175925
2046	3.939638	3.939638
2047	4.298645	4.298645
2048	4.356709	4.356709

2049 rows × 2 columns

Figure [6]: Shows an absolute match between predicted and true values, a clear indicator of a very good model fit.

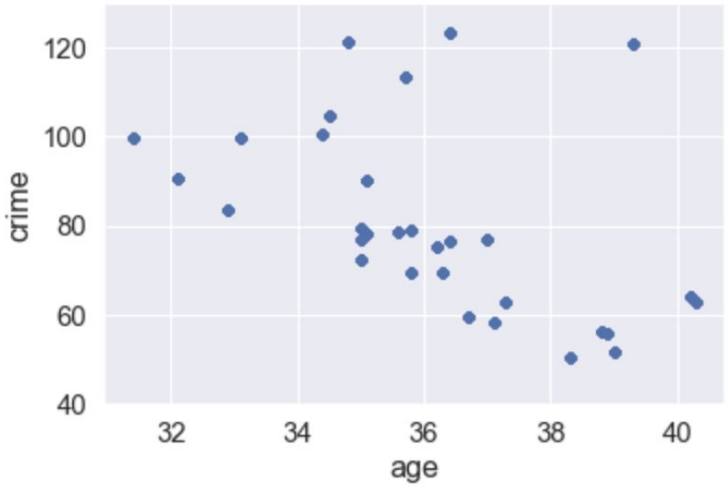


Figure [7]: Scatter plot of age Vs crime generated with matplotlib

Scatter Plots helped find a correlation between crime and age, here we observed an interesting trend that crime and age have a negative correlation, hence confirming individuals tend to commit more crime at a younger age.

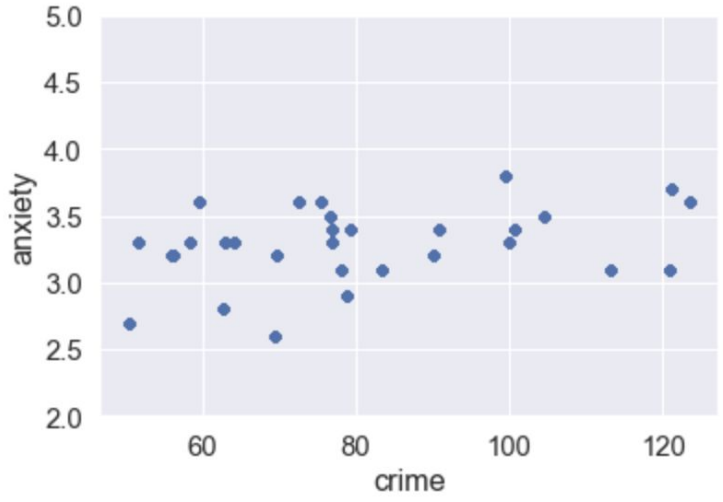


Figure [8]: Scatter plot of crime Vs anxiety generated with matplotlib

Figure 8 shows that as anxiety increases there is a linear increase in crime, hence it can be concluded that in order to help prevent crime the wellbeing and mental health of individuals between the age of 25 - 35 should be of increased priority.

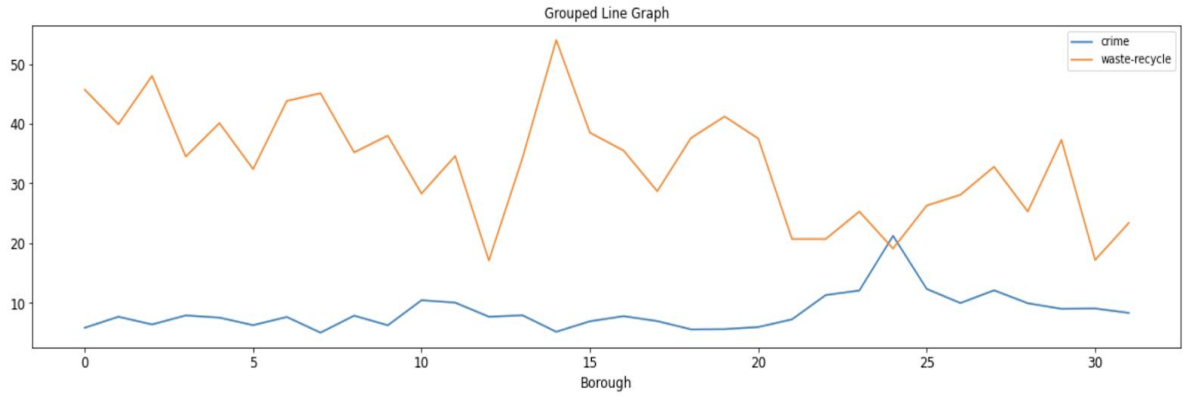


Figure [9]: A plot of crime (blue) and waste recycling (red) by borough

Figure 9 shows the crime rate is high where the waste recycling in the area is particularly low. This is an interesting new trend from which we could conclude that communities which have proper waste management have more responsibility and citizen care, resulting in a decrease in Violent Crime.

Discussion and Conclusion

This project enables the prediction of occurrence of Violent Crime on a postcode basis to a high degree of accuracy based on both RMSE and R^2 scores. Deciding when to stop adding data was a major challenge since we were conscious of developing an overly complex model, in this we were guided with extensive research of similar studies. This enabled us to successfully incorporate the major associative factors of Violent Crime cited by field experts, as well as incorporating a unique stance by finding crimes based on distance from properties rather than looking at entire areas.

Whilst this is accurate for London properties, it is unknown whether this would be the case for other cities. Indeed this study found little to no correlation between Education factors; School Proximity and School Quality with Violent Crime. These findings were surprising and conflicted with existing studies done in different cities or parts of the country. It is likely these findings evidence a generally high standard of education in London. Consequently, in order to be confident in this model when extending it to other cities, the correlation grid should be re-run to confirm the most important variables associated with crime in that city before training.

We are conscious that whilst this research has many commercial applications, some of these could further increase crime in at-risk areas. Using this as a tool on property websites to provide insight to users could drive up house prices in less violent areas, causing an increase in some of the prime associative factors of crime such as job density in more at-risk areas. Instead, we would recommend this tool for Intelligent Policing and for Councils. From this study it is evident that crime cannot be combated by policing alone, instead, we would suggest that greater visibility of contributing factors and how these affect certain areas could act both as a crime prevention tool and to increase visibility into the mental impact of crime and how it uniquely affects different individuals.

Bayesian network structure learning analysis

Q1:

Domain expertise was obtained from extensively researching the subject area by reading previous studies and papers. Initially, this was done from broadly researching 'violent crime in cities' and finding associated factors, these were further examined using more in-depth research papers looking at causation.[1][2][6][7][8] Due to the large breadth of studies examined, we were careful to consider factors only relevant to London (factors such as climate and weapon laws were discounted). An effort was made to use reputable sources. Since knowledge was elicited through sources rather than individuals in the group, we could remain unbiased and avoid disagreements.

Stats from metrics and scoring functions

Structure learning elapsed time: 4 seconds total (Phase 1 = 0 secs, Phase 2 = 0 secs).
_____ Evaluation _____
Nodes: 12
Sample size: 8196
TrueDAG arcs: 22
TrueDAG independencies: 44
LearnedDAG arcs: 43
LearnedDAG independencies: 23
_____ Confusion matrix stats _____
Arcs discovered (TP): 16.0
Partial arcs discovered (TP*0.5): 3.0
False dependencies discovered (FP): 24.0
Independencies discovered (TN): 20.0
Dependencies not discovered (FN): 4.5. [NOTE: # of edges missed is 3.0]
_____ Stats from metrics and scoring functions _____
Precision score: 0.407
Recall score: 0.795
F1 score: 0.538
SHD score: 28.500
DDM score: -0.500
BSF score: 0.250
_____ Inference-based evaluation _____
BIC/MDL score -56641.925
# of free parameters 1307

Q2:

Both graphs feature all our variables. Phase_1 is an initial best guess produced by associated learning. It is an undirected graph which is based on pairwise associational scores ranging from 0 to 1. Higher scores represent greater dependency.

In the Phase_2 the algorithm performs conditional independence tests across all pairs of nodes conditional on the remaining nodes. Each triplet is classified into conditional dependence, independence and insignificance. In Phase_2 constraint-based learning has orientated all the edges in phase_1. If an orientation increases the BIC score, the edge is oriented; otherwise, they remain undirected.

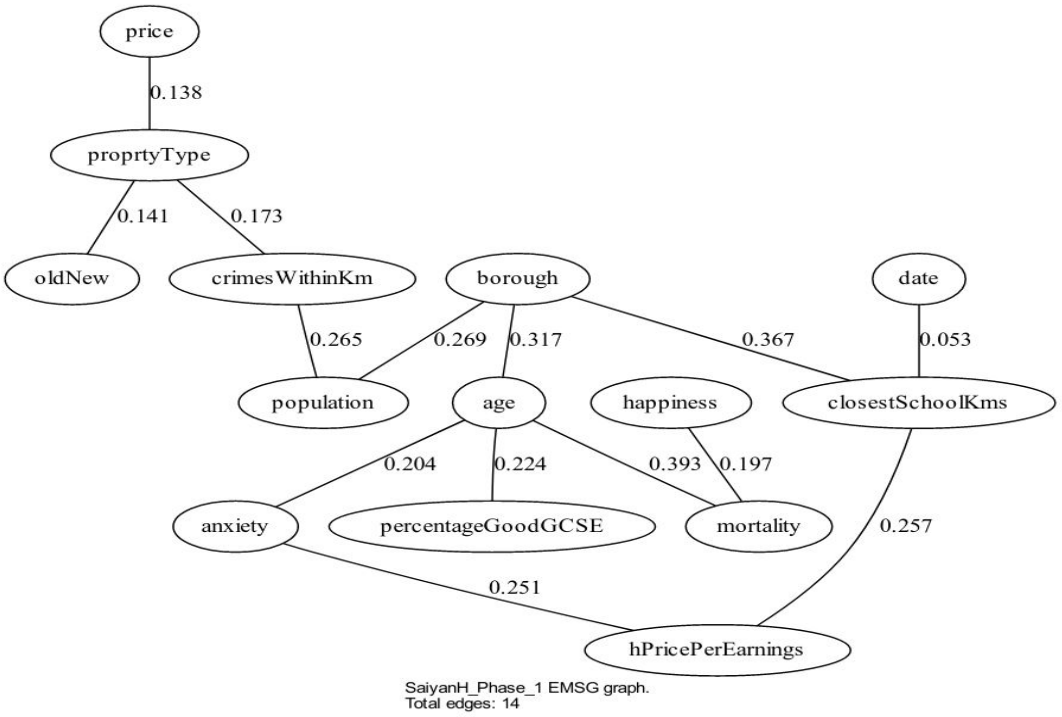


Figure [10]: SaiyanH phase_1 graph

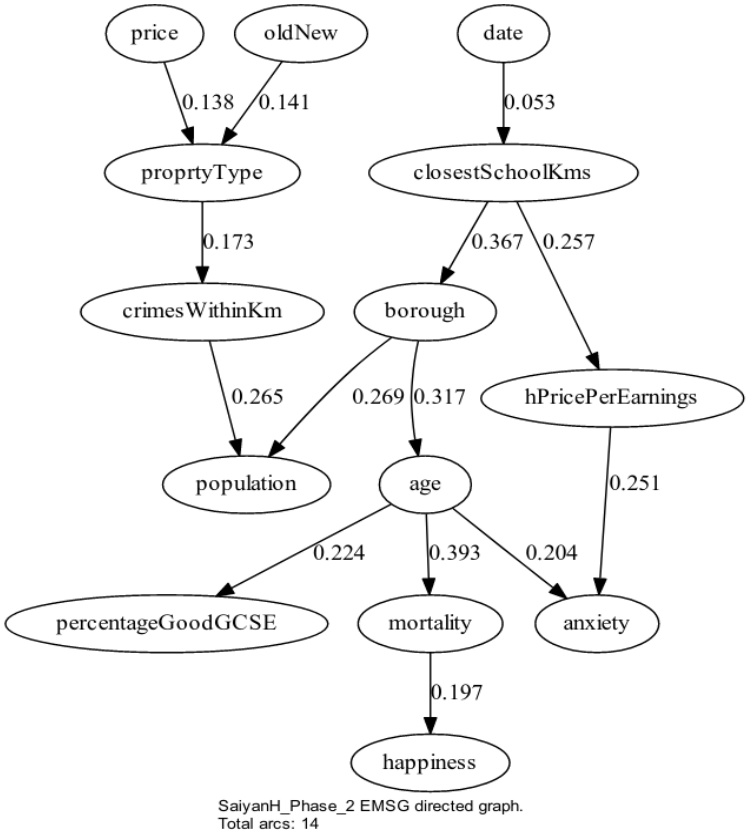


Figure [11]: SaiyanH phase_2 graph

Q3:

Phase_2 results from constraint-based learning which determines the orientation of the edges in phase_1, which is used to prune the search space of possible graphs being explored in phase_3.

Phase_3 uses a search method that explores neighbouring graphs and scoring criteria to evaluate each graph. The output of phase_2 serves as the starting graph for search in phase 3. The algorithm uses the BIC to score graphs. In our phase_3 graph, we see evidence of Hill-Climbing as many edges are reversed and added to the phase_2 graph until no neighbouring graph edge will further increase the BIC score.

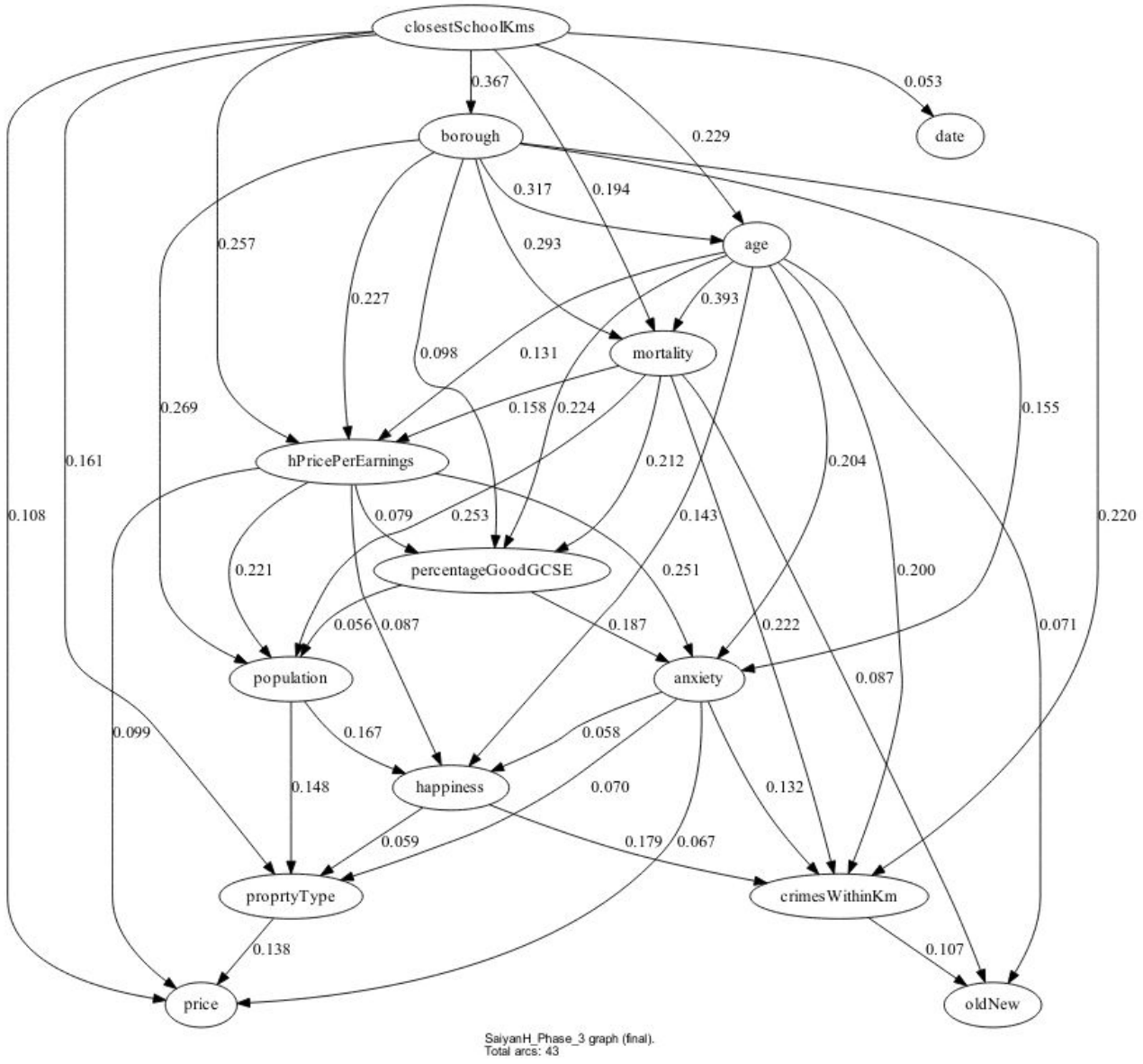


Figure [12]: SaiyanH phase_3

Q4:
CSV scores generated: conditionalIInsignificance.csv '922', conditionalDep.csv '142', marginalDep.csv '91', conditionalIndep.csv '28'.

conditionalInsignifcance.csv holds the associations that don't meet the threshold for conditional dependence/independence criteria in phase_2, therefore it's unsurprising this has significantly more rows since the criteria are relatively conservative. conditionalDep.csv and marginaldependence.csv have a relatively high percentage of the remaining number of rows and correspond to common effect and directly dependent features respectively. We expected conditionalIndep.csv which corresponds to both causal chain and common cause relationships to account for more associations. Perhaps more conditional independent variables exist in the population, but do not hold in our sample.

Q5:
Our data-set is size range 5000-10000, comparable to the Sports and Property studies with 12 nodes, 43 arcs, 1307 free parameters. Our recall and precision score, F1 was 0.538, lower by ~0.2. Comparatively, we would expect an SHD score of 5-20 but ours was higher, 28.500 indicating a less accurate graph. Our BSF confusion matrix score, 0.250 was lower than the expected range of 0.3-0.8. These results were unsurprising since we would expect missing values in a study as broad as crime where not all associative factors can be described. In addition, accuracy was lost in the discretization of our dataset.

Q6:
In the paper we see runtime increase rapidly with the number of nodes and sample size.
Despite an average sample size and number of free parameters, our dataset had a low number of nodes. Therefore the most time-consuming section, phase_2 remains quick due to a low number of triplets. This hypothesis is in line with our runtime, 4 seconds which is consistent with the results in table_2 for like studies where run-times of 1 and 9 seconds are observed. Since our sample size cannot be classified as small, phase_1 is unlikely to be the most time-consuming phase of the algorithm.

Q7:
The BIC/MDL score takes into account model fit and dimensionality since our dataset remained the same size, we can compare the BIC score in steps 2 and 4. Our BIC/MDL value from stage_2 was: -108898.579, after stage_4 this increased dramatically to -56641.925. This is as expected since we know the algorithm works using 'Hill climbing' to increase the BIC score by orienting and adding edges. This is further evidenced by looking at our phase_2 and phase_3 graphs which are also dramatically different (increased edges with changed orientation), since we know any change is only performed if the BIC score increases.

Q8:
A free parameter is one that can be adjusted to make the model fit the data, it is not predefined by the model but is chosen to improve prediction, in the BN this is essentially an edge. In phase_3 the model explores neighbouring graphs and adds to the phase_2 graph, increasing the number of free parameters; in our case dramatically, from 312 after phase_2 to 1307 after phase_4. This is in agreement with our initial expectations as stated above, despite the BIC penalization of free parameters in order to prevent overfitting and preferring simple BN's over more complex ones

References

[1] Aliyu & Muhammad, Maryam & Bukar, Mohammed & Singhry, Ibrahim. (2016). IMPACT OF CRIME ON PROPERTY VALUES: LITERATURE SURVEY AND RESEARCH GAP IDENTIFICATION. African Scholar Journal of Social Science (AJHSS). 4. 68-99

[2]Cooper, K. Stewart, K. (2018). Does Money Affect Children's Outcomes? London. Centre of Analysis of Social Exclusion, Research at LSE ISSN 1460-5023

[3] Lochner, L.(2004) "Education, Work And Crime: A Human Capital Approach," I*nternational Economic Review*, v45(3,Aug), 811-843

[4] Ranapurwala, S. I., Berg, M. T., & Casteel, C. (2016). Reporting Crime Victimizations to the Police and the Incidence of Future Victimizations: A Longitudinal Study. *PloS one*, *11*(7), e0160072. https://doi.org/10.1371/journal.pone.0160072

[5] Variables affecting crimeFBI. (2011) https://ucr.fbi.gov/hate-crime/2011/resources/variables-affecting-crime

[6] Fajnzylber, P. Lederman, D, (2002) What causes violent crime? *European Economic Review* 46 (2002) 13231357

[7]Marvell, T., & Moody, C. (1991). Age Structure and Crime Rates: The Conflicting Evidence. *Journal of Quantitative Criminology, 7*(3), 237-273. www.jstor.org/stable/23365759

[8]Rennó SM, Testa A, Porter LC, Lynch JP (2019) The contribution of age structure to international homicide decline. PLoS ONE 14(10): e0222996. https://doi.org/10.1371/journal.pone.0222996

[9] Department of education, House prices and schools: do houses close to the best performing schools cost more? (2017) https://assets.publishing.service.gov.uk/govrnment/uploads/system/uploads/attachment_data/file/600623/House_prices_and_schools.pdf

[10]Youngme Seo and Robert Simons (*2009*) The Effect of School Quality on Residential Sales Price. Journal of Real Estate Research: 2009, Vol. 31, No. 3, pp. 307-327.

[11] Downes, T. and JEZabel. (2002) The Impact of School Characteristics on House Prices: Chicago 1987–1991. Journal of Urban Economics, 2002, 52, 1–25.

[12]Janko, Z. and Popli, G. (2015) Examining the link between crime and unemployment: a time-series analysis. Applied Economics, 47 (37). pp. 4007-4019. ISSN 0003-6846

[13] Wes, M., 2017. *Python for Data Analysis*. 2nd ed. US: O'REILLY

[14] Kennedy, B P. Kawachi, I. Prothrow-Stith, D. Lochner, K. Gupta, V. (1998) Social capital, income inequality, and firearm violent crime. Social Science & Medicine, 47(1).7-17. ISSN 0277-9536

Additional graphs

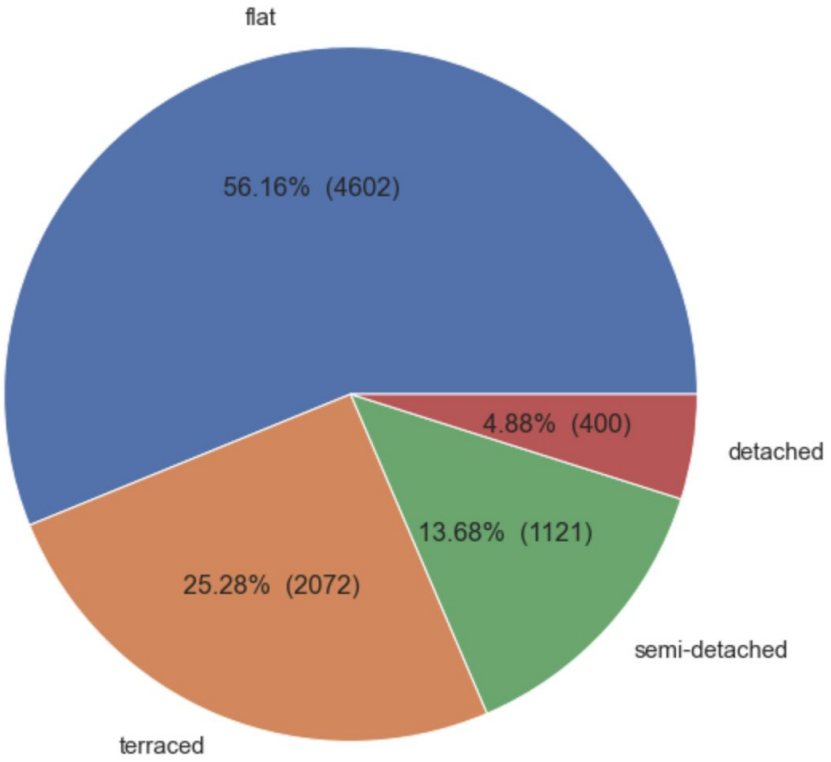
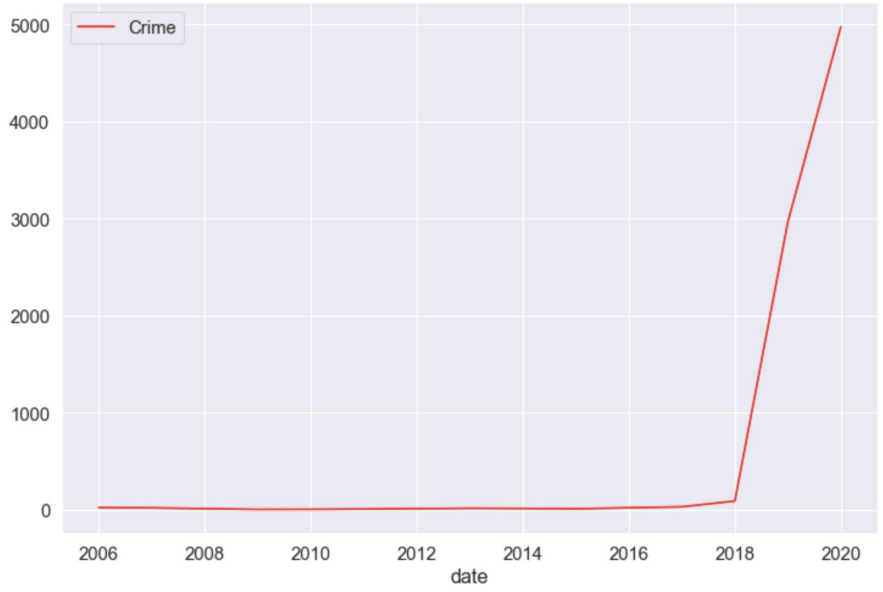
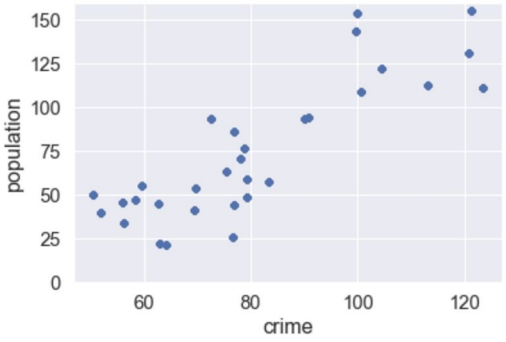
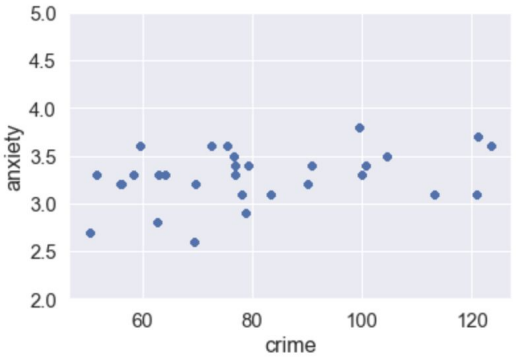


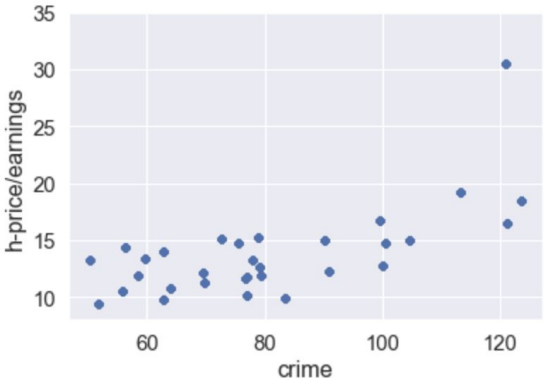
Figure [13]: Pie chart of London resident type

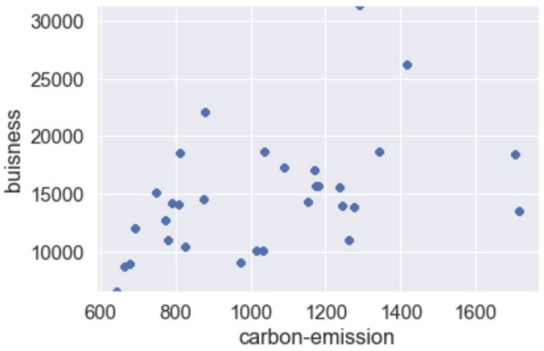


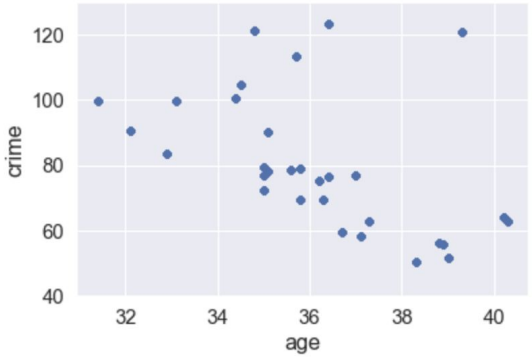
Figure[14]: Line Graph with grids, crime in years 2016 -2020

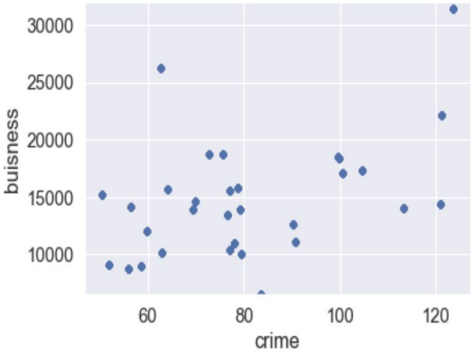


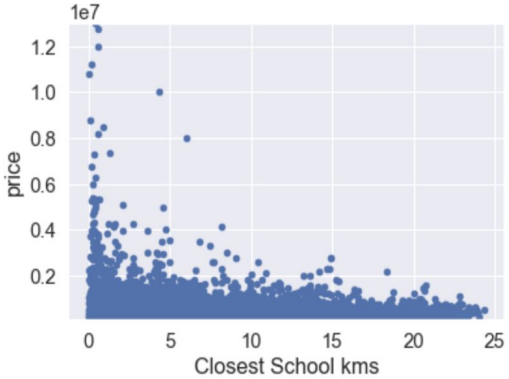












Figures [15]-[21]: Additional scatter plots of interest

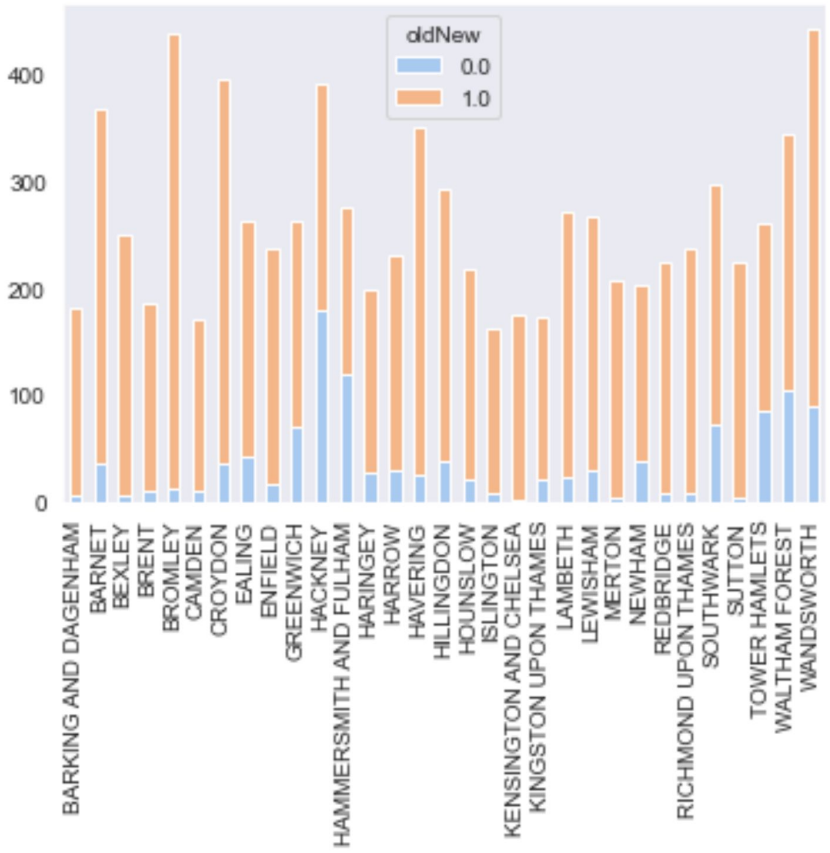


Figure [22]: Borough Bar Graph of proportion new (orange) and old (blue) residencies

Code Appendices

- Importing libraries

```
import pandas as pd
import geopandas as gpd
from pandas import read_excel
import numpy as np
import datetime
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import skew,norm
from scipy.stats.stats import pearsonr
import seaborn as sns
from functools import reduce
import seaborn as sns
from functools import reduce
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn import linear_model

%matplotlib inline
```

- Importing CSV's

```
colnames = ['ref', 'price', 'date','postcode', 'proprty type', 'oldNew', 'hold', 'house num', 'flat num', 'road'
df = pd.read_csv("pp-monthly.csv", names = colnames, header=None)

#https://data.london.gov.uk/dataset/london-borough-profiles
df_profile = pd.read_csv("london-borough-profiles.csv", header=0,encoding = 'unicode_escape')

#https://data.london.gov.uk/dataset/ratio-house-prices-earnings-borough
df_price_to_earn = pd.read_csv("median-earnings-to-price-ratio.csv", header=0,encoding = 'unicode_escape')

#https://data.london.gov.uk/dataset/gcse-results-by-borough
df_gcse_by_borough = pd.read_csv("gcse-results.csv", header=0,encoding = 'unicode_escape')

#https://data.london.gov.uk/dataset/gcse-results-by-borough
df_uk_postcodes_lat_long = pd.read_csv("ukpostcodes.csv", header=0,encoding = 'unicode_escape')

#https://www.compare-school-performance.service.gov.uk/download-data
df_uk_schools=pd.read_csv("School.csv", index_col=0, header=None).T
```

- Function to calculate distance from latitude and longitudes

```
from math import sin, cos, sqrt, atan2, radians

# approximate radius of earth in km
R = 6373.0

#############################################################################################################
#
# Function that returns distance in km given two latitude longitude coordinates
# (used to obtain distance from schools)
#
#############################################################################################################

def calculateDistance(lat1,long1,lat2,long2):
    lat1 = radians(lat1)
    lon1 = radians(long1)
    lat2 = radians(lat2)
    lon2 = radians(long2)

    dlon = lon2 - lon1
    dlat = lat2 - lat1

    a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
    c = 2 * atan2(sqrt(a), sqrt(1 - a))

    distance = R * c

    return distance
```

- Adding latitudes and longitudes to schools

```
df_uk_schools  = df_uk_schools.rename(index=str, columns={"POSTCODE":"postcode"})

#Merge latitude and longitude on postcode to give latitude and longitudes of desired postocodes

df_uk_schools=pd.merge(df_uk_postcodes_lat_long, df_uk_schools, on="postcode")

df_uk_schools.to_csv ('newSchools.csv', index = False, header=True)

df_uk_schools
```

-Removing houses not in London and adding latitude and longitudes of each house

```
#Get rid of all house data not from london boroughs
borough_list = ['CITY OF LONDON', 'BARKING AND DAGENHAM', 'BARNET', 'BEXLEY',
      'BRENT', 'BROMLEY', 'CAMDEN', 'CROYDON', 'EALING', 'ENFIELD',
      'GREENWICH', 'HACKNEY', 'HAMMERSMITH AND FULHAM', 'HARINGEY',
      'HARROW', 'HAVERING', 'HILLINGDON', 'HOUNSLOW', 'ISLINGTON',
      'KENSINGTON AND CHELSEA', 'KINGSTON UPON THAMES', 'LAMBETH',
      'LEWISHAM', 'MERTON', 'NEWHAM', 'REDBRIDGE', 'RICHMOND UPON THAMES',
      'SOUTHWARK', 'SUTTON', 'TOWER HAMLETS', 'WALTHAM FOREST',
      'WANDSWORTH', 'WESTMINSTER']

df = df[df['borough'].isin(borough_list)]


df.borough.unique()

Merge latitude and longitude on postcode to give latitude and longitudes of desired postocodes

df=pd.merge(df_uk_postcodes_lat_long, df, on="postcode")
```

Find the distance from the closest school from each house

```
number_of_houses = len(df.index)
number_of_schools = len(df_uk_schools.index)


for y in range (0,number_of_houses):
    minDistance = 100
    lat1 = df.iloc[y]['latitude']
    long1 = df.iloc[y]['longitude']
    for x in range (0,number_of_schools):

        lat2 = df_uk_schools.iloc[x]['latitude']
        long2 =df_uk_schools.iloc[x]['longitude']
        distance = calculateDistance(lat1,long1,lat2,long2)
        if minDistance>distance:
            minDistance = distance

    df.loc[df.index[y], 'Closest School kms'] = minDistance

df.to_csv ('newpp_monthly.csv', index = False, header=True)
```

-Isolate % good GCSE results of mixed-sex schools by Borough

```
df_gcse_by_borough = df_gcse_by_borough.loc[df_gcse_by_borough["Sex"] == "All"]
df_gcse_by_borough = df_gcse_by_borough.loc[df_gcse_by_borough["Year"] == "2018/19"]
df_gcse_by_borough = df_gcse_by_borough[["Area","Attainment8"]]
df_gcse_by_borough = df_gcse_by_borough.head(33)


df_gcse_by_borough  = df_gcse_by_borough .rename(index=str, columns={"Area":"borough", "Attainment8": "Percentag
df_gcse_by_borough ['borough'] = df_gcse_by_borough ['borough'].str.upper()

df_gcse_by_borough.head()
```

- Add house price per earning per Borough

```
df_price_to_earn = df_price_to_earn[["Area","2016"]]
df_price_to_earn = df_price_to_earn.rename(index=str, columns={"Area":"borough", "2016": "h-price/earnings" })
df_price_to_earn['borough'] = df_price_to_earn['borough'].str.upper()
df_price_to_earn.head()
```

Convert borough to upper case

```
df_main['borough'] = df_main['borough'].str.upper()
```

Removed property type ‘other’

```
df.drop(df[ df['proprty type'] == 'O' ].index , inplace=True)
```

Checking for null values and removing them

```
# removing all rows having null value
df = df.dropna(how='any',axis=0)

# check if any null value exists
df.isnull().sum()
```

Import crime and isolate ‘Violent crime and sexual assault’

```
df_crime= pd.read_csv("mayCrime.csv")
df_crime

examinedCrimes = ['Violence and sexual offences']
df_crime = df_crime[df_crime['Crime type'].isin(examinedCrimes)]


df_crime.reset_index(drop=True)
```

Find number of crimes within 3 km of each property

```
number_of_houses = len(df_section.index)
number_of_crimes = len(df_crime.index)


for y in range (0,number_of_houses):
    counter = 0
    lat1 = df_section.iloc[y]['latitude']
    long1 = df_section.iloc[y]['longitude']
    for x in range (0,number_of_crimes):

        lat2 = df_crime.iloc[x]['Latitude']
        long2 =df_crime.iloc[x]['Longitude']
        distance = calculateDistance(lat1,long1,lat2,long2)
        if distance<1:
            counter = counter +1
    df_section.loc[df_section.index[y], 'Crimes within 1 km'] = counter

df_section.to_csv ('houses_with_crimes.csv', index = False, header=True)
```

Final Dataset Data Types

```
df.dtypes
```

```
borough                  object
buisness                  int64
crime                   float64
average-age             float64
Inner/_Outer_London      object
employment%             float64
life-satisfaction       float64
waste-recycle           float64
gross_annual_pay         object
unemployment            float64
carbon-emission         float64
greenspace%             float64
price                     int64
date                      int64
proprty type            float64
oldNew                    int64
Closest School kms      float64
Crimes within 1 km      float64
population              float64
age                     float64
happiness               float64
anxiety                 float64
mortality                 int64
h-price/earnings        float64
Percentage_Good_GCSE    float64
dtype: object
```

Final Dataset first 5 rows

```
pd.set_option('display.max_columns', None)
df.head()
```

	borough	buisness	crime	average-age	Inner/_Outer_London	employment%	life-satisfaction	waste-recycle	gross_annual_pay	unemployment	carbon-emission	greenspac
0	BARKING AND DAGENHAM	6560	4.435567	32.9	Outer London	65.8	7.1	23.4	27886	11.0	644.0	
1	BARKING AND DAGENHAM	6560	4.435567	32.9	Outer London	65.8	7.1	23.4	27886	11.0	644.0	
2	BARKING AND DAGENHAM	6560	4.435567	32.9	Outer London	65.8	7.1	23.4	27886	11.0	644.0	
3	BARKING AND DAGENHAM	6560	4.435567	32.9	Outer London	65.8	7.1	23.4	27886	11.0	644.0	
4	BARKING AND DAGENHAM	6560	4.435567	32.9	Outer London	65.8	7.1	23.4	27886	11.0	644.0	

```
pd.set_option('display.max_columns', None)
df.head()
```

bon-ssion	greenspace%	price	date	proprty type	oldNew	Closest School kms	Crimes within 1 km	population	age	happiness	anxiety	mortality	h-price/earnings	Percentage_Good_GCSE
644.0	33.6	277000	2020	0.5	1	15.149416	82.0	57.9	32.9	7.1	3.1	228	9.89	46.4
644.0	33.6	160000	2016	0.1	1	15.404252	92.0	57.9	32.9	7.1	3.1	228	9.89	46.4
644.0	33.6	335000	2020	0.7	1	15.642070	80.0	57.9	32.9	7.1	3.1	228	9.89	46.4
644.0	33.6	285500	2020	0.5	1	15.112509	61.0	57.9	32.9	7.1	3.1	228	9.89	46.4
644.0	33.6	221000	2020	0.1	1	16.021886	46.0	57.9	32.9	7.1	3.1	228	9.89	46.4

Plotting Correlation Matrix

```
#correration matrix
corrmat1 = df_main.corr()
f, ax = plt.subplots(figsize=(15, 9))
sns.heatmap(corrmat1, square=True)
plt.show();
```

Feature scaling

```
# Feature selction process based on high correlation value,
# after some testing we have decided to keep these 10 features which give us a perfect model
cols = corrmat.nlargest(10, 'crime')['crime'].index
cm = np.corrcoef(df[cols].values.T)
sns.set(font_scale=1.35)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f', annot_kws={'size': 10}, yticklabels=cols.values,
                 xticklabels=cols.values)
plt.yticks(rotation=0)
plt.xticks(rotation=90)
plt.show()
```

Kernel Density Plotting

```
#kernel density plot
sns.distplot(df.crime,fit=norm);
plt.title = ('Crimes Normal Distribution');
#fitted parameters used by the function
(mu, sigma) = norm.fit(df['crime']);
#QQ plot
fig = plt.figure()
res = stats.probplot(df['crime'], plot=plt)
plt.show()
print("skewness: %f" % df['crime'].skew())
print("kurtosis: %f" % df['crime'].kurt())
```

Logarithmic transformation on crime

```
# logarithmic transformation to make highly skewed distributions less skewed

df["crime"] = np.log1p(df_test["crime"])
```

Splitting dataset as Training and Testing data

```
X_train, X_test, y_train, y_test = train_test_split(X_crimes,y_crimes, test_size = 0.25, random_state =5)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(6146, 10)
(2049, 10)
(6146,)
(2049,)
```

Model Evaluation and Performance

```
# Model evaluation for training set
y_train_predict = lm_crimes.predict(X_train)
rmse = (np.sqrt(mean_squared_error(y_train, y_train_predict)))
r2 = r2_score(y_train, y_train_predict)

print("The model performance for training set")
print("--------------------------------------")
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
print("\n")

# Model evaluation for testing set
y_test_predict = lm_crimes.predict(X_test)
rmse = (np.sqrt(mean_squared_error(y_test, y_test_predict)))
r2 = r2_score(y_test, y_test_predict)

print("The model performance for testing set")
print("--------------------------------------")
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
```

```
The model performance for training set
--------------------------------------
RMSE is 1.1466007598398963e-15
R2 score is 1.0


The model performance for testing set
--------------------------------------
RMSE is 1.1986310075941976e-15
R2 score is 1.0
```

Scatter plot crime vs age

```
#scatter plot Crime/Age
var = 'age'
data =pd.concat([df['crime'], df[var]], axis=1)
data.plot.scatter(x=var,y="crime", ylim= (40,130));
plt.show()
```

Pie chart of price and house type

```
def make_autopct(values):
    def my_autopct(pct):
        total = sum(values)
        val = int(round(pct*total/100.0))
        return '{p:.2f}%  ({v:d})'.format(p=pct,v=val)
    return my_autopct

fig = plt.figure()
plt.figure(figsize=[10,10])

values = df["proprty type"].value_counts()
plt.pie(values, labels=["0.1","0.5","0.7","1.0"] , autopct=make_autopct(values), pctdistance=0.6, labeldistance=1.1)
```

Crime and date Line Graph

```
#group by on crime and date
data = df.groupby("date")[["crime"]].count()
data.columns = ["Crime"]
data.reset_index(level=0, inplace=True)
data.plot(x ='date', y="Crime", kind = 'line',color='red',figsize=(12, 8))
plt.show()
```

Loading London Geodata shapefile

```
# set the filepath and load in a shapefile
fp = "london-map/ESRI/London_Borough_Excluding_MHW.shp"
map_df = gpd.read_file(fp)
map_df.head()
```

	NAME	GSS_CODE	HECTARES	NONLD_AREA	ONS_INNER	SUB_2009	SUB_2006	geometry
0	Kingston upon Thames	E09000021	3726.117	0.000	F	None	None	POLYGON ((516401.600 160201.800, 516407.300 16...
1	Croydon	E09000008	8649.441	0.000	F	None	None	POLYGON ((535009.200 159504.700, 535005.500 15...
2	Bromley	E09000006	15013.487	0.000	F	None	None	POLYGON ((540373.600 157530.400, 540361.200 15...
3	Hounslow	E09000018	5658.541	60.755	F	None	None	POLYGON ((521975.800 178100.000, 521967.700 17...
4	Ealing	E09000009	5554.428	0.000	F	None	None	POLYGON ((510253.500 182881.600, 510249.900 18...

Plotting Happiness in London Choropleth map

```
# setting the column we want to visualise on the map
variable = "happiness"
# range for the choropleth
vmin, vmax = 0, 10
# create figure and axes for Matplotlib
fig, ax = plt.subplots(1, figsize=(30, 11))


ax.axis("off")

ax.set_title('Happiness in London', fontdict={'fontsize': '24'})

# create map
sm = mdf.plot(column=variable, cmap="YlGnBu", missing_kwds={"color": "lightgrey","edgecolor": "red","hatch": "///","lab


# create colorbar as a legend
sm = plt.cm.ScalarMappable(cmap="YlGnBu", norm=plt.Normalize(vmin=vmin, vmax=vmax))
# empty array for the data range
sm._A = []
# add the colorbar to the figure
cbar = fig.colorbar(sm)

# Add Labels
mdf['coords'] = mdf['geometry'].apply(lambda x: x.representative_point().coords[:])
mdf['coords'] = [coords[0] for coords in mdf['coords']]
for idx, row in mdf.iterrows():
    plt.annotate(s=row['borough'], xy=row['coords'],horizontalalignment='center', color='grey')
```

Grouped Bar Graph Crime vs Unemployment

```
ax = mdf[['crime','employment%']].plot(kind='bar', title ="Grouped Bar Graph", figsize=(20, 5), legend=True, fontsize=1
ax.set_xlabel("Borough", fontsize=12)
ax.set_xticklabels(mdf['borough'])
plt.show()
```