Data collection:

Sources per section:

**Why Random Forests?**

Lara, G.L. and Gurulingappa, H. (2017). Random Forest Ensemble Algorithm for AUV Classification. International Journal of Engineering and Technology, 8(2.12), pp.195-199.

Schonlau, M. and Zou, R.Y. (2020). The random forest algorithm for statistical learning. *The Stata Journal: Promoting communications on statistics and Stata*, 20(1), pp.3–29. doi:10.1177/1536867x20909688.

**How did we clean our datasets?**

pandas.pydata.org. (n.d.). *pandas.melt — pandas 1.3.2 documentation*. [online] Available at: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.melt.html.

‌

**Optimizing the Regressor score**

scikit-learn (2018). *3.2.4.3.2. sklearn.ensemble.RandomForestRegressor — scikit-learn 0.20.3 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.

‌

**Importance score per attribute**

Płoński, P. (2020). *The 3 Ways To Compute Feature Importance in the Random Forest*. [online] Medium. Available at: https://medium.com/towards-data-science/the-3-ways-to-compute-feature-importance-in-the-random-forest-96c86b49e6d4 [Accessed 8 Jan. 2023].

‌scikit-learn.org. (n.d.). *3.2.4.3.2. sklearn.ensemble.RandomForestRegressor — scikit-learn 0.23.2 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#sklearn-ensemble-randomforestregressor.

‌Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

Yu, Tian-Jian, et al. "Performance Mining: Automatically Diagnosing Performance Issues in Production ML Pipelines." arXiv preprint arXiv:1811.02125 (2018). chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://arxiv.org/pdf/2108.13557.pdf

**Difference in results between the models**

Hagan, John, et al. "An International Perspective on Serious Property Crime." Crime & Delinquency, vol. 56, no. 3, 2010, pp. 293-321.

**Most important borough attributes that predict crime rates**

‌ Levitt, S., & Dubner, S. (2005). Freakonomics: A Rogue Economist Explores the Hidden Side of Everything. HarperCollins.

**Limitations**

Brieman, Leo, et al. "Random Forests." Machine Learning, vol. 45, no. 1, April 2001, pp. 5-32.

Salzer, Csaba. "Feature importance in random forest." Towards Data Science, 20 May 2017, https://towardsdatascience.com/feature-importance-in-random-forest-234df8cc1eed.

Donges, N. (2021). *A Complete Guide to the Random Forest Algorithm*. [online] Built in. Available at: https://builtin.com/data-science/random-forest-algorithm.

**Why Random Forests?**

For our main data analysis method, we have chosen Random Forests. Random Forests is a machine learning technique that creates a collection of decision trees, or "forests", and utilizes them to generate predictions. It fits several decision tree classifiers to various subsets of the data and combining the results (Lara and Gurulingappa, 2017). This is done by a process called “bagging” (Schonlau and Zou, 2020). Random Forests may be used for both classification tasks (such as classifying results) and regression tasks. Additionally, Random Forests can adapt to nonlinearities in the data, and work well with medium to large datasets. Another benefit to using Random forests is that it can help mitigate the problem of overfitting through the “bagging” process, which is a common issue with decision trees. This helps to reduce overfitting and improve the generalization of the model (Schonlau and Zou, 2020).

For our project, which aims to identify the most important indicators of London boroughs that contribute to crime rate, it makes sense to use random forest regression. We chose regression instead of classification as our model is trying to predict crime rates, which is a continuous variable. If we were predicting a categorical variable, we would use random forest classification. The model uses a range of attributes of a London Borough, to predict the crime rate. To do this, the algorithm identifies which factors are most influential in contributing to the crime rate per borough, which helps us answer our research question. The random forest algorithm is well suited to this type of regression task, and its ability to handle nonlinear relationships makes it a good choice for analysing the complex factors that may influence crime rate in London boroughs.

‌

**Visualization of how decision trees work for Random Forest**

To visualize how Random Forest’s work, here is an example of one of the decision trees out of 110 decision trees created by our Random Forest regression:

A screenshot of a computer

Description automatically generated with low confidence

The diagram shows the nodes, branches, and decisions made at each node. At each node of the tree, a decision is made based on the value of one of the borough attributes, and the data is partitioned into two or more subsets based on the decision. The process continues until the subsets are pure, meaning that they contain only one class of output values. The diagram also shows the resulting squared error, number of samples, and value. The squared error is a measure of the difference between the predicted values and the actual values, and it is used to evaluate the performance of the model at each node. The number of samples is the number of data points that reach each node, and the value is the mean of the output values (in this case, the crime rate) for the samples at that node. In this case the “output value” refers to crime rate. As we can see each terminal node (the nodes that have no child nodes), the “squared error” is always 0. The model stops building trees at this point because the squared error being 0 indicates that the model is making perfect predictions for the data. In other words, the predicted crime rate values are exactly the same as the actual crime rate values, so there is no need to continue building trees.

**How did we choose our attributes per borough?**

How and where did we collect our data?

**How did we clean our datasets?**

The main goal for the data cleaning process was to form a dataset that could be used by our random forest. To do this, we need a large data set where each row refers to a unique borough and year, with columns that contain all the relevant borough attribute’s, such as “population per hectare”, or “wage per hour”. The method for this was making several individual datasets for each borough attribute, where the three columns were “Borough”, “Year”, and relevant borough attribute/s. It is important to note that the “Borough” and “Year” columns form a primary key together within each dataset. Ultimately, to make the large dataset for the random forest, we performed a merge across all these individual datasets.

To achieve this there were several rudimentary practices that we had to do during the data cleaning process. These practices include converting excel to csv files, removing unnecessary data, changing the datatypes of columns, renaming columns for effective data frame merges, reshaping data frames, and data standardization.

As previously mentioned, most of our data was collected from the London data store website. The data on the website was always in the form of an Excel or CSV files. If the file was an Excel spreadsheet, we had to convert it to a CSV file. Most of the time, there was unnecessary data in these excel files, which is why we had to select the relevant column and rows manually from these files. Also, though most of the data manipulation was done in Python, if there was a task that more efficiently completed in Excel, it was done in Excel. Once the csv files were ready, they were loaded into Python using the pandas function read\_csv(). An an example of a common row we had to remove from the datasets using python were rows including “City of London”. Some datasets considered “City of London” to be a borough, even though it is not officially considered one which is why we had to remove it. Below is an example of how we removed these rows from the Average GCSE scores per borough dataset:

#Remove the City of London rows

gcse\_p\_borough = gcse\_p\_borough.loc[~(gcse\_p\_borough['Area'] == 'City of London')]

gcse\_p\_borough

Another procedure was changing the datatypes of our columns. Sometimes our data that should have been integer or float types, were strings. This is a problem because arithmetic operations can’t be performed on strings. To change this, we had to implement code similar to the following example:

list\_of\_dfs = [df\_crime\_rate, df\_perecentage\_j\_p\_sector\_1999\_2020, df\_dwellings\_p\_hectare\_2000\_2019, df\_homlessness\_2005\_2018, df\_gcse\_p\_borough, df\_happiness\_2012\_2019, df\_anxiety\_2012\_2019, df\_worthwhile\_score\_2012\_2019, df\_life\_satisfaction\_2012\_2019, df\_Ratio\_House\_Prices\_to\_Earnings\_Borough\_2002\_2021, df\_wage\_hour\_per\_borough\_2002\_2021,df\_pop\_density, life\_expectancy\_at\_65\_men\_2000\_2018, life\_expectancy\_at\_65\_women\_2000\_2018, percentage\_public\_sector\_2009\_2019, percentage\_private\_sector\_2009\_2019,votes\_2014,alcoholic\_beverage\_expenditure\_2000\_2021,pubs\_winebars\_expenditure\_2000\_2021]

for df in list\_of\_dfs:

for column in df:

if column == 'Borough' or column == 'Year':

df[column] = df[column].astype(str)

else:

df[column] = df[column].astype(float)

This code first defines a list called list\_of\_dfs that contains all the borough attribute’s data frames. It then uses a for loop to iterate through each data frame in the list. For each data frame, it uses another for loop to iterate through each column. If the column is named "Borough" or "Year", it converts the datatype of the column to a string using the astype() function. If the column is not named "Borough" or "Year", it converts the datatype of the column to a float. We needed to ensure that all of the attribute columns in the dataset were float types. This is because the model may interpret columns of data as categorical if they are strings, rather than numerical, if they are not explicitly cast as floats or integers.

When merging dataframes, it is important to ensure that the columns need to be merged on have the same name in all dataframes. If the column names are different, the merges don’t execute properly. The final merge was done on “Borough” and “Year”, hence all datasets had to have their respective borough and year column labelled specifically as the string “Borough” and “Year”. Here is an example where we renamed the column “Name” to “Borough” for the Population per hectare dataset:

#Rename Area name to borough

df\_pop\_density = df\_pop\_density.rename(columns={'Name': 'Borough'})

Doing this for datasets where it was necessary to, ensured for an effective merge of datasets. Here is the code that was used to perform an inner merge on all of the datasets, with the exception of the votes per borough datasets:

list\_of\_dfs\_no\_votes=[df\_dwellings\_p\_hectare\_2000\_2019, df\_homlessness\_2005\_2018, df\_gcse\_p\_borough, df\_happiness\_2012\_2019, df\_anxiety\_2012\_2019, df\_worthwhile\_score\_2012\_2019, df\_life\_satisfaction\_2012\_2019, df\_Ratio\_House\_Prices\_to\_Earnings\_Borough\_2002\_2021, df\_wage\_hour\_per\_borough\_2002\_2021,df\_pop\_density, life\_expectancy\_at\_65\_men\_2000\_2018, life\_expectancy\_at\_65\_women\_2000\_2018, percentage\_public\_sector\_2009\_2019, percentage\_private\_sector\_2009\_2019,alcoholic\_beverage\_expenditure\_2000\_2021,pubs\_winebars\_expenditure\_2000\_2021]

df = df\_crime\_rate.merge(df\_perecentage\_j\_p\_sector\_1999\_2020, on=['Borough','Year'])

for dataframe in list\_of\_dfs\_no\_votes:

df= df.merge(dataframe, on=['Borough', 'Year'])

We needed to reshape several of the data frames so that they had desired consistent column structure of “Borough”, “Year”, and relevant borough attribute/s, to ultimately facilitate the merge. Here is an example with the Homelessness per 1000 data set:

# Reshape the DataFrame using the melt function

df\_homlessness\_2005\_2018 = df\_homlessness\_2005\_2018.melt(id\_vars=['Borough'], var\_name='Year', value\_name='Average number of homeless people per 1000')

Initially the data was arranged in a wide format with multiple columns, where each column represented a different year and the values in the columns represented the average number of homeless people per 1000 in each borough for that year. However, this code uses the melt() function to reshape the data frame (pandas.pydata.org, n.d.). The id\_vars parameter specifies the columns that should be kept as-is in the reshaped data frame. In this case, the "Borough" column is specified as an id variable, so it will be kept as a separate column in the reshaped dataframe. The var\_name parameter specifies the name of the new column that will be created to hold the old column names. In this case, the new column will be called "Year". Finally, the value\_name parameter specifies the name of the new column that will be created to hold the old column values. In this case, the new column will be called "Average number of homeless people per 1000". Restructuring the datasets to look like this was very important to execute the final merge.

Finally, we utilized data standardization as a method of modifying our data in order to make it more comparable and easier to analyze. This involved expressing certain data as a percentage of a whole or as a ratio or proportion relative to some other value. For instance, we had data on the number of jobs per sector per borough. For the random forest to compare the proportions of workers in each sector across boroughs, we standardized the data by expressing the number of jobs per sector as a percentage of the total number of workers in each borough. This process was done for other borough attributes.

Overall, the data cleaning process was an essential part of our analysis, as it allowed us to transform and standardize the data to make it more suitable for use with our random forest model. While the methods mentioned were some of the most important techniques we used during the data cleaning process, they were not the only methods we employed. All of these techniques were learned during lectures and workshops and were vital for creating the dataset.

**Challenges faced when collecting data?**

The main challenges we faced was the availability of data. There were many different datasets available for different years, but in the end, the only three years where almost all our borough attributes had overlapping data was 2016-2018. To address this issue, we had to create a dataset where we imputed the missing data with statistically relevant values. In this case, we chose to impute the mean value per borough, per borough attribute. For example, if the data for "Life expectancy of women at 65" was missing for Brent in 2021, we used the mean "Life expectancy of women at 65" for Brent for all the years where the data was available to fill in the missing value for 2021. This was done by grouping the data by borough and calculating the mean of the values in each column, and then iterating through the rows and columns of the original dataset to impute the missing values with the mean value for each borough. This process is shown in this code excerpt:

# Group the data by borough

grouped = df\_1999\_2021.groupby("Borough")

# For each group, calculate the mean of the values in each column

mean\_by\_borough = grouped.mean()

# Iterate through the rows of df\_1999\_2021

for index, row in df\_1999\_2021.iterrows():

# Iterate through the columns of the row

for col in df\_1999\_2021.columns:

# If the value is NaN, impute the mode instead of the mean

if pd.isnull(row[col]):

variable = df\_1999\_2021[col]

df\_1999\_2021.loc[index, col] = mean\_by\_borough.loc[row['Borough'], col]

Imputing the mean value can be a good approach when the data is generally relatively similar to the mean value and can be a reasonable estimate of the missing values. We also considered imputing the mode, instead of the mean value. However, we decided not to because all of our data is continuous, and imputing mode is typically used for categorical data. Though imputing the mean value is e a simple approach, it can also reduce the variability in the data, ultimately leading to a less accurate random forest model. This is because imputing the mean value for missing values can reduce the overall variation in the data, which can affect the model's ability to capture accurate patterns and relationships in the data.

Given these considerations, we ultimately decided to run two random forest models: one on the dataset that had no imputed values (years 2016-2018), and the other on the dataset with the imputed values. We will then compare our findings in both through our analysis.

**Optimizing the Regressor score**

To evaluate our Random Forest Regression performance, we used the “.score()” method. The random forest regressor score method returns the coefficient of determination (R2), which is a statistical measure of how well the model's predictions match the true labels.

R2 equals to 1 - (residual sum of squares / total sum of squares). The residual sum of squares is the sum of the squared differences between the true crime rates and the predicted crime rates, while the total sum of squares is the sum of the squared differences between the true crime rates and the mean of the true crime rates. A model with a high R^2 score is making more accurate predictions, while a model with a low R^2 score is making less accurate predictions. The best possible score is 1.0, which indicates a perfect prediction, while a score of 0.0 indicates that the model is no better than just guessing the mean value of the true crime rates. Thus, as the score of the model increases, the model is making more accurate predictions (scikit-learn, 2018).

Hence, this suggests that the borough-attribute importance scores are more reliable if they are based on the best performing model, which occurs when the regressor score is optimized. However, there isn’t a direct way to “optimize” the regressor score. Instead, we ran the random forest model 10,200 times, each time recording it’s regressor score, to ultimately chose the run that had the highest regressor score. The process by which we did this can be seen on our GitHub. The model that had the highest regressor score were the results we used for analysis. Though the main aim of running the random forest model 10,200 times was to optimize the regressor score, it also gave us insight in which dataset performed better for the Random Forest regression model.

|  |  |
| --- | --- |
|  |  |

|  |  |  |
| --- | --- | --- |
|  | Regressor Score 2016-2018 | Regressor Score 1999-2021 |
| Minimum | 0.34 | 0.26 |
| Quartile 1 | 0.74 | 0.89 |
| Mean | 0.8 | 0.9 |
| Quartile 3 | 0.89 | 0.93 |
| Maximum | 0.97 | 0.96 |
| Interquartile range (IQR) | 0.15 | 0.04 |

Table 1- Summary statistics for Regressor Score’s for 2016-2018 data compared to 1999-2021 data

The boxplots above visualize the regressor scores for the 10,200 iterations of our random forest model, for both the 2016-2018 dataset, and the 1999-2021 dataset. The summary statistics for each boxplot can be in the table. As we can see the 1999-2021 dataset had a higher mean score and a lower IQR compared to the 2016-2018 dataset. A lower IQR means the data is more tightly distributed around the median. The visualizations demonstrate that the model performed more effectively on the 1999-2021 dataset, as shown by the higher mean regressor scores and the more consistent performance indicated by the lower IQR. This is most likely because the dataset was much larger than the 2016-2018 dataset. It should also be noted that these visualizations show that both datasets show very high mean regressor score’s, demonstrating that on average our random forest models were very good at predicting crime rates based on our datasets. On average the 2016-2018 model predicted crime rates with 20% error, while 1999-2021 model predicted crime rates with 10% error. Regardless, both of their maximum regressor scores were around the same with 0.97 and 0.96 for the 2016-2018 and 1999-2021 datasets respectively. These are the iterations of the random forest model we will use for our analysis.

Visualization methods:

**Importance scores per attribute**

Scoring the relative importance of predictors is a vital tool to identify which factors of a Random Forest regression model most greatly affect the model's performance. By assigning importance scores to each predictor, the most relevant predictors and those with the least contribution can be identified (Yu et al., 2018).

In the context of our investigation, the respective importance score’s per borough attribute, quantify how much each feature contributes to the random forest model's crime rate predictions. It is based on how much the model's prediction accuracy increases when a feature is utilised vs when the feature is not used. We used Plonski's 'The 3 Ways to Compute Feature Importance in the Random Forest' guide as well as the “sklearn.esemble.RandomForestRegressor” documentation when computing our feature importance’s (Płoński, 2020). It is important to note that importance scores vary depending on the method used to calculate them, such as tree-based methods, permutation-based methods, or variance-based methods (Yu, 2018). However, for our investigation we simply use sckit-learn’s built in feature importance method of calculation, which is ‘Gini importance’ (scikit-learn.org, n.d.). Also, the importance scores of each attribute in the model, sum to 1, which is what makes it easier to compare each attribute against each other (Breiman, 2001).

Here is an example code snippet of how we are computing and visualizing the importance scores:

# Get the feature importances

importances = regressor.feature\_importances\_

# Sort the importance scores in ascending order

indices = np.argsort(importances)

# Get the column names in the correct order

cols = X\_train.columns[indices]

# Create a bar chart using the px.bar() function

fig = px.bar(

x=cols,

y=importances[indices],

labels={'x': 'Feature', 'y': 'Importance'}

)

# Show the plot

fig.show()

To visualize the importance scores of the features in our random forest model, we are using the feature\_importances\_ attribute of the model object to get an array of the scores. We are then creating a bar chart using the plotly.express library, with the feature names on the x-axis and the importance scores on the y-axis. The chart shows the scores in descending order, with the most important features on the right side of the plot. We are using the show method to display the chart, and the labels parameter to specify the labels for the x-axis and y-axis. The chart is interactive, so the in order to hover over the bars to see the exact importance scores of each borough attribute.

Visualisation and analysis:

Random Forest visualisation and analysis

**Why are we using two models?**

We conducted two Random Forest regression’s: on the 2016-2018 dataset, and 1999-2021 dataset. The 1999-2021 dataset had 38 attributes per borough (columns) with 736 rows, whereas the 2016-2018 dataset had 35 attributes per borough with 96 rows. The three attributes that weren’t included in the 2016-2018 data were the 'Proportion of seats won by Conservatives', 'Proportion of seats won by Labour' and 'Proportion of seats won by Liberal Democrats' datasets as there was only data for 2014. As previously stated, the 2016-2018 dataset was formed because it was the only time span that had almost all the borough attributes, meaning we didn’t need to impute for any variables. This gives us more accurate results as its based on real data. For instance, in the 1999-2021 dataset we imputed 24% of the values for the mean. Most of these were the proportion of seats won datasets. As 24% is a relatively significant percentage of the data, it made sense to also run a random forest on the years where we had accurate data. However, the cost of running the model on 2016-2018 data is at the size of the dataset. Hence, we still wanted to run the model on data from 1999-2021 as random forests generally do better on larger datasets.

**Visualisation of 1999-2021 model**

Graph here

The bar chart above shows the relative importance of different borough attributes in predicting crime rates in London boroughs for the 1999-2021 period. The model, which had a regressor score of 0.96, was able to predict crime rates with an error rate of around 4%. This suggests that the model was effective at using the available data to predict crime rates in the different boroughs. The bar chart demonstrates that the top 5 Borough attributes with their corresponding importance scores were:

1. % of Health workers: 0.31

2. Dwellings per hectare: 0.21

3. Life expectancy of men at 65: 0.12

4. % of Education workers: 0.04

5. % of Accommodation and food service activities workers: 0.04

These results mean that the % of Health works per borough, is the most important attribute when predicting a boroughs crime rate. This is followed by Dwellings per hectare, and life expectancy of men at 65. The attributes at 4th and 5th place have a much lower importance score than the first three attributes, hence even though they are the fourth and fifth most important attributes, they aren’t very relevant in predicting crime rates. All the other attributes in the model have near to no importance when predicting crime rates.

**Visualisation of 2016-2018 model**

Graph here

The bar chart above illustrates the relative importance of different attributes in predicting crime rates in London boroughs for the 2016-2018 period. The model used had a regressor score of 0.96, meaning the model predicted crime rates with an error rate of 4%. This suggests that the model was effective at using the available data to predict crime rates in the different boroughs. According to the chart, the top 5 most impactful borough attributes were:

1. Population per hectare: 0.29
2. Consumer expenditure on Pubs and Wine bars (£mn): 0.14
3. % of Health workers: 0.12
4. % of Public Admin and defence workers: 0.08
5. Dwellings per hectare: 0.07

These results indicate that population per hectare was the most important attribute when it comes to predicting crime rates in London boroughs. This was followed by consumer expenditure on pubs and wine bars, % of Health workers, % of public administration and defense workers and dwellings per hectare. Population per hectare has a far higher importance score of 0.29, than the following four attributes. However, the following four attributes have relatively similar scores, demonstrating how they are relatively equally important for the model’s performance. The remaining attributes in the model have relatively low importance scores, indicating that they have little effect on crime rates.

**Comparing the models**

As we can see, we our models produce similar but different results. Both of our models have identified the percentage of health workers and dwellings per hectare as important attributes but have also identified different attributes as the remaining top five most important factors. The difference in the results of the two models is simply due to the different time spans. While certain borough attributes may have had a significant impact on predicting crime from 2016 to 2018, it is possible that the relative importance of these attributes may have changed over the longer time period from 1999 to 2021. It should also be noted that the important scores of the attributes in the two data sets cannot be directly compared due to the different numbers of attributes in each data set (38 in the 1999-2021 data set and 35 in the 2016-2018 data set). However, the ranking of the attributes can be compared, and we expected that these rankings would be more similar than they currently are. Despite the different time spans covered by the two data sets, we would expect factors that influence crime to stay the same. On the other hand, crime rates may fluctuate depending on the political climate of a given time period, which could potentially lead to changes in the factors that influence crime rates (Hagan, et al. 20). This could explain the difference in results between the two time periods.

However, we believe that the primary cause of the discrepancy in the results is likely the method used to impute for missing values in the 1999-2021 data set. Firstly, the 1999-2021 data set contained many imputed values, accounting for 24% of the data. The use of a significant number of imputed values may have introduced inaccuracies into the model. Secondly, imputing for missing values using the mean may have not been appropriate, as it decreased the variation in data and may have misrepresented what those values were. These factors may have contributed to the differences in results between the model using the 1999-2021 data set and the model using the 2016-2018 data set.

**Most important borough attributes when predicting crime rates**

In this investigation, we found that the most important borough attributes for predicting crime rates were population per hectare, consumer expenditure on pubs and wine bars, percentage of health workers, percentage of public administration and defence workers, and dwellings per hectare. However, it is important to note that these factors are not necessarily linearly correlated with crime rates, as the use of random forest regression allows for the analysis of non-linear relationships in the data. Rather, it is the combination of these factors that allows for accurate predictions of crime rates.

Three of these factors - population per hectare, consumer expenditure on pubs and wine bars, and dwellings per hectare - were expected to be influential in predicting crime rates based on our literature review. The population per hectare and dwellings per hectare attributes are indicators for population density. As mentioned previously, population density tends to be a key factor that contributes to crimes (Melling, 2022). Boroughs which have more dwellings and people per hectare, can lead to an overcrowded environment, facilitating pickpocketing (OER Services, (n.d.). In addition, areas which have more alcohol consumption can lead to a disordered societal environment, leading to more crime (Aksoy, 2017).

However, we were surprised to find that the percentage of health workers and the percentage of public administration and defense workers were also significant attributes. We were unable to find any literature to support these findings. One potential explanation for this unexpected relationship may be the concept of "Freakonomics," where seemingly unrelated economic and societal indicators are connected (Levitt and Dubner, 2005). Further research could be conducted to explore the complex relationship between these factors and crime rates, as there is currently a lack of literature on this specific topic.

**Limitations**

Though the model has its strengths such as a model regressor score of 0.97, the dataset being based on accurate data, and Random Forest regression generally performing well on the dataset shown by the average 0.8 regressor score after 10,200 iterations, our findings have limitations.

Firstly, we are unable to tell exactly what relationship the attributes have with crime rates, making it difficult to use our findings to solve crime problem in London. This is because random forest is a predictive modelling tool and not a [descriptive](https://builtin.com/data-science/intro-descriptive-statistics) tool (Donges, 2021). However, we are still able to get an idea of what factors are important in causing crime rates which is still helpful to tackle the problem at hand.

Secondly, the model being based over the time period 2016-2018 is a limitation. It can be argued that it makes our findings less applicable to present day London. However, we attempted to use more up to date data, by imputing for missing values, but this led to relatively untrustworthy results. We prioritized accuracy of data, over the size of the dataset, which is why we used the 2016-2018 data.

In addition, our analysis and conclusions are based on the importance scores generated by the model. However, these importance scores do not always necessarily accurately demonstrate the overall impact each feature had on the model (Brieman, et al., 2001). Factors such as data set size, and feature selection can all influence the resulting importance scores (Salzer, 2017). However, we took steps such as choosing relatively large dataset sizes for our analysis to improve the reliability of our importance scores.

The reliability of the importance scores is based on the assumption that all these factors we have used, are the main indicators that represent the population density, level of education, employment, income, social inequality, and wellbeing of a borough. The issue with this assumption is that there are many other indicators we could use to represent these factors. Some examples of additional data that might be interesting to include could be better data on voting patterns, government spending per borough, street light data etc. Despite this, by using 35 attributes, we believe we have captured a good representation of the various factors that could represent each borough.

**Conclusion**

In conclusion, our analysis of crime rates in London using Random Forest regression models found that the most important borough attributes for predicting crime rates were population per hectare, consumer expenditure on pubs and wine bars, percentage of health workers, percentage of public administration and defence workers, and dwellings per hectare. It is important to note that these factors are not necessarily linearly correlated with crime rates, and it is the combination of these factors that allows for accurate predictions of crime rates. Each one of our borough attributes represents a small portion of the holistic way in which socioeconomic factors influence crime rate. Consequently, though there may not be a strong correlation between each individual factor and crime rate, random forest effectively allows us to better understand the interconnected nature of these factors, which is a better model of reality.

EXPLAIN HOW OUR FINDINGS MEET OUR AIMS AND OBJECTIVES, AND TALK ABOUT POTENTIAL POLICY IMPLICATIONS

**Further lines of research**

There are several additional lines of research that could be pursued to build on the findings of our analysis of crime rates in London. Though there is available literature on the relationship between population density statistics, and alcohol consumption statistics with crime rate, there is no available literature on the relationship between the percentage of health workers per borough and the percentage of public administration and defence workers per borough. Hence, it would be interesting to explore this underlying relationship further to better understand the role that these factors may play in influencing crime rates.

Another area of research could be to increase the number of attributes used in the random forest model. While our analysis included 35 attributes, it is possible that other factors could be relevant to predicting crime rates in London that was not included in our model. By adding additional attributes to the model, it may be possible to find that other attributes have greater importance than the ones we found.

It could also be beneficial to increase the number of years included in the analysis. By increasing the time period covered by the analysis, it may be possible to gain a more comprehensive understanding of the factors that influence crime rates in London over a longer time, making the findings more reliable.