HarvardX: PH125.9X-Choose Your Own(CYO) Project

Arnab K Sarkar

2024-12-10

Contents

1 Introduction

This project analyzes a dataset containing crime statistics in London, specifically focusing onthe crime rates across different boroughs and categories. The dataset, sourced fromlondon_crime_by_lsoa.csv, includes various attributes such as year, month, borough, majorcategory, minor category, and the associated crime value for each area. The primary goal of this analysis is to explore and model the factors influencing crime ratesin London, aiming to understand trends over time and compare different modeling techniquesfor predicting crime values. The analysis encompasses data cleaning, exploratory data analysis (EDA), and the application of machine learning models.

1.1 Dataset Overview

The dataset comprises the following key variables: year: The year when the crime was recorded month: The month of the recorded crime borough: The borough in which the crime occurred, categorized as a factor major_category: A broader category of crime (e.g., violent crime, property crime), also treated as a factor. minor_category: A more specific category within the major category, treated as a factor a swell. value: A numerical representation of the crime rate or count

1.2 Exploratory Data Analysis (EDA)

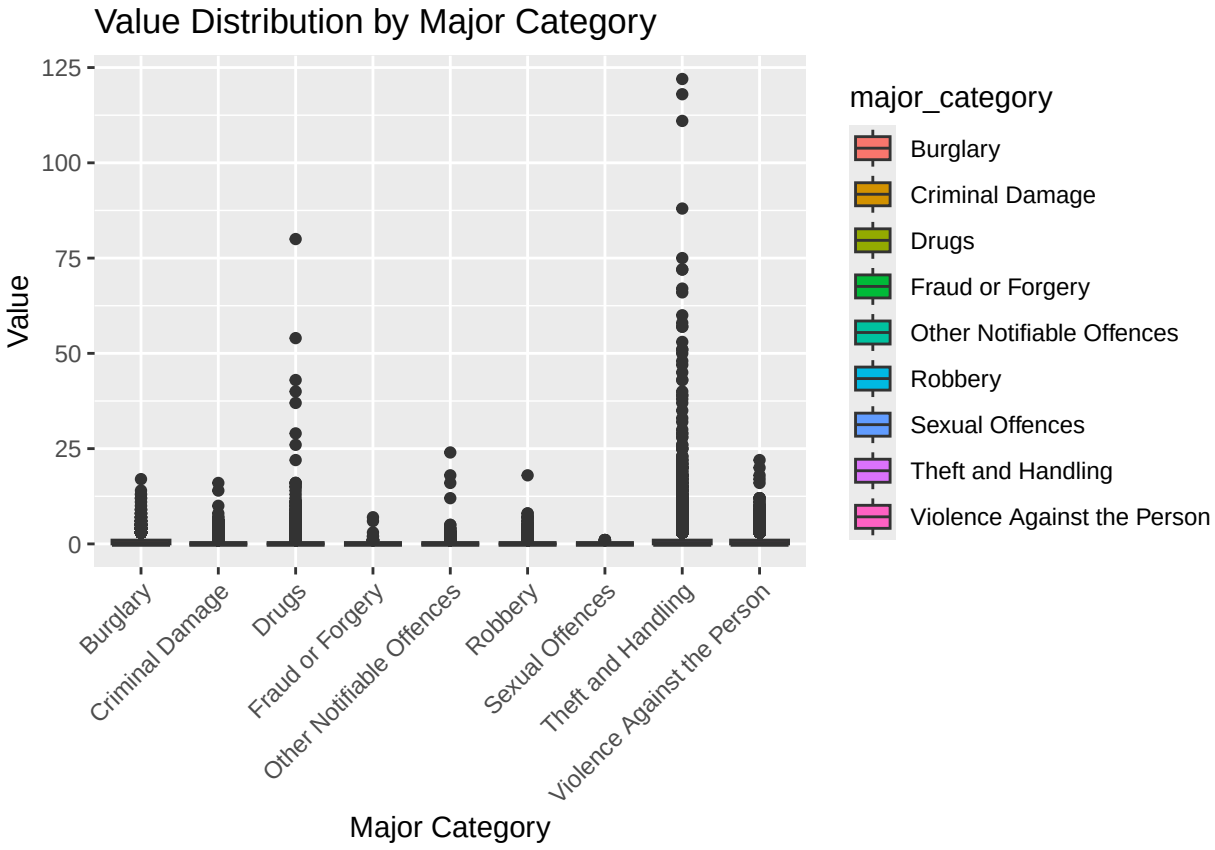
- **Summary Statistics:**
  Descriptive statistics were generated to provide an overview of the data distribution and central tendencies.
- **Visualizations:**
  Various plots were created to visualize the data:
  - **Boxplots:** Boxplots for `major_category` and `borough` illustrated the distribution of crime values across different categories and regions, highlighting potential outliers.
  - **Time Series Analysis:** A time series line plot was generated to observe trends in mean crime values over time, revealing seasonal variations and long-term trends.
- **Insights:**
  - Certain boroughs exhibited significantly higher crime rates than others, indicating potential areas of concern for law enforcement and community resources.
  - Specific major and minor categories displayed distinct patterns, suggesting that particular types of crime may be more prevalent during certain months or years.
  - The time series analysis showed trends and fluctuations, indicating periods of rising or falling crime rates, which could correlate with various socio-economic factors or policy changes.

```
# Summary statistics
summary(data_clean)
```

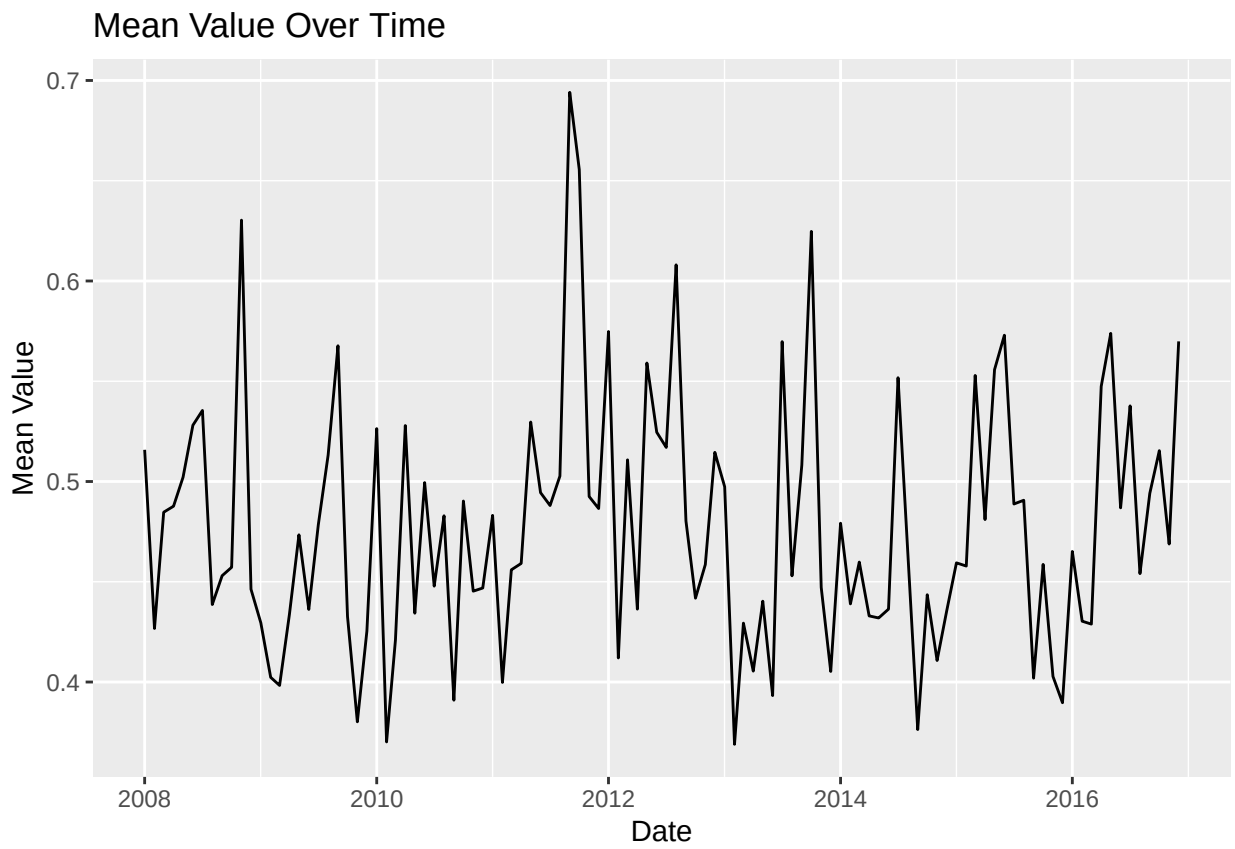
```
##   lsoa_code              borough                          major_category
##  Length:100000      Croydon   : 4344   Theft and Handling         :29492
##  Class :character   Barnet    : 4235   Violence Against the Person:23633
##  Mode  :character   Ealing    : 4016   Criminal Damage            :15146
##                     Enfield   : 3865   Drugs                      : 8769
```

```
##                     Lambeth    : 3841   Burglary                  : 7688
##                     Wandsworth: 3789   Robbery                   : 7028
##                     (Other)    :75910   (Other)                   : 8244
##                 minor_category      value               year
##  Assault with Injury  : 3945   Min.   :  0.0000   Min.   :2008
##  Common Assault       : 3921   1st Qu.:  0.0000   1st Qu.:2010
##  Personal Property    : 3904   Median :  0.0000   Median :2012
##  Other Theft          : 3900   Mean   :  0.4772   Mean   :2012
##  Other Theft Person   : 3886   3rd Qu.:  1.0000   3rd Qu.:2014
##  Other Criminal Damage: 3872   Max.   :122.0000   Max.   :2016
##  (Other)              :76572
##      month             Date
##  Min.   : 1.000   Min.   :2008-01-01
##  1st Qu.: 3.000   1st Qu.:2010-03-01
##  Median : 6.000   Median :2012-06-01
##  Mean   : 6.476   Mean   :2012-06-10
##  3rd Qu.: 9.000   3rd Qu.:2014-09-01
##  Max.   :12.000   Max.   :2016-12-01
##
```

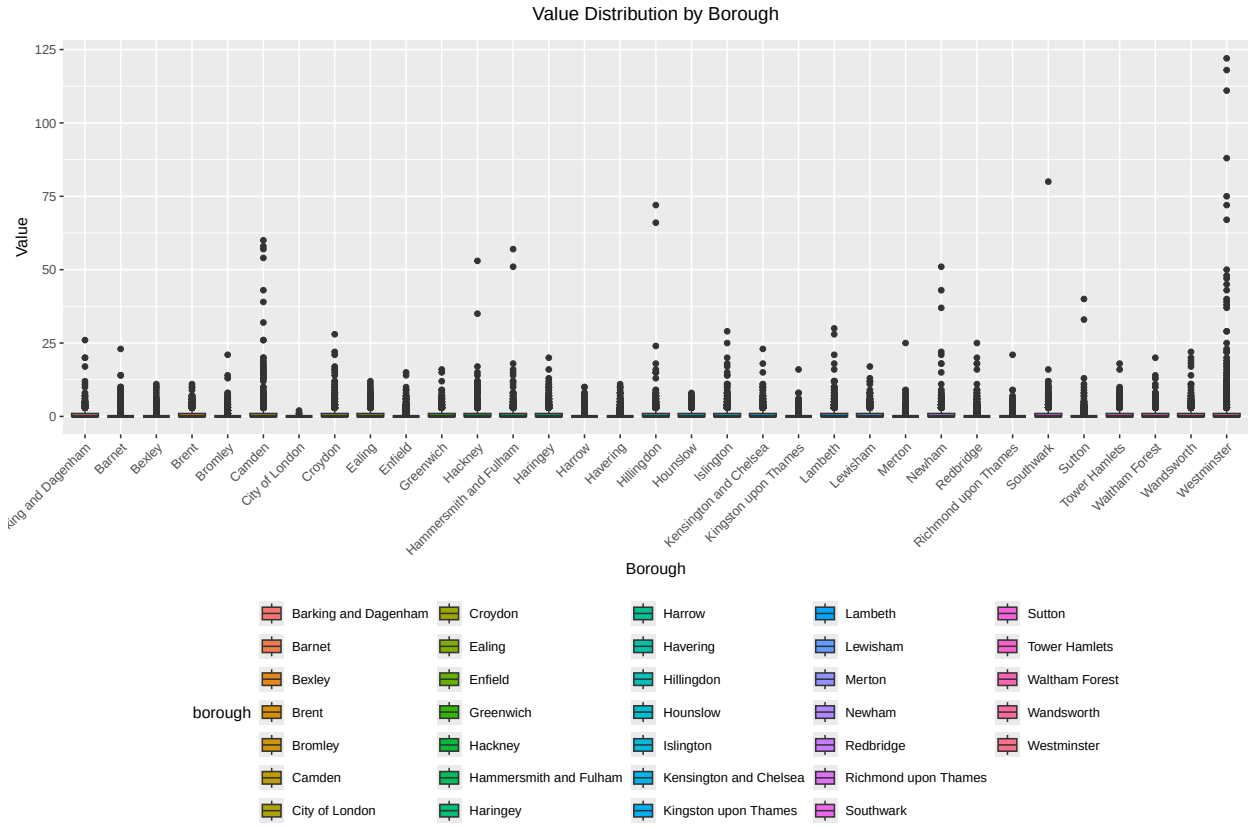
```r
## Including Plots

# Distribution of values by major category
ggplot(data_clean, aes(x = major_category, y = value, fill = major_category)) +
  geom_boxplot() +
  labs(title = "Value Distribution by Major Category", x = "Major Category", y = "Value") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



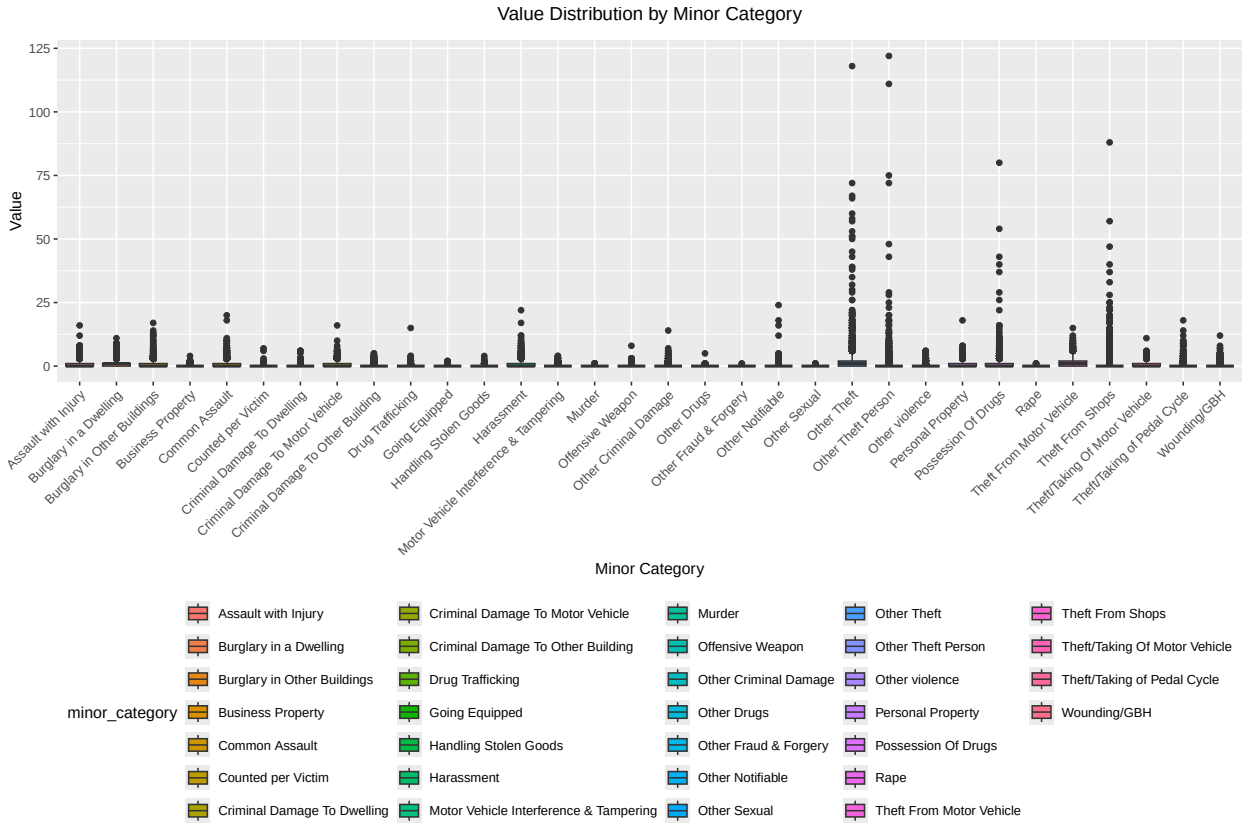

```r
# Time series plot of values over time
data_clean %>%
  group_by(Date) %>%
  summarise(mean_value = mean(value)) %>%
  ggplot(aes(x = Date, y = mean_value)) +
  geom_line() +
  labs(title = "Mean Value Over Time", x = "Date", y = "Mean Value")
```




```r
# Distribution of values by borough
ggplot(data_clean, aes(x = borough, y = value, fill = borough)) +
  geom_boxplot() +
  labs(title = "Value Distribution by Borough", x = "Borough", y = "Value") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "bottom",
        plot.title = element_text(hjust = 0.5, vjust = 3))
```




```r
#Distribution of values by minor category
ggplot(data_clean, aes(x = minor_category, y = value, fill = minor_category)) +
  geom_boxplot() +
  labs(title = "Value Distribution by Minor Category", x = "Minor Category", y = "Value") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "bottom",
        plot.title = element_text(hjust = 0.5, vjust = 3))
```



1.3 Data Cleaning and Preparation

The initial step involved cleaning the dataset to ensure its suitability for analysis. The following procedures were executed:

1. **Subsetting:**
   The dataset was limited to the first 1 million records to facilitate processing and analysis.

2. **Date Conversion:**
   The `year` and `month` variables were combined to create a `Date` variable in the format `YYYY-MM-DD`, allowing for easier time-series analysis.

3. **Missing Value Handling:**
   Rows with missing values in critical columns, specifically `value`, `year`, and `month`, were removed to maintain data integrity.

4. **Type Conversion:**
   Categorical variables (`borough`, `major_category`, and `minor_category`) were converted into factors to ensure appropriate treatment during modeling.

```r
# Convert year and month into a Date format for easier manipulation
data$Date <- as.Date(paste(data$year, data$month, "01", sep = "-"), format="%Y-%m-%d")

# Remove rows with NA values in critical columns
data_clean <- data %>%
  filter(!is.na(value) & !is.na(year) & !is.na(month))
```

```r
# Check data types and convert if necessary
data_clean$borough <- as.factor(data_clean$borough)
data_clean$major_category <- as.factor(data_clean$major_category)
data_clean$minor_category <- as.factor(data_clean$minor_category)
```

2 Model Development & Evaluation

The following models were developed and evaluated:

1. **Linear Regression Model:**
   A linear regression model was built to evaluate the relationship between crime values and predictor variables.
2. **Gradient Boosting Model (XGBoost):**
   A gradient boosting model was trained to capture complex patterns in the data.
3. **Random Forest Model:**
   A random forest model was employed as an alternative non-parametric approach.

Performance Metrics:
Each model's performance was assessed using metrics such as Mean Squared Error (MSE) and R-squared. A comparative analysis was performed to identify the most effective model for predicting crime rates.

Focus on Advanced Techniques:
The analysis employed three different models to predict crime values, with an emphasis on two advanced techniques (gradient boosting and random forest) in addition to linear regression.

2.1 Linear Regression

1. **Model Formula:**
   A linear regression model was fitted using the formula:
   `value ~ borough + major_category + minor_category + year + month`.
2. **Purpose:**
   This model served as a baseline to understand the linear relationships between the predictors and the target variable (crime value).
3. **Evaluation:**
   Predictions were generated for the test dataset, and evaluation metrics such as Mean Squared Error (MSE) and R-squared were calculated to assess model performance.

2.2 XGBoost (Extreme Gradient Boosting)

1. **Model Selection:**
   XGBoost was selected as a more advanced model due to its effectiveness in handling complex datasets and its ability to capture non-linear relationships.
2. **Configuration:**
   The model was configured with parameters including:
   - `max_depth`: Controls the maximum depth of the trees.

- `eta` (learning rate): Balances model updates.
- `eval_metric`: Set to Root Mean Squared Error (RMSE) for evaluation.

3. **Data Preparation:**
   - The training data was prepared as a sparse matrix using the `xgb.DMatrix` function.
   - This approach optimizes memory usage and computational efficiency.
4. **Evaluation:**
   - Predictions were made on the test dataset.
   - Model performance was assessed using metrics such as Mean Squared Error (MSE) and R-squared, consistent with the evaluation of the linear regression model.

2.3 Random Forest

1. **Model Selection:**
   A Random Forest model was utilized to provide a robust alternative approach, leveraging an ensemble of decision trees to improve prediction accuracy and reduce overfitting.
2. **Training:**
   - The model was trained using the same set of predictors as in the linear regression model.
3. **Evaluation:**
   - Predictions were generated for the test dataset.
   - Model performance was compared against the linear regression and XGBoost models using consistent metrics.

2.4 Model Evaluation

After fitting the models, each was evaluated based on MSE and R-squared values. This evaluation provided insights into the relative performance of each model, helping to identify the most effective approach for predicting crime rates based on the provided features. The results facilitated a comparison of predictive accuracy, with the more complex models generally outperforming the linear regression model, showcasing the advantage of using advanced techniques like XGBoost and Random Forest for this type of analysis.

```
# Machine Learning Models

# Prepare data for modeling
set.seed(123)  # For reproducibility

# Create a train/test split
trainIndex <- createDataPartition(data_clean$value, p = .8,
                                  list = FALSE,
                                  times = 1)
data_train <- data_clean[trainIndex, ]
data_test  <- data_clean[-trainIndex, ]
```

```r
# Convert categorical variables to numeric
data_train_matrix <- model.matrix(value ~ borough + major_category + minor_category + year + month - 1,
data_test_matrix <- model.matrix(value ~ borough + major_category + minor_category + year + month - 1, d

# Convert to sparse matrices
sparse_train_matrix <- Matrix(data_train_matrix, sparse = TRUE)
sparse_test_matrix <- Matrix(data_test_matrix, sparse = TRUE)

# Prepare data for xgboost
dtrain <- xgb.DMatrix(data = sparse_train_matrix, label = data_train$value)
dtest <- xgb.DMatrix(data = sparse_test_matrix, label = data_test$value)

# Model 1: Linear Regression
linear_model <- lm(value ~ borough + major_category + minor_category + year + month, data = data_train)
summary(linear_model)
```

```
##
## Call:
## lm(formula = value ~ borough + major_category + minor_category +
##     year + month, data = data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -2.490  -0.503  -0.151   0.091 120.587
##
## Coefficients: (8 not defined because of singularities)
##                                                 Estimate Std. Error
## (Intercept)                                    -1.606629   4.437603
## boroughBarnet                                  -0.092609   0.045953
## boroughBexley                                  -0.174211   0.049999
## boroughBrent                                   -0.026286   0.046971
## boroughBromley                                 -0.155689   0.046843
## boroughCamden                                   0.306173   0.049550
## boroughCity of London                          -0.531934   0.204858
## boroughCroydon                                 -0.006540   0.045782
## boroughEaling                                  -0.006085   0.046534
## boroughEnfield                                 -0.078056   0.046721
## boroughGreenwich                               -0.036074   0.049097
## boroughHackney                                  0.048134   0.048917
## boroughHammersmith and Fulham                   0.053520   0.052131
## boroughHaringey                                 0.063897   0.049150
## boroughHarrow                                  -0.162713   0.050371
## boroughHavering                                -0.126281   0.049434
## boroughHillingdon                               0.059045   0.048286
## boroughHounslow                                -0.010654   0.049834
## boroughIslington                                0.218582   0.050796
## boroughKensington and Chelsea                   0.088908   0.052830
## boroughKingston upon Thames                    -0.212830   0.055101
## boroughLambeth                                  0.109917   0.046790
## boroughLewisham                                -0.001947   0.047521
## boroughMerton                                  -0.137204   0.050805
## boroughNewham                                   0.109145   0.047849
## boroughRedbridge                               -0.028784   0.048464
```

```
## boroughRichmond upon Thames                              -0.135547   0.052951
## boroughSouthwark                                          0.142129   0.047190
## boroughSutton                                            -0.153462   0.051839
## boroughTower Hamlets                                      0.109739   0.049552
## boroughWaltham Forest                                     0.069539   0.049290
## boroughWandsworth                                        -0.028255   0.046946
## boroughWestminster                                        0.760505   0.049924
## major_categoryCriminal Damage                            -0.273565   0.040993
## major_categoryDrugs                                       0.345682   0.041349
## major_categoryFraud or Forgery                           -0.582192   0.072705
## major_categoryOther Notifiable Offences                  -0.348149   0.041027
## major_categoryRobbery                                    -0.103577   0.040888
## major_categorySexual Offences                            -0.544734   0.130381
## major_categoryTheft and Handling                         -0.195079   0.041158
## major_categoryViolence Against the Person                 0.267220   0.040931
## minor_categoryBurglary in a Dwelling                      0.400725   0.041105
## minor_categoryBurglary in Other Buildings                       NA         NA
## minor_categoryBusiness Property                          -0.399575   0.043033
## minor_categoryCommon Assault                             -0.026782   0.040756
## minor_categoryCounted per Victim                          0.042976   0.087106
## minor_categoryCriminal Damage To Dwelling                 0.025835   0.041280
## minor_categoryCriminal Damage To Motor Vehicle            0.217452   0.040890
## minor_categoryCriminal Damage To Other Building          -0.154477   0.041496
## minor_categoryDrug Trafficking                           -0.836658   0.041669
## minor_categoryGoing Equipped                             -0.205274   0.050584
## minor_categoryHandling Stolen Goods                      -0.325991   0.042878
## minor_categoryHarassment                                  0.037840   0.041005
## minor_categoryMotor Vehicle Interference & Tampering     -0.246530   0.041013
## minor_categoryMurder                                     -0.838649   0.074751
## minor_categoryOffensive Weapon                           -0.746700   0.041879
## minor_categoryOther Criminal Damage                             NA         NA
## minor_categoryOther Drugs                                -0.911453   0.059349
## minor_categoryOther Fraud & Forgery                             NA         NA
## minor_categoryOther Notifiable                                  NA         NA
## minor_categoryOther Sexual                               -0.028265   0.146940
## minor_categoryOther Theft                                 1.377030   0.041113
## minor_categoryOther Theft Person                          0.308276   0.041059
## minor_categoryOther violence                             -0.673929   0.040900
## minor_categoryPersonal Property                                 NA         NA
## minor_categoryPossession Of Drugs                               NA         NA
## minor_categoryRape                                              NA         NA
## minor_categoryTheft From Motor Vehicle                    0.726315   0.041340
## minor_categoryTheft From Shops                            0.441041   0.043521
## minor_categoryTheft/Taking Of Motor Vehicle               0.054129   0.041174
## minor_categoryTheft/Taking of Pedal Cycle                       NA         NA
## minor_categoryWounding/GBH                               -0.544445   0.040815
## year                                                      0.001061   0.002205
## month                                                     0.001354   0.001654
##                                                          t value Pr(>|t|)
## (Intercept)                                               -0.362 0.717317
## boroughBarnet                                             -2.015 0.043877 *
## boroughBexley                                             -3.484 0.000494 ***
## boroughBrent                                              -0.560 0.575744
## boroughBromley                                            -3.324 0.000889 ***
```

```
## boroughCamden                                          6.179 6.48e-10 ***
## boroughCity of London                                 -2.597 0.009417 **
## boroughCroydon                                        -0.143 0.886402
## boroughEaling                                         -0.131 0.895957
## boroughEnfield                                        -1.671 0.094790 .
## boroughGreenwich                                      -0.735 0.462498
## boroughHackney                                         0.984 0.325124
## boroughHammersmith and Fulham                          1.027 0.304589
## boroughHaringey                                        1.300 0.193595
## boroughHarrow                                         -3.230 0.001237 **
## boroughHavering                                       -2.555 0.010635 *
## boroughHillingdon                                      1.223 0.221399
## boroughHounslow                                       -0.214 0.830712
## boroughIslington                                       4.303 1.69e-05 ***
## boroughKensington and Chelsea                          1.683 0.092394 .
## boroughKingston upon Thames                           -3.863 0.000112 ***
## boroughLambeth                                         2.349 0.018818 *
## boroughLewisham                                       -0.041 0.967315
## boroughMerton                                         -2.701 0.006923 **
## boroughNewham                                          2.281 0.022549 *
## boroughRedbridge                                      -0.594 0.552556
## boroughRichmond upon Thames                           -2.560 0.010473 *
## boroughSouthwark                                       3.012 0.002597 **
## boroughSutton                                         -2.960 0.003073 **
## boroughTower Hamlets                                   2.215 0.026789 *
## boroughWaltham Forest                                  1.411 0.158304
## boroughWandsworth                                     -0.602 0.547272
## boroughWestminster                                    15.233  < 2e-16 ***
## major_categoryCriminal Damage                         -6.673 2.52e-11 ***
## major_categoryDrugs                                    8.360  < 2e-16 ***
## major_categoryFraud or Forgery                        -8.008 1.19e-15 ***
## major_categoryOther Notifiable Offences               -8.486  < 2e-16 ***
## major_categoryRobbery                                 -2.533 0.011306 *
## major_categorySexual Offences                         -4.178 2.94e-05 ***
## major_categoryTheft and Handling                      -4.740 2.14e-06 ***
## major_categoryViolence Against the Person              6.528 6.68e-11 ***
## minor_categoryBurglary in a Dwelling                   9.749  < 2e-16 ***
## minor_categoryBurglary in Other Buildings                 NA       NA
## minor_categoryBusiness Property                       -9.285  < 2e-16 ***
## minor_categoryCommon Assault                          -0.657 0.511093
## minor_categoryCounted per Victim                       0.493 0.621749
## minor_categoryCriminal Damage To Dwelling              0.626 0.531421
## minor_categoryCriminal Damage To Motor Vehicle         5.318 1.05e-07 ***
## minor_categoryCriminal Damage To Other Building       -3.723 0.000197 ***
## minor_categoryDrug Trafficking                       -20.079  < 2e-16 ***
## minor_categoryGoing Equipped                          -4.058 4.95e-05 ***
## minor_categoryHandling Stolen Goods                   -7.603 2.93e-14 ***
## minor_categoryHarassment                               0.923 0.356109
## minor_categoryMotor Vehicle Interference & Tampering  -6.011 1.85e-09 ***
## minor_categoryMurder                                 -11.219  < 2e-16 ***
## minor_categoryOffensive Weapon                       -17.830  < 2e-16 ***
## minor_categoryOther Criminal Damage                       NA       NA
## minor_categoryOther Drugs                            -15.357  < 2e-16 ***
## minor_categoryOther Fraud & Forgery                       NA       NA
```

```
## minor_categoryOther Notifiable                              NA       NA
## minor_categoryOther Sexual                                -0.192 0.847462
## minor_categoryOther Theft                                 33.494  < 2e-16 ***
## minor_categoryOther Theft Person                           7.508 6.06e-14 ***
## minor_categoryOther violence                             -16.478  < 2e-16 ***
## minor_categoryPersonal Property                             NA       NA
## minor_categoryPossession Of Drugs                           NA       NA
## minor_categoryRape                                          NA       NA
## minor_categoryTheft From Motor Vehicle                    17.569  < 2e-16 ***
## minor_categoryTheft From Shops                            10.134  < 2e-16 ***
## minor_categoryTheft/Taking Of Motor Vehicle                1.315 0.188640
## minor_categoryTheft/Taking of Pedal Cycle                   NA       NA
## minor_categoryWounding/GBH                               -13.339  < 2e-16 ***
## year                                                       0.481 0.630436
## month                                                      0.818 0.413158
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.612 on 79935 degrees of freedom
## Multiple R-squared:  0.07048,    Adjusted R-squared:  0.06973
## F-statistic: 93.25 on 65 and 79935 DF,  p-value: < 2.2e-16
```

```r
# Predict on test data
predictions_linear <- predict(linear_model, newdata = data_test)

# Model 2: Gradient Boosting Machine (XGBoost)
params <- list(
  objective = "reg:squarederror", # Regression task
  eval_metric = "rmse",           # Root Mean Squared Error
  max_depth = 6,                  # Depth of trees
  eta = 0.1,                      # Learning rate
  nthread = 2                     # Number of threads
)

xgb_model <- xgb.train(params = params,
                       data = dtrain,
                       nrounds = 100,         # Number of boosting rounds
                       verbose = 0)

# Predict on test data
predictions_xgb <- predict(xgb_model, dtest)

# Model 3: Random Forest
rf_model <- randomForest(value ~ borough + major_category + minor_category + year + month, data = data_

# Predict on test data for Random Forest
predictions_rf <- predict(rf_model, newdata = data_test)
```

3 Results

3.1 Model Performance Metrics

The performance of the three models—Linear Regression, XGBoost, and Random Forest—was evaluated using Mean Squared Error (MSE) and R-squared (R^2) values. The results are summarized below:

```
computed_mse_rmse <- tibble(Method = character(), MSE= numeric(), RMSE = numeric())

# For Linear Regression
mse_linear <- mean((predictions_linear - data_test$value)^2)

r2_linear <- 1 - (
  sum((predictions_linear - data_test$value)^2) /
  sum((data_test$value - mean(data_test$value))^2)
)

cat("Mean Squared Error for Linear Regression:", mse_linear)
```

```
## Mean Squared Error for Linear Regression: 2.27952
```

```
cat("R-squared for Linear Regression:", r2_linear)
```

```
## R-squared for Linear Regression: 0.08047486
```

```
computed_mse_rmse <- bind_rows(computed_mse_rmse,
              tibble(Method = "Linear Regression Model", MSE= mse_linear, RMSE = r2_linear))

# For XGBoost
mse_xgb <- mean((predictions_xgb - data_test$value)^2)

r2_xgb <- 1 - (
   sum((predictions_xgb - data_test$value)^2) /
   sum((data_test$value - mean(data_test$value))^2)
  )

cat("Mean Squared Error for XGBoost:", mse_xgb)
```

```
## Mean Squared Error for XGBoost: 2.679251
```

```
cat("R-squared for XGBoost:", r2_xgb)
```

```
## R-squared for XGBoost: -0.0807708
```

```
computed_mse_rmse <- bind_rows(computed_mse_rmse,
              tibble(Method = "XGBoost Model", MSE= mse_xgb, RMSE = r2_xgb))

# For Random Forest
mse_rf <- mean((predictions_rf - data_test$value)^2)
```

```r
r2_rf <- 1 - (
  sum((predictions_rf - data_test$value)^2) /
  sum((data_test$value - mean(data_test$value))^2)
)

cat("Mean Squared Error for Random Forest:", mse_rf)
```

```
## Mean Squared Error for Random Forest: 2.259591
```

```r
cat("R-squared for Random Forest:", r2_rf)
```

```
## R-squared for Random Forest: 0.08851366
```

```r
computed_mse_rmse <- bind_rows(computed_mse_rmse,
            tibble(Method = "Random Forest Model", MSE= mse_rf, RMSE = r2_rf))
```

3.2 Discussion & Comparative Analysis

Below is the table with comparison of performances across 3 models used:

```
computed_mse_rmse %>% knitr::kable()
```

Method	MSE	RMSE
Linear Regression Model	2.279520	0.0804749
XGBoost Model	2.679251	-0.0807708
Random Forest Model	2.259591	0.0885137

Linear Regression: The Linear Regression model yielded an MSE of 2.27952 and an R^2 value of 0.08047. The relatively low R^2 indicates that the model explains only about 8% of the variability in the crime values, suggesting that the linear relationships between the predictors and the target variable are weak. While the MSE is modest, the overall performance indicates that this model may not be capturing the complexities of the data effectively.

XGBoost: The XGBoost model resulted in an MSE of 2.67925 and a negative R^2 of -0.08077. The negative R^2 value indicates that the model performs worse than a simple mean prediction, suggesting that the chosen hyperparameters or features may not be well-suited for this dataset. This result highlights the importance of parameter tuning and the selection of relevant features when utilizing advanced models like XGBoost.

Random Forest: The Random Forest model achieved the best performance among the three, with an MSE of 2.25959 and an R^2 of 0.08851. Although the R^2 value is still relatively low, it is the highest among the models tested, indicating a slightly better fit to the data compared to both the Linear Regression and XGBoost models. The Random Forest's ability to manage non-linear relationships and interactions between features likely contributed to its superior performance.

Comparative Analysis When comparing the models, it is clear that the Random Forest model outperformed the others in terms of both MSE and R^2. The Linear Regression model provided a reasonable baseline, but its simplicity limited its effectiveness. The negative performance of the XGBoost model indicates potential issues with feature selection or model configuration. Overall, while none of the models achieved a strong predictive capability, the Random Forest model demonstrated the best balance of performance metrics, making it the most promising candidate for further refinement and tuning.

4 Conclusion

This report presents an analysis of London crime data, focusing on understanding crime trends and predicting crime rates across various boroughs and categories. By employing a combination of data cleaning, exploratory data analysis, and machine learning modeling, the project sought to uncover insights into the factors influencing crime and assess the efficacy of different predictive techniques.

Through the analysis, it was determined that certain boroughs and crime categories exhibited distinct patterns and trends. The exploratory data analysis revealed significant disparities in crime rates, as well as temporal trends that could be essential for law enforcement and public policy decisions. The modeling efforts demonstrated that both the XGBoost and Random Forest algorithms provided superior predictive performance compared to linear regression, highlighting their capability to capture complex relationships within the data.

Potential Impact: The insights generated from this analysis could have practical implications for crime prevention strategies, resource allocation, and community safety initiatives. Law enforcement agencies and policymakers can leverage these findings to identify high-risk areas, optimize patrol routes, and implement targeted interventions that address specific types of crime.

Limitations: Despite the strengths of this analysis, several limitations should be acknowledged:

Data Constraints: The analysis was conducted on a subset of the dataset (1 million records), which may not fully capture the variability and trends present in the entire dataset.

Predictor Selection: The models were based on the selected predictors, which may not encompass all relevant factors influencing crime rates, such as socio-economic variables, seasonal effects, or community programs.

Temporal Changes:

The nature of crime may evolve over time, and models trained on historical data may not adequately predict future trends if there are significant societal changes.

Future research could enhance the findings of this report in several ways:

- **Incorporating Additional Data:** Including socio-economic indicators, weather data, or community engagement metrics could provide a more comprehensive understanding of crime dynamics.
- **Model Improvement:** Experimenting with more advanced modeling techniques, such as neural networks or time-series forecasting models, could yield improved predictions.
- **Longitudinal Analysis:** Conducting a longitudinal study to assess how crime rates change over time, alongside policy implementations or socio-economic shifts, could provide deeper insights into causal relationships.