University of East London

Pioneering Futures Since 1898

SCHOOL OF ARCHITECTURE, COMPUTING AND ENGINEERING

Department of Engineering and Computing

Crime Analysis in London and it's Borough Level

Adithyan Sunil Kumar

2409798

A report submitted in part fulfilment of the degree of

BSc (Hons) in Data Science and Artificial Intelligence

Supervisor**: Reena Popat**

CN6000

8 May 2025

Abstract

This research uses statistical and geographical data mining tools to examine and forecast London's crime trends. Understanding the root causes of criminal behaviour and identifying high-risk locations are the main goals in order to promote crime prevention initiatives. Since crime is a persistent problem in urban areas, this study uses crime statistics at the borough level along with pertinent socioeconomic factors to create prediction models and geographical visualisations that provide law enforcement and policymakers with useful information.

The technique focusses on several kinds of offences, including theft, violence, burglary, and drug-related crimes, and entails gathering and preparing historical crime data from publicly accessible sources. Spatial distributions are mapped using Geographic Information System (GIS) tools, while trends and future hotspot predictions are made using machine learning methods, namely classification and clustering approaches. To increase model accuracy, key characteristics were chosen and designed, and the effectiveness of methods include K-Mean clustering, decision trees, and support vector machines was evaluated. In order to improve interpretability and facilitate data-driven decision-making, the project also included data visualisation.

The project's conclusion shows how statistical and geographical analysis may be used to find trends in the distribution of crimes and help with predictive modelling of future crime incidents. This result shows distinct geographical differences in crime amongst London boroughs, and there is a strong correlation between high crime rates and specific socioeconomic factors. With more, detailed and up-to-date data, the prediction models; performance might be significantly enhanced, especially when it comes to identifying crime-prone locations. In the end, this study creates a workable framework for combining machine learning with spatial analytics in urban crime investigation, which helps with resource allocation and proactive policing in urban areas like London.

Acknowledgments

My profound appreciation goes out to my supervisor, Reena Popat, for her helpful advice, unwavering support, and perceptive criticism during this project. Her knowledge and support have greatly influenced the course and results of my work.

Additionally, I would like to thank Ragulgowtham Panneer Selvam, whose counsel and help during the project's development have been invaluable. His advice has enabled me to improve my strategy and get beyond a number of technical obstacles.

Lastly, I would want to express my gratitude to everyone who helped to complete this project by directly or indirectly lending their time, expertise, or resources. My accomplishment has been greatly aided by your support.

Contents

Chapter 1: **Introduction**

1.1 Introduction

Crime analysis is pivotal in urban planning and public safety. Given the social and economic diversity of London's 32 boroughs, brough-level crime data offers vital information for focusing interventions. With a focus on statistical and geographic approaches, this study summarises the body of research on crime analysis metrology and draws attention to the particular difficulties in examine crime trends in London.

1.2 Problem Statement and Project Purpose:

Problem statement

The crime rates in London, a major city with an international impact, vary greatly throughout its boroughs. Crime is still a major issue despite several attempts to combat it , with noticeable borough-level differences caused by intricate socioeconomic and demographic factors (Boba, 2001). Even through there is a lot of research on urban crime, there aren't many thorough studies that combine statistical and geospatial methods to find patterns and relationships unique to London boroughs (Meghanathan, 2015). Designing successful crime prevention methods requires an understanding of how variable like population density, economic disparity, and unemployment contribute to these discrepancies. Additionally, to enhance community safety initiatives, policymaking, and resource allocation , localised insight are crucial. Addressing the underlying causes of crime disparities at the borough level is still difficult in the absence of such analyses.

Project purpose

In order to find patterns, trends, and contributing causes, this project will use publically accessible statistics to examine the statistics to examine the statistical and geographical aspects of crime in London's boroughs. The project will identify hotspots and visualise the distribution of crime using geospatial mapping. correlations between crime rates and socioeconomic indicators will next be investigated using statistical analysis. This study has two goals: it will advance scholarly knowledge of the dynamics of urban crime and offer useful information to law enforcement and policymakers.in the end , the results will help create a safer and more just London by guiding plans to improve urban safety and deal will borough-specific issues

1.3 Aim and Objectives

The aim of the project is to analyse crime trends, patterns, and variations across the boroughs of London using publicly available datasets, employing statistical and geospatial methodologies to identify key factors influencing crime rates and their implications for urban safety strategies.

1. To research extant literature, with an emphasis on London and its boroughs, on crime analysis, statistical and spatial techniques, and crime pattern in urban location.
2. To research crime and socioeconomic statistics for London boroughs that are accessible to public from the sources such as the UK Government, the office for National statistics (ONS) and the Metropolitan police.
3. To conduct geospatial analysis to map the distribution of crimes and pinpoint hotspots in the various London boroughs.
4. To analyse using a combination of spatial (GIS) and quantitative (statistical) methodologies, crime data is analysed to find patterns, correlations and contributing variable for each borough's crime rate.
5. To implement data visualisation techniques (e.g. Chart, graph, map) to represent the findings clearly and effectively.
6. To evaluation the outcome by analysing the effects to demographic and socioeconomic variable on the differences in the crime among boroughs and contrasting the findings the previous research.
7. To reflect on the result and findings by discussing their implications for crime prevention, policy-making, and urban safety strategies in London .

Chapter 2: **Literature Review**

2.1 Definition of crime analysis

In the study of Rachel Boba, he explained that in order to catch criminals, stop crime ,lessen disturbance, and assess organisational processes, crime and law enforcement data are studied both qualitatively and quantitatively , along with sociodemographic and geographic aspects. .(Rachel Boba Ph.D. 2001.p -9). Any intentional act that violates the law, results in property damage or psychological suffering is considered a crime. A seriocomic issue, crime has an impact on both economic progress and the quality of life.

2.1.1 Crime Analysis type

The following are five type of analysis that fall under the umbrella of crime analysis. As you will see, each is specific in the type of data and analysis used as well as in its purpose.

Intelligence Analysis

According to Rachel Boba (2001), It has been used in intelligence analysis to aid sworn personnel in identifying of networks and the understanding of individuals to subsequently deter criminal activities. A related objective is to connect information, give priority information, find relation, establish relationships, and locate the domain for further investigation by setting. This allows for analysis within an understandable framework. Most of the materials assessed in the field of intelligence analysis, such officers are not called to the police by any citizen but are collated by the law enforcement agencies. Certain data collated includes follow-ups, secret informers and participant observation. Besides the category of information encompasses more than merely criminal data; it may also comprise telephone details, conversation, travel information, financial/tax information, family and business issues intelligence analysis has historically placed a greater emphasis on organised crime includes syndicates for drugs and prostitution. The Qualitative data exist in abundance since most data are usually analysed qualitatively. Utilising qualitative methodologies.

Criminal Investigative Analysis

In the findings of Rachel Bobac (2001), this type of analysis is also referred to as "profiling," whereby the process involves building an "offender profile "with information derived from the characteristics of the offense and the facts of the offense and the facts of the case, and the characteristics of the victim. Similar to intelligence analysis, the form of analysis focuses very much on the

qualitative data of serious serial. Crimes like murder and rape are included under this category A data are collected and analysed individually. Norm for the persons who are directly or indirectly concerned with what happens. The spatial attributes of the occurrences and related sites, such as the scenes of disposal, encounter sites is also put into consideration. The primary object of criminal investigative it will aid analysts in developing patterns of serial crime crossing city, state, and even national borders. Set boundaries based on relating activities and justification to each other across multiple instances of Apprehend suspect and/or close cases.

Tactical Crime Analysis

In the finds of Rachel Bobac(2001) Tactical crime analysis depends upon crime data newly available and reported to the police, focusing on specific information such as modus operandi, victim type, and location. Information from the field is considered, which includes data from patrol officers regarding suspicious activities and identifying marks, like scars or tattoos, on people. Most of all, it attempts to search for patterns, series, and linkages with a view towards suspect identification and case closure. Utilised are qualitative methods, including critical thinking and content analysis, but also quantitative, although qualitative methods are the most prevalent in determining initial patterns. This type of analysis occurs on a daily basis, and when patterns or leads are developed, the subsequent findings are passed on to the patrol officers and detectives to help further their active investigations.

Strategic Crime Analysis

Strategic Crime analysis focuses on understanding the persistent crime problems by analysing aggregated data; examples include crime statistics, service calls, and traffic events.it doesn't focus on individual cases but, instead, looks at overall trends and patterns based on a set of variables that include demographics and geographic locations.it aims at identifying ongoing problems-a drug problem or car theft-sometimes targeting an assessment or refinement of strategies utilised by police agencies. That includes program evaluation, studying of specific problems in details, and garnering of public comments about crime and policing. It may also cover officer deployment analysis, beat assignment, and review of data accuracy. In other words, data and research methodologies are used to deal with the persistent problems and enhancement of police operations. Those analysts are often called research or problem analysts.

Administrative Crime Analysis

Administrative crime analysis is a summary process and does not include statistical analysis or research. It identifies what information will be

communicated and how that information will be communicated and how that information will be packaged, depending on who will receiving the information and for what purpose. It is often a higher-level overview of more detailed analysis, a type of executive report. The information will be legally, politically and pragmatically framed with concerns for privacy and complexity. This is for enlightening various group of audiences like police executives, city councils, media, citizens, and community groups.

A common example is the publishing of police website showcasing crime information to be passed to the public. Since it targets a wide audience, the information should be clear and simple for it to be understood by all, without revealing sensitive information. A good guide will be publishing material that could comfortably appear on the evening news.

2.2 Data Mining and Machine Learning algorithms

2.2.1 Data mining

Data mining as defined by Natarajan Meghanathan (2015), is the process by which raw data gets analysed and inferred into information.it helps make proper predictions and applies the same to real-life situation, such as market trends or consumer behavior. Data mining tools like WEKA software uses machine learning algorithms to do the following important tasks:

1. Association: To find relationships between different values that may exist in a dataset.
2. Classification: Data is grouped into predefined classes, predicting an outcome using a model, such as a decision tree or neural network.
3. Clustering: This finds unknown patterns by bringing the similar data together. It is generally done using neural networks.
4. Forecasting: Extrapolating, from logically related pattern found in the data, into the future.
5. Visualisation: The representation of data is clear, graphical formats to enable user to quickly identify patterns and relationships.

These methods jointly enable detection, pattern generation, prediction, and representation of complicated data in very interpretable format; thus, turning data mining into a powerful process for any kind of analysis and decision-making.

2.2.2 Machine learning

According to Arthur Natarajan Meghanathan (2015), machine learning is the process through which computers improve their performance through experience gained from the data. it depends on learning by example and, essentially, can focus on two bases:

1. Supervised Learning: Examples include learning from labeled data to predict a particular outcome or attribute.
2. Unsupervised Learning: the algorithms deal with untagged data to find out unseen patterns and structures.

Basically, machine learning is using five kinds of major algorithms for data. In general, machine learning employs five major kinds of algorithms for data mining:

1. Classification Analysis: predicts discrete values based on attributes of the dataset
2. Regression Analysis: This will predict the continuous values-in other words, profit or loss-resulting from the relationship between variables.
3. Segmentation Analysis: separate data items into groups of similar nature.
4. Association Analysis: to find the association of an attribute with another, common applications include market analysis.
5. Sequence Analysis: identifies patterns or relationship over time, like the flow of navigation through the web. These algorithms help in extracting insights from data, driving prediction, pattern recognition, and decision-making in various applications.

These include advanced visualisation tools that allow the presentation of complex data in an intelligible form. Data mining and machine learning combined show a sound process of identifying patterns, predicting trends, and making decisions upon data from a wide range of real-world applications.

According to Kavala et al., (2022) ,this research really explained how python may be applied in the area of crime analytics, sustained by broad support with its powerful libraries, from machine learning and deep learning with NumPy, Pandas, and Kera's, Google Colab served as the building environment. Among the feature set was a prediction of the output variable for sex, first charge,

location, and weapon. Different algorithm comparison showed the random forest to contain 75.7% and the neural networks 83% for accuracy.

There was an imbalance in the dataset that biased the performance of the model. The usage of directed under-sampling techniques helped in balancing the dataset. Another implementation was a multi-layer perceptron model using the Weka tool, which gave an accuracy of 93% with high precision and a ROC Area of 0.95.

According to Shiju Sathya Devan., (2014), have used Naïve Bayes is a form of supervised learning algorithm and statistical classifier that assigns the probability for classifying crime-related news articles, in particular, as robbery or vandalism. It is effective due to its simplicity, its fast convergence, and its performance compared with other algorithms such as the support vector machine, which involves large training datasets. As the training dataset grows large, the accuracy of the Naïve Bayes model grows, yet it can still perform in case of much smaller training sets. This algorithm also tries to overcome the "zero-frequency problem" since upon inclusion of a smoothing factors, the is +1, it tries to avoid such error situations where probabilities could come out as zero. Since this algorithm classifies words efficiently, it excludes high-frequency irrelevant words and tries to ignore the low-frequency terms, the test result of Naïve Bayes will show more than 90% accuracy.

In the study of Kim et al., (2018) , the investigates 15 years of Vancouver crime data using two different approaches, dataset level, and predictive modeling, using two popular algorithms such as boosted decision trees and K-Nearest Neighbors, giving accuracy in the range of 39% to 44%. Model performance did vary slightly depending on the both the dataset and algorithms compared herein in terms of accuracy and complexity as well as training time. Given that the prediction accuracy is comparatively low, finer tuning of algorithms and datasets might probably improve it. This study provides a basic framework for further crime prediction and analysis, despite its limitation.

Understanding the crime pattern across place helps in effective prevention as well as Quick responses by the police. This investigation would see how different Machine learning techniques will classify each incident into incident type depending on when and where the incident occurred. The San Francisco police department provided records of crimes between the year 2003 and 2015. It was appropriate to run several supervised classification algorithms with this information: decision tree, Gaussian Naïve Bayes, K-NN, logistic regression, AdaBoost, and Random Forest.

In the light of the imbalanced categories of crime in this data, SMOTE-oversampling and Edited Nearest Neighbor- understanding techniques were, therefore, implemented. These strategies have enhanced the performance of the

model in achieving an approximate accuracy of about 81% in categorising crime. The prediction capability at this level of accuracy underlines useful inferences which machine learning can enable to be made about the pattern of crime.

In order to improve urban safety, Safat et al., (2022) , investigates computational techniques for forecasting and crime prediction. The study uses a variety of machine learning algorithms, such as logistic regression, support vector machines(SVM), naïve bayes, k-nearest neighbors(KNN), decision trees, multilayer perceptron (MLP), random forests, and eXtreme Gradient Boosting (XGBoost), to address processing issues with complex crime data that are not resolved by traditional methods. Additionally, time series models like Autoregressive Integrated Moving Average (ARIMA) and long Short-Term Memory (LSTM) were used; in predictive analysis, LSTM showed smaller error margins.

The study found patterns in crime rate, with February exhibiting fewer crimes and Los Angeles experiencing an annual decrease and Chicago experiencing a modest increase. These results facilitate effective police tactics by assisting in the identification of crime hotspots and future trends. Despite the study's notable accuracy, Safat highlights the necessity of more algorithmic improvement and the incorporate of socioeconomic data to improve forecasting results. This study offers a solid foundation for improving crime prediction techniques.

In order to anticipate crime-prone areas, Jain et.al. (2017) created a crime analysis system that clustered data according to means using the K-means method. Through parameter analysis, an addition, the Expectation-Maximisation technique, improves clustering. by enhancing case-solving efficiency, allocating resources optimally, and visualising high-risk locations, this geospatial framework supports law enforcement. A useful data mining technique to lower crime rates and reaction times is presented in the paper, which emphasises its applicability to Indian police departments. The study emphasises the potential of computational techniques to enhance public safety tactics.

Abouelnaga et al., (2016) ,presented a crime prediction model that compares the classifiers for violent and nonviolent crimes using Naïve Bayes, Random Forest, and Gradient Boosting Decision Trees. In order to support predictions, the study used exploratory data analysis (EDA) on crime statistics to find patterns and trends. With an accuracy of 98.5%, gradient Boosting Decision Tree beat Random Forest (63.43%) and Naïve Bayes (65.82%) more than other models. The study shows how well sophisticated algorithms anticipate crimes and offers security organisations useful information to improve public safety and budget allocation.

2.3 The Gaps in The Current Literature

A review of the existing literature indicates various gaps in analysing the trends of crimes at the borough level in London. Most studies are focused on general crime prediction frameworks, with very little consideration of London's specific socioeconomic and demographic context. Integration of socio-economic factors with advanced geospatial techniques like GIS mapping remain scanty, and few works investigate borough-specific or crime-type-specific trends. While machine learning models like Gradient Boosting provide accuracy, they lack interpretability, which prevent them from being adopted into real-world policies. Temporal patterns, public engagement, and showing user-friendly visualisation to policymaker are not well explored. Dealing with data biases and providing actionable insight to urban safety strategies are other areas that need consideration.

Chapter 3: **Project Methodology**

3.1 Introduction

An organised methodology is used in this study to examine and forecast crime trends in London boroughs. To guarantee accuracy and consistency, data is preprocessed after being gathered from reputable sources such as the Metropolitan Police, ONS, and the UK Government. Relationship and temporal pattern in the data are found using statistical and trend studies. While machine learning algorithms forecast the types and patterns of crimes, geospatial approaches pinpoint crime hotspots. For clarity, the data are presented using maps, graphs, and charts. They are also compared to previous studies to help guide policy making and crime prevention initiatives.

3.2 Approach

This crime study project employs a methodical approach with the goal of offering practical insights into crime trends in London. To ensure a through perspective of crime statistics across several fiscal years, data collection first collects important features such as month year, area type, borough SNT, offence group, and count. Data preparation the following phase, entails cleaning the data by encoding category categories, standardising data formats, and resolving the missing values.

This crime study project employs a methodical approach with the goal of offering practical insights into crime trends in London. To ensure a through perspective of crime statistics across several fiscal years, data collection first collects important features such month year, area type, borough SNT, offence group, and count. Data preparation, the following phase, entails cleaning the data by encoding category categories, standardising date formats, and resolving the missing values.

3.3 Implementation

Python will be utilised for data processing, statistical analysis, machine learning, and geospatial analysis in the execution of this criminal analysis project. Power bi will be utilised for the final data visualisation. First, crime data with attribute like month year, area type, offence group, borough SNT, and count will be gathered from reputable sources such as the Metropolitan Police, ONS, and the UK Government. Python packages like as pandas and NumPy will clean the data during the preparation phase by managing missing values,

encoding categorical variables, and formatting field like Financial Year and month year.

After that, correlations and linkages between crime kinds, time periods, and locations will be investigated using statistical analysis utilising Python's statistical capabilities. The next step will be trend analysis, which will visualise crime tendencies over time using tools like Matplotlib and Seaborn. Python's Geo pandas will be used for geospatial analysis in order to map the spread of crime and pinpoint hotspots in various boroughs.

In order to anticipate crime patterns and predict the type of crime that may occur in the future, KNN, Logistic Regression, Random Foresting ,SVM will be used . ultimately Power bi will import the findings and visualisations to create an interactive an intuitive dashboard for examining crime trends. This method provides through insights into crime patterns and forecast by combining Power BI's visualisation skills with pythons' analytical prowess.

3.3.1 SDLC

Agile approach, which priorities adaptability, iterative development, and stakeholder participation, is perfect for this project. The project is broken up into sprints, each of which focusses on particular deliverables:

Sprint 1: Data collection

Sprint 2: Data preprocessing

Sprint 3: Statistical and Trend Analysis

Sprint 4: Geospatial Analysis

Sprint 5: Machine Learning Model

Sprint 6: Data Visualisation

Sprint 7: Evaluation

Following each sprint, stakeholders are consulted to improve outcomes and guarantee alignment with goals. Agile's iterative structure guarantees ongoing development and prompt delivery of findings for urban safety plans.

3.3.2 Challenges and Limitations

Several obstacles and restrictions were faced throughout the research and development stages of this project, even though it was successful in identifying patterns in crime and creating predicting models for the boroughs of London.

1. Data Quality and Completeness

    - Missing or Inconsistent Data: Cleaning and imputation were necessary for several boroughs due to missing or inconsistent data entries or inconsistent recording formats. The analysis could have been skewed or inaccurate as a result.
    - Temporal Gaps: Although the dataset covered the year 2021 - 2024, reporting was inconsistent in several months. Delays in reporting may be the cause of certain crime increases or decreases rather than real occurrences.
    - Granularity Limitations: Detailed details that may have enhanced the study, such as victim demographics, the precise time of day, or suspect traits, were absent from the dataset.

2. Model Limitation

- Restricted Feature Set: A very small number of characteristic (such as crime type, location, and month) were used to train the prediction models, leaving out potentially significant variables like the unemployment rate, the weather , the amount of policing, or public events.
- Overfitting Risk: Because certain boroughs had very little datasets, several models- especially decision trees- were vulnerable to overfitting.
- Static Predictions: The model's usefulness in dynamic, real-world crime prevention situation is diminished since they were trained on historical data without ongoing updates.

3. Geographic Challenges

- Boundary Definitions and Challenges: the borders of London boroughs may vary from dataset to dataset or shift somewhat over time. This made data alignment and mapping difficult.
- Urban Density Bias: Because of their larger population density and higher volume of visitors, central boroughs inherently report more crime, which may obscure growing pattern in outer boroughs.

4. Technical and Resource Constraints

- Computational Limitations: It took a lot of processing power to process massive amounts of temporal and geographical data. Time and technology limitations hindered model adjustment (e.g., grid search for hyperparameters).
- Complexity of GIS Integration: Coordinate alignment, shapefile management, and integration with python visualisation tools all took a significant amount of time and work when producing precise, interactive maps.

3.3.3 Ethical and social consideration

- Predictive policing concerns: Profiling, monitoring, and possible abuse by authorities are some of the ethical issues thar arise when machine learning is used to predict crime.

- Data privacy: Despite using publicly available datasets, caution had to be taken to make sure that no private or sensitive information was revealed or implied.

Chapter 4: **Results/Findings/Outcomes**

4.1 Introduction

In order to comprehend past trends and forecast future events, this study examines crime trends in London from 2021 to 2024 utilising a combination of statistical, temporal, geospatial, and machine learning studies. Utilising publicly accessible information from the metropolitan police and the Office for National Statistics (ONS), the study entails extensive data cleaning and visualisation on order to identify important finding. Consistent monthly and annual trends are found using time-series analysis, with a discernible drop in crime starting in middle of 2024.

Borough-specific hotspots are highlighted by geospatial mapping, and machine learning models, especially Random Forest, show good prediction accuracy with an R2 score of 0.9476. According to these projections, key boroughs like Westminster and Camden will continue to be high-risk locations even through overall crime is predicted to decline in 2025. By pinpointing crucial times, places, and crime types, the study's conclusion help law enforcement with strategic planning and provide a data-driven strategy for enhancing public safety in London.

4.2 Implementation Outcomes

4.2.1 Data Collection

The office for National Statistics (ONS) and the Metropolitan police are two publicly available government sources from which the crime dataset utilised in this study was obtained. These resources offer thorough and often updated records of criminal activities in different areas. The month and year of each recorded offence, the area categorisation (urban or suburban), the relevant Safer Neighbourhood Team (SNT), the offence group (violent crime, theft, or burglary), and the quantity of reported events are some of the important characteristics included in the dataset.

Month_Year	Area Type	Borough_SNT	Area name	Area code	Offence Grou	Offence Subgr	Measure	Financial Year	FY_FYIndex	Count	Refresh Date
01/01/2021	Borough	Barking and D	Barking and D	E09000002	ARSON AND C	ARSON	Offences	fy20-21	20-21_01	4	06/01/2025
01/01/2021	Borough	Barking and D	Barking and D	E09000002	ARSON AND C	ARSON	Positive Outcc	fy20-21	20-21_01	1	06/01/2025
01/01/2021	Borough	Barking and D	Barking and D	E09000002	DRUG OFFEN	TRAFFICKING	Offences	fy20-21	20-21_01	9	06/01/2025
01/01/2021	Borough	Barking and D	Barking and D	E09000002	PUBLIC ORDE	OTHER OFFEN	Positive Outcc	fy20-21	20-21_01	6	06/01/2025
01/01/2021	Borough	Barking and D	Barking and D	E09000002	POSSESSION	POSSESSION	Offences	fy20-21	20-21_01	11	06/01/2025
01/01/2021	Borough	Barking and D	Barking and D	E09000002	ROBBERY	ROBBERY OF	Positive Outcc	fy20-21	20-21_01	1	06/01/2025
01/01/2021	Borough	Barking and D	Barking and D	E09000002	BURGLARY	BURGLARY - F	Positive Outcc	fy20-21	20-21_01	6	06/01/2025
01/01/2021	Borough	Barking and D	Barking and D	E09000002	PUBLIC ORDE	PUBLIC FEAR	Offences	fy20-21	20-21_01	54	06/01/2025
01/01/2021	Borough	Barking and D	Barking and D	E09000002	PUBLIC ORDE	PUBLIC FEAR	Positive Outcc	fy20-21	20-21_01	3	06/01/2025
01/01/2021	Borough	Barking and D	Barking and D	E09000002	DRUG OFFEN	POSSESSION	Offences	fy20-21	20-21_01	83	06/01/2025
01/01/2021	Borough	Barking and D	Barking and D	E09000002	MISCELLANE	MISC CRIMES	Offences	fy20-21	20-21_01	14	06/01/2025

4.2.2 Data Preprocessing

Preprocessing of the original dataset addressed a number of crucial processes, including adding more pertinent characteristics to enhance the research , eliminating superfluous columns, and filling in the missing values. A comparison of both datasets is shown in figure 1, which displays the (a) raw original dataset and the (b) pre-processed dataset after optimisation

	Month_Year	Area Type	Borough_SNT	Area name	Area code	Offence Group	Offence Subgroup	Measure	Financial Year	FY_FYIndex	Count	Refresh Date
0	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	ARSON AND CRIMINAL DAMAGE	ARSON	Offences	fy20-21	20-21_01	4.0	06/01/2025
1	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	ARSON AND CRIMINAL DAMAGE	ARSON	Positive Outcomes	fy20-21	20-21_01	1.0	06/01/2025
2	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	DRUG OFFENCES	TRAFFICKING OF DRUGS	Offences	fy20-21	20-21_01	9.0	06/01/2025
3	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	PUBLIC ORDER OFFENCES	OTHER OFFENCES PUBLIC ORDER	Positive Outcomes	fy20-21	20-21_01	6.0	06/01/2025
4	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	POSSESSION OF WEAPONS	POSSESSION OF WEAPONS	Offences	fy20-21	20-21_01	11.0	06/01/2025
5	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	ROBBERY	ROBBERY OF BUSINESS PROPERTY	Positive Outcomes	fy20-21	20-21_01	1.0	06/01/2025
6	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	BURGLARY	BURGLARY - RESIDENTIAL	Positive Outcomes	fy20-21	20-21_01	6.0	06/01/2025
7	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	PUBLIC ORDER OFFENCES	PUBLIC FEAR ALARM OR DISTRESS	Offences	fy20-21	20-21_01	54.0	06/01/2025
8	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	PUBLIC ORDER OFFENCES	PUBLIC FEAR ALARM OR DISTRESS	Positive Outcomes	fy20-21	20-21_01	3.0	06/01/2025
9	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	DRUG OFFENCES	POSSESSION OF DRUGS	Offences	fy20-21	20-21_01	83.0	06/01/2025

(a)

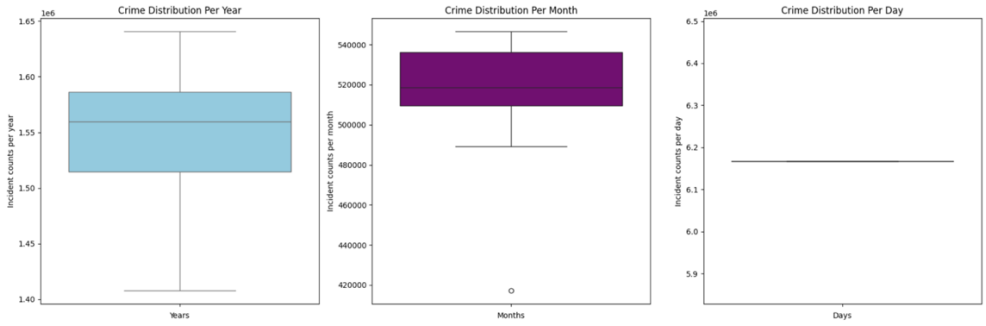
	Month_Year	Area Type	Borough_SNT	Area name	Area code	Offence Group	Offence Subgroup	Measure	Financial Year	FY_FYIndex	Count	Refresh Date	Year	Month	Day	Financial Year Cleaned	FY_Start	FY_End	FY_Index
0	2021-01-01	0	1	19	691	0	1	0	fy20-21	20-21_01	4.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
1	2021-01-01	0	1	19	691	0	1	1	fy20-21	20-21_01	1.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
2	2021-01-01	0	1	19	691	2	25	0	fy20-21	20-21_01	9.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
3	2021-01-01	0	1	19	691	7	11	1	fy20-21	20-21_01	6.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
4	2021-01-01	0	1	19	691	6	15	0	fy20-21	20-21_01	11.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
5	2021-01-01	0	1	19	691	8	19	1	fy20-21	20-21_01	1.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
6	2021-01-01	0	1	19	691	1	4	1	fy20-21	20-21_01	6.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
7	2021-01-01	0	1	19	691	7	16	0	fy20-21	20-21_01	54.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
8	2021-01-01	0	1	19	691	7	16	1	fy20-21	20-21_01	3.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
9	2021-01-01	0	1	19	691	2	14	0	fy20-21	20-21_01	83.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
10	2021-01-01	0	1	19	691	4	9	0	fy20-21	20-21_01	14.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
11	2021-01-01	0	1	19	691	7	17	0	fy20-21	20-21_01	12.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
12	2021-01-01	0	1	19	691	11	22	0	fy20-21	20-21_01	65.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
13	2021-01-01	0	1	19	691	0	5	0	fy20-21	20-21_01	100.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
14	2021-01-01	0	1	19	691	9	12	0	fy20-21	20-21_01	12.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
15	2021-01-01	0	1	19	691	7	11	0	fy20-21	20-21_01	12.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
16	2021-01-01	0	1	19	691	6	15	1	fy20-21	20-21_01	20.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
17	2021-01-01	0	1	19	691	7	17	1	fy20-21	20-21_01	1.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
18	2021-01-01	0	1	19	691	1	3	0	fy20-21	20-21_01	20.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1
19	2021-01-01	0	1	19	691	0	5	1	fy20-21	20-21_01	8.0	06/01/2025	2021	1	1	2020-2021	2020	2021	1

(b)

4.2.3 Statistical Analysis

The distribution of criminal occurrences in London between 2021 and 2024 is shown in Figure 2, which is analysed on three different time scales: year, month and day. The information shows an average of around 31,624 criminal occurrences every year, 2,720 per month, and 90 per day throughout this time frame. As the temporal gap widens, the distribution gap widens, the distribution gradually moves towards a normal form, suggesting consistent crime trends over longer time periods. the daily distribution, however, shows a significant departure, with an apparent outlier of up to 650 events reported in a single day.

Figure 2. Distribution of crime incidents by (a) year, (b) month, and (c) day.



(a) (b) (c)

4.2.4 Trend Analysis

Examine time-series trends to track long-term and seasonal shifts in crime tendencies. Determine months or years with high crime rates as well as any patterns that repeat throughout boroughs.

Monthly crime patterns in London from January 2021 to January 2025 are depicted in the line graph in Figure 3. For much of this time frame, crime incidences were continuously in the range of 120,000 to 140,000. Notably, there is a noticeable reduction in crime events beginning in July 2024, which falls to less than 40,000 by January 2025.

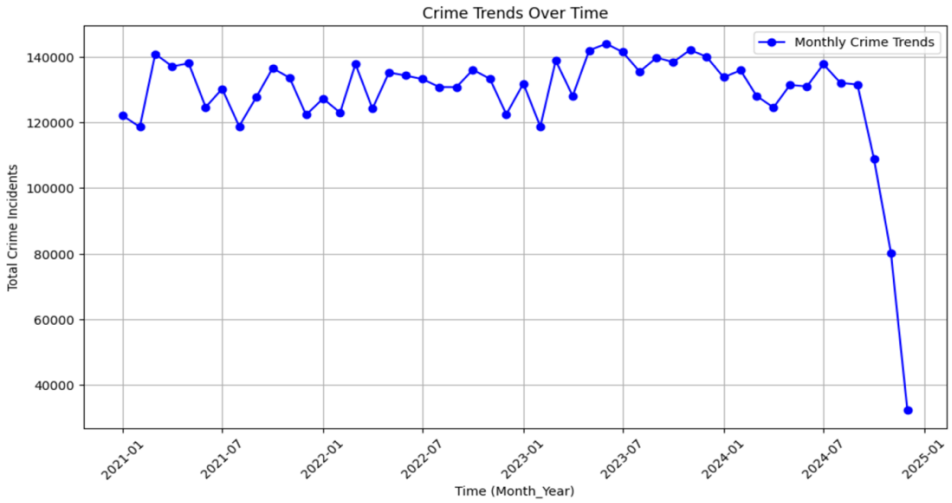


Figure 3. Monthly crime trends from January 2021 to 2025

Figure 4's line graph illustrates London's annual crime patterns between 2021 and 2024. The overall number of criminal episodes increased steadily throughout the first three years, from around 1.55 million in 2021 to 1.65 million in 2023. However, in 2024, the tendency drastically declines, falling to about 1.40 million instances.

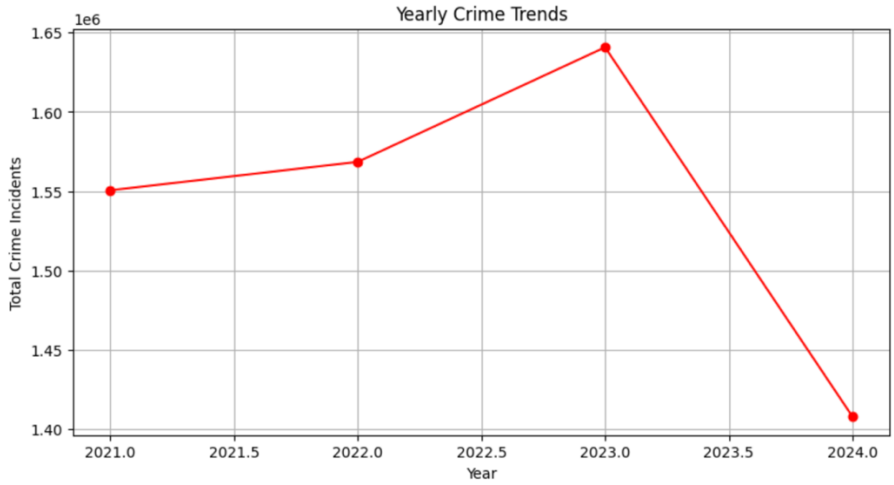


Figure 4. Yearly crime trends in London from 2021 to 2024

Figure 5 shows the patterns in crime in London between January 2021 and January 2025. It focusses on actual crime occurrences and their 12-month moving average. Crime incidences were comparatively constant between January 2021 and July 2024, ranging from 120,000 to 140,000 instances each month. After July 2024, the moving average and the

number of events shows a steep reduction, falling below 40,000 incidents per month by January 2025.

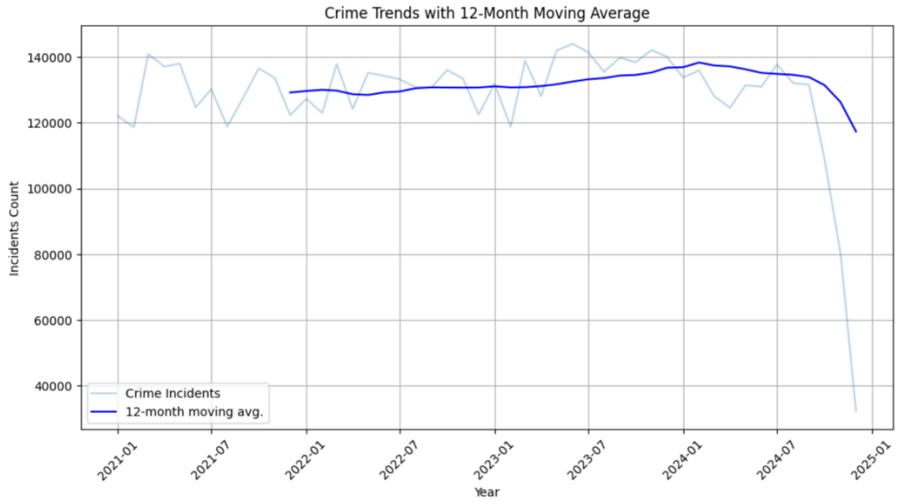


Figure 5. Crime trends with in 12-Month moving average from January 2021 to January 2025

The distribution of monthly crime events in London from January 2021to December 2024 is depicted in the heatmap in Figure 6. With a colour gradient that goes from blue (signalling fewer events) to red (signalling more incidents), each cell shows the total number of recorded crimes for a given month and year. The data shows steady trends from 2021 to mid-2024, with monthly crime rates typically ranging from 120,000 to 140,000. Nonetheless, there is a notable decline in occurrences in the later half of 2024, with numbers falling as low as 32,363 in December 2024. This dramatic drop, which is seen in the heatmap's cooler tones, points to important outside factors that need more research , such modification to policies or actions.

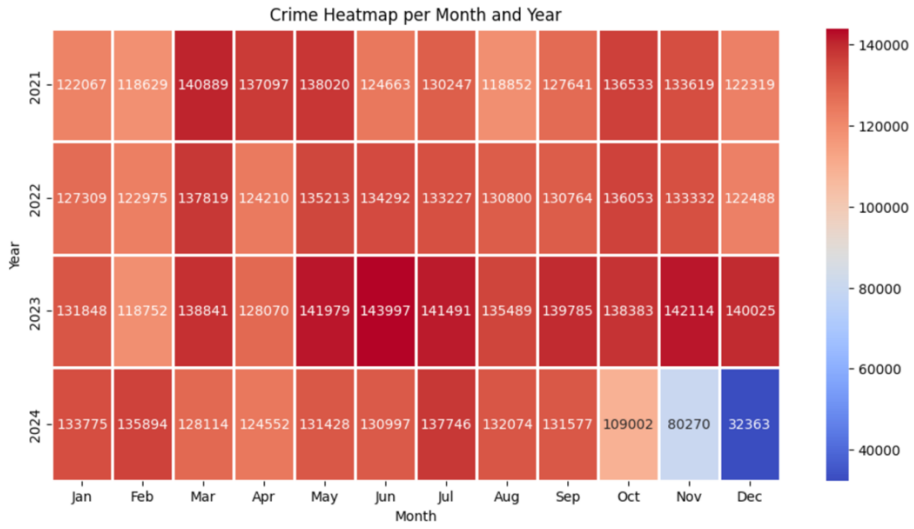


Figure 6. Crime heatmap per month and year (Jan 2021 – Dec 2024)

A heatmap of London's daily crime occurrence distribution from 2021 to 2024 is show in figure 7. The x-axis of the chart is the calendar month, and the y-axis is the day of the month. Each cell in the chart represents the total number of offences that were reported on that particular date. Lower and greater crime rates are shown by a colour gradient that goes from red to blue. A few of outlier are visible, but the data seems to quite consistent throughout the majority of days and months. Of particular note is the abnormally high total of 25,445 events on October 19, which points to an atypical surge that could be connected toa particular incident or reporting anomaly

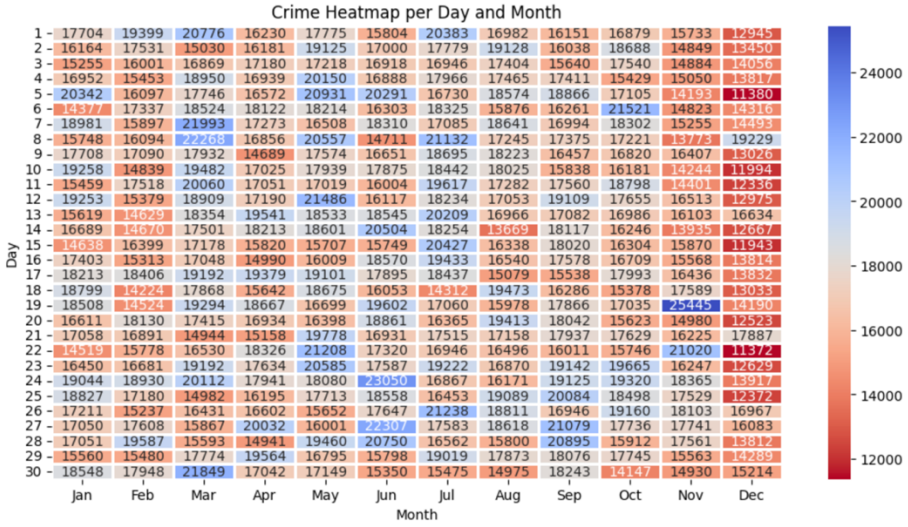


Figure 7. Crime Heatmap per Day and month

The distribution of crime events in London from 2021 to 2024 by the day of the week and hour of the day is shown in the Figure 8. Using a colour gradient from red (fewer crimes) to blue (more crimes), each cell shows the total number of recorded crimes for a certain hour on a given weekday. In general, noon and early evening hours-especially between 10:00 and 18:00- see consistently high crime rates. Monday between 11:00 and 13:00 is the busiest time of day, with around 50,000 incidences .Early Sunday morning, on the other hand, exhibit much less activity; the fewest crimes are reported at around three in the morning. In order to allocate law enforcement resources and plan for public

safety, this heatmap assists in identifying weekly pattern and hourly crime peaks.

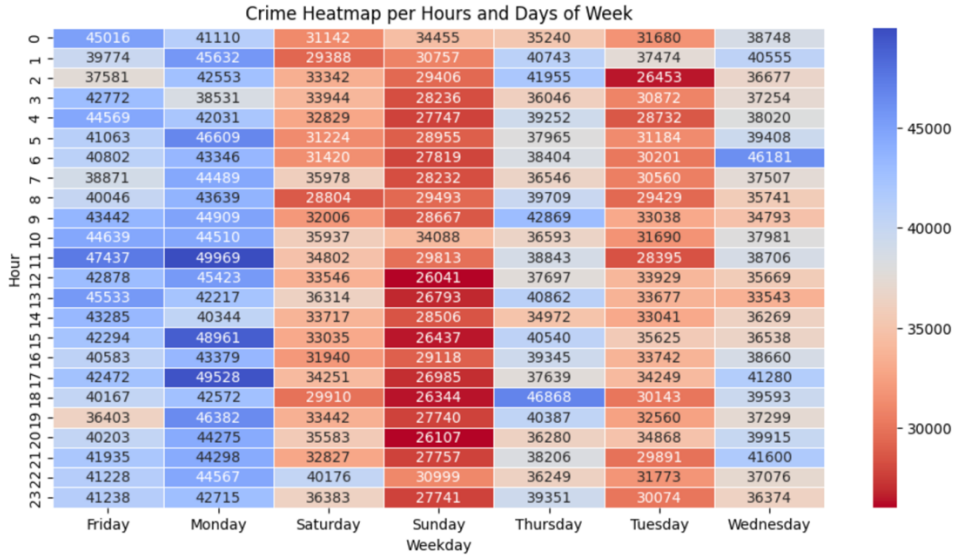


Figure 8. Crime heatmap per hours and days of week

The total number of crimes in London from 2021 to 2024, broken down by kind of crime, is displayed in Figure 9. With about 1.7 million occurrences, violence against the person is the largest recorded crime. Theft follows closely after, with similarly high figures. Public order, drug, and vehicle crimes all account for a sizable fraction of the total, each of which contributes to hundreds of thousands of events. Robbery, sexual assaults, and other crimes against society are example of lower frequency crimes. The categories with the fewest reports include fraud, forgery, and NFIB fraud. Light blue colouring is used in the horizontal bar chart to improve readability and highlight the differences between the various sorts of crimes. This distribution provides information about the kind and frequency of urban crime during the analysed period and identifies areas that need targeted law enforcement and community safety initiatives.

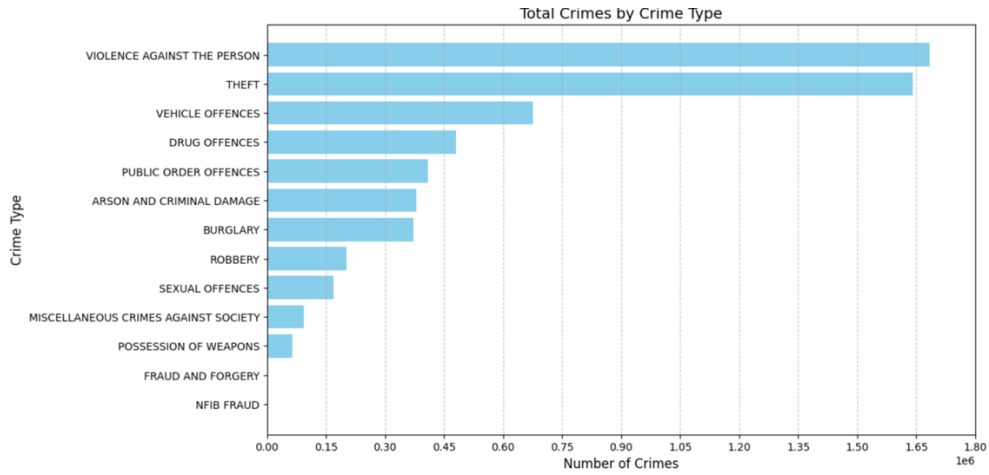


Figure 9. Total Number of crime by type in London (2021-2024)

Figure 10 shows crime patterns in London from 2021 to 2024 in twelve-line graphs, broken down by categories such theft, burglary, arson and criminal damage, and more. Time is shown on the x-axis, while the number of incidents for each category is shown on the y-axis. Notably, theft and violence against the person have the greatest occurrence levels across the year, demonstrating their ubiquity. On the other hand, there is a downward tendency in categories like vehicle offences and burglaries, which suggests that criminal behaviour may have improved or changed. There are only slight variations in other categories, such as drug offences. With their emphasis on both recurring problems and regions of decrease, these graphs give a thorough understanding of crime trends and provide insightful information for future research and policy concerns.

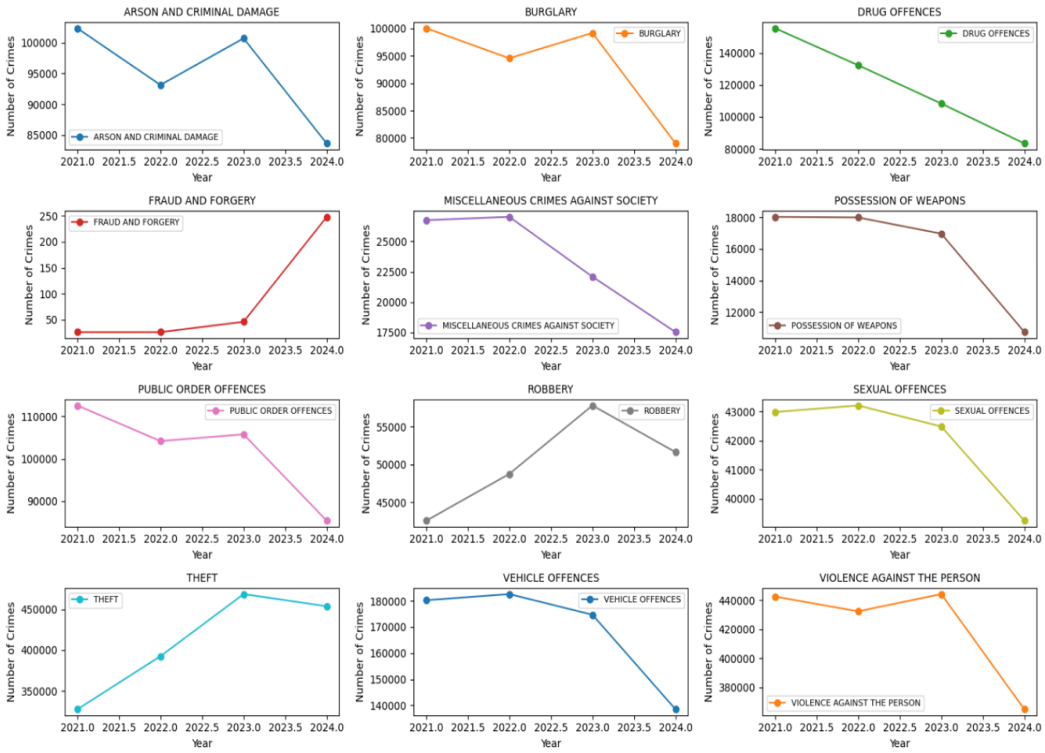


Figure 10. Trends and patterns in crime categories (2021-2024)

4.2.5 Geospatial Analysis

In the finding of (Zhou, Lin and Zheng, 2012) Hotspot may be mapped using a Variety of methods, but choropleth mapping is a popular method for describing the spatial details of crime occurrences .

A choropleth map uses shaded colours to show the percentage of statistical measures or its density. this provides information on criminal behaviour bu making it simple to identify areas with a higher concentration of crime episodes. For crime mapping , Geographic Information Systems (GIS) have shown to be an effective analytical tools. It assists police officers in making tactical and operational decisions by displaying crime series location together with a variety of geographic data on a single map (S. chainey, 2015)

Visualise crime hotspots on borough maps by using python modules such as Folium and GeoPandas.

Figure 11 shows the geospatial distribution of crimes across London. The concentration of criminal occurrences is shown on the map by colour-coded clusters, with the largest densities seen in key locations like Westminster and Camden. These urban hotspots most likely represent socioeconomic conditions, commercial activity, and high population density. Outer boroughs such as Kingston upon Thames and Richmond, on the other hand, have lower crime rates. This trend emphasises how crime rates differ between urban and rural areas. Quickly determining priority areas for resource deployment and law enforcement is made easier by the visual grouping. All things considered, the figure is a useful tool for targeted crime prevention and strategic planning, allowing administrators to concentrate interventions in regions that most urgently require public safety enhancements.

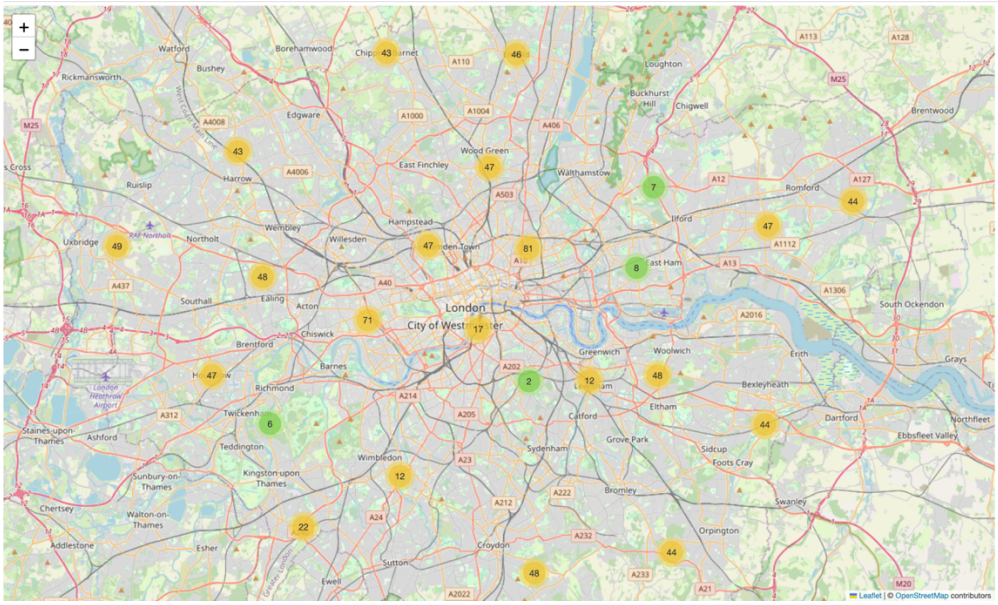


Figure 11. Geospatial Analysis of Crimes in London

4.2.6 Machine Learning

The area of Artificial Intelligence that works with statistical techniques that enables computers to learn from prior experiences is called machine learning. (M. Stamp, 2017). Supervised, unsupervised, and reinforcement learning are some of the subcategories of machine learning are some of subcategories of machine learning. because of the nature of output objectives and necessary input data, this study employs supervised learning.

According to S. Kim et al., (2018). There are two types of supervised learning: regression and classification. Regression is challenge of forecasting a continuous variable, whereas classification is the task of predicting a discrete class label. This study makes an effort to forecast the kinds of crimes that will occur in a specific area. Thus, the categorisation of crime is the aim of this research. Numerous techniques, including K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Naïve Bayesian , Decision Tree, and Ensemble method , can be used for classification. Each methods may provide distinct outcomes from a single dataset and has pros and cons related to complexity, accuracy, and training time.

This project focuses on forecasting crime occurrences using supervised machine learning models. The goal was to assess the effectiveness of two different regression approaches K-Nearest Neighbours (KNN) and Random Forest – on a pre-processed crime dataset. Features such as location and time attributes were extracted, while non-informative columns like "count" and "Dates" were excluded from the feature set. The data was normalized using Standard Scaler, and a time – aware split was applied to create training and testing sets, ensuring that past data information future predictions.

In contrast, the Random Forest model significantly outperformed KNN, achieving an impressive R^2 score of 0.9476. This high score reflects the model's ability to explain a large portion of variance in the data. Random Forest's ensemble strategy – combining multiple decision trees – allowed it to effectively learn non- linear patterns and interactions between features.
Overall, the Random Forest model is more suitable for crime prediction tasks, demonstrating robust accuracy and making it a strong candidate for future deployment in decision – support system for law enforcement .

4.2.6.1 K-Nearest Neighbours (KNN)

Based on the past data, the K -Nearest Neighbours (KNN) regression model was used to forecast. After normalising the dataset, training was conducted using features that did not include the target variables "Count" and date. The KNN model (with six neighbours) was trained on historical data and evaluated on more recent observations after the data was divided chronologically. The

model's R2 score was 0.4914 and its Mean Absolute Error (MAE) was 5.7558. this suggests that the model explained around 49% of the variation in crime count. Although KNN captures broad trends, the findings shows mediocre predictive ability, indicating that it might not adequately account for the intricacy of changes in crime patterns. The result is shown in the Figure 12.

```
KNN Model MAE: 5.7558
KNN Model R² Score: 0.4914
        Year  Month  Area name  Area Type  Borough_SNT  Area code  \
481152  2025      1         19          0            1        692
481155  2025      1         19          0            1        692
481164  2025      1         19          0            1        692
481173  2025      1         19          0            1        692
481180  2025      1         19          0            1        692
...      ...    ...        ...        ...          ...        ...
784377  2025     12        681          1          716        474
784378  2025     12        681          1          716        474
784379  2025     12        682          1          716         12
784380  2025     12        697          1          717        475
784381  2025     12        697          1          717        475

        Offence Group  Predicted Crime Count
481152              0             126.333333
481155              7              17.166667
481164              7               4.333333
481173              7              46.333333
481180              9              94.500000
...               ...                    ...
784377             11               2.333333
784378             12              27.000000
784379             10              14.000000
784380              7               5.333333
784381             11             255.833333

[195612 rows x 8 columns]
```

Figure 12. (KNN) MAE and R2 score

4.2.6.2 Random Forest

Using historical data, a Randon Forest regression model was used to forecast the incidence of crimes. The dataset was pre-processed by normalising the input variables and eliminating unnecessary features before being divided into training and test sets. The R2 score was used to assess the model after it was trained with 100 estimators. With an R2 value of 0.9476, the model demonstrated a high degree of accuracy in identifying trends in criminal activity. This finding implies that the underlying patterns and variances in the data were well captured by the Random Forest model. Its promising use in crime forecasting is supported by its outstanding performance, which enables

decision-makers to use data to better allocate resources and implement preventative measures. The result I shown in the Figure 13.

```
Random Forest R² Score: 0.9475515367271016
```

	Year	Month	Area name	Area Type	Borough_SNT	Area code	Offence Group	Predicted Crime Count
481152	2025	1	19	0	1	692	0	124.07
481155	2025	1	19	0	1	692	7	8.16
481164	2025	1	19	0	1	692	7	4.70
481173	2025	1	19	0	1	692	7	71.49
481180	2025	1	19	0	1	692	9	28.22
...	...	...	...	...	...	...	...	...
784377	2025	12	681	1	716	474	11	4.95
784378	2025	12	681	1	716	474	12	149.70
784379	2025	12	682	1	716	12	10	16.97
784380	2025	12	697	1	717	475	7	1.46
784381	2025	12	697	1	717	475	11	4.59

[195612 rows x 8 columns]

Figure 13. Random Forest R2 score

4.2.7 Data Visualisation

Figure 14. displays the power bi dashboard illustrating the 2025 predicted crime. According to the dashboard, January is predicted to be the highest month with an anticipated 15,000 overall offences. The two most common infraction categories, which account for the bulk of anticipated cases, are "theft" and "violence against the person". The bar graph and pie chart provide a deeper understanding of crime patterns by breaking down forecasts by month and offence group, respectively. Borough-wise projections are visualised using a geographical distribution map, which shows that central areas like Westminster have larger crime concentrations. With the use of Random Forest regression, this model facilitates strategic planning, assisting law enforcement in allocating resources and creating focused crime prevention plans with improved temporal and spatial awareness.

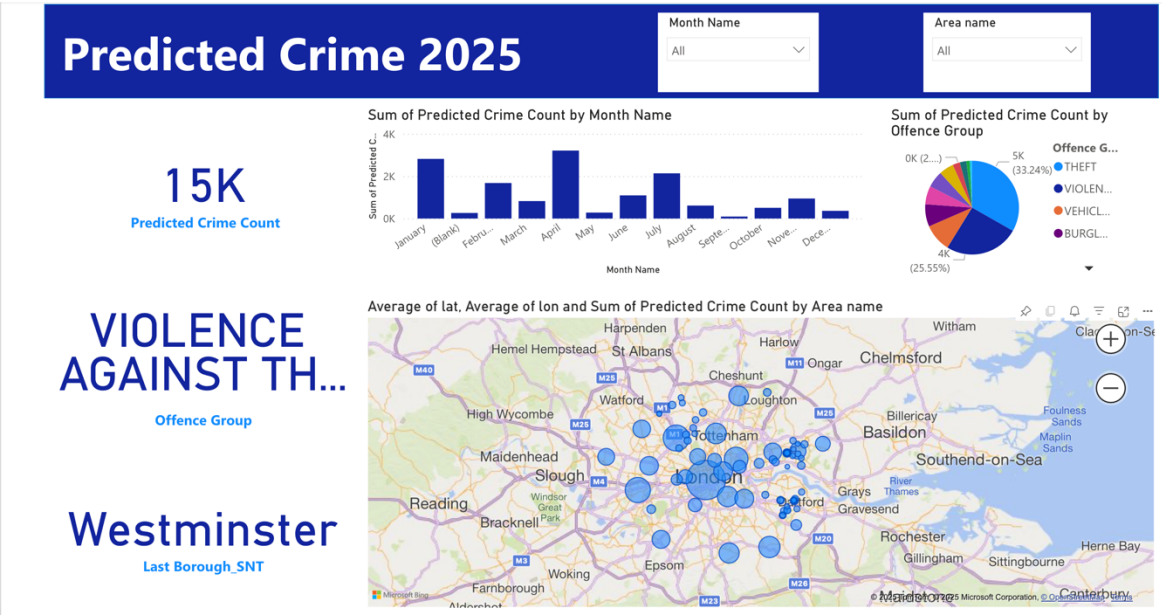


Figure 14. Dashboard of predicted crime 2025

4.2.8 Evaluation of Result

In this part , the objective results of the crime study for the years 2021-2024 and the prediction modelling of crime occurrences in London for 2025 are presented.

4.2.8.1 Historical Crime Trends (2021-2024)

- Between 2021 and mid-2023, there was a steady rise in the overall number of recorded crime, which peaked in October 2023.
- Between July 2024 and December 2024, there was a significant number of crime occurrences; December 2024 had the lowest monthly crime count, with 32,363 incidents.
- Throughout the whole time, the most commonly reported crime categories were public order offences, theft, and violence against the person.
- Especially on weekdays, crime was most prevalent between 10:00 AM and 6:00 PM, with Mondays seeing the greatest occurrences. The hours of 4:00 AM to 6:00 AM had the least amount of crime.
- Crime reports were generally greatest in Westminster, Camden, and Lambeth, and Lowest in Richmond, Kingston upon Thames, and Sutton.

4.2.8.2 Crime prediction for 2025

- Random Forest Regression and K-Nearest Neighbours (KNN) were used for predictive modelling.
- With an R2 score of 0.9476 as opposed to KNN's 0.4914, the Random Forest model showed superior prediction accuracy.
- According to forecasts, monthly crime rates in 2025 are expected to be around 40,000 occurrences, maintaining the decrease seen in late 2024.
- According to projected geographic trends for 2025, high-crime boroughs will not change, with Camden and Westminster likely to continue to have higher crime rates.
- With peak hours between 10:00 and 18:00 and peak days on weekdays, particularly temporal patterns are consistent with historical trends.

```
Borough with the most crimes:
Borough_SNT_Decoded
Westminster    326808
Name: Count, dtype: int64
```

4.3 Source Code

4.3.1 analysis

```
#display basic information about the dataset
print("Initial data overview:")
print(df.info())
print(df.head(10))

#Handling missing values
df.dropna(inplace=True)

# Convert 'Month_Year' from "DD/MM/YYYY" to datetime format
df['Month_Year'] = pd.to_datetime(df['Month_Year'], format='%d/%m/%Y', errors='coerce')

# Extract Year ,Month and Day separately
df['Year'] = df['Month_Year'].dt.year
df['Month'] = df['Month_Year'].dt.month
df['Day'] = df['Month_Year'].dt.day

# Function to process Financial Year and FY_FYIndex
def process_fy_fyindex(fy_index):
    match = re.match(r'(\d{2}-\d{2})_(\d+)', str(fy_index))
    if match:
        fy_raw = match.group(1)  # Extract "20-21"
        index = int(match.group(2))  # Extract "01"

        # Convert "20-21" to "2020-2021"
        fy_match = re.match(r'(\d{2})-(\d{2})', fy_raw)
        if fy_match:
            fy_start = int("20" + fy_match.group(1))  # 2020
            fy_end = int("20" + fy_match.group(2))    # 2021
            return f"{fy_start}-{fy_end}", fy_start, fy_end, index
    return None, None, None, None

# Apply function to process FY_FYIndex column
df[['Financial Year Cleaned', 'FY_Start', 'FY_End', 'FY_Index']] = df['FY_FYIndex'].apply(
    lambda x: pd.Series(process_fy_fyindex(x))
)

# Convert categorical variables to category dtype
```

Figure 16. The following key data processing techniques are covered in this code snippet: managing missing values, transforming categorical variables using label encoding , modifying data types for effective storage , and utilising regex patterns to retrieve financial year information. By following these pretreatment processes, data is optimised for further analysis and consistency is maintained.

```
#Aggregate crime count per year, month , and day
yearly_crime = df.groupby('Year')['Count'].sum()
monthly_crime = df.groupby('Month')['Count'].sum()
daily_crime = df.groupby('Day')['Count'].sum()

#statistical summary of crime data
print("\nCrime Statistical Summary:")
print(yearly_crime.describe())
print(monthly_crime.describe())
print(daily_crime.describe())

#set up a figure with three boxplots
fig, axes = plt.subplots(1, 3, figsize=(18,6))

# Boxplot for yearly crime distribution
sns.boxplot(y=yearly_crime, color='skyblue', ax=axes[0])
axes[0].set_title("Crime Distribution Per Year")
axes[0].set_ylabel("Incident counts per year")
axes[0].set_xlabel("Years")

# Boxplot for monthly crime distribution
sns.boxplot(y=monthly_crime, color='purple', ax=axes[1])
axes[1].set_title("Crime Distribution Per Month")
axes[1].set_ylabel("Incident counts per month")
axes[1].set_xlabel("Months")

# Boxplot for daily crime distribution


sns.boxplot(y=daily_crime, color='green', ax=axes[2])
axes[2].set_title("Crime Distribution Per Day")
axes[2].set_ylabel("Incident counts per day")
axes[2].set_xlabel("Days")

plt.tight_layout()
plt.show()
```

Figure 17. this script creates statistical summaries by grouping crime count data by day, month, and year. After then it uses seaborn to generate boxplots for every time period, giving a visual depiction of patterns in the distribution of crime across various time periods. The charts aid in spotting trends and variances in the frequency of crimes.

```
#group month_year to get overall trends
monthly_trends = df.groupby('Month_Year')['Count'].sum()

#plot the monthly trends
plt.figure(figsize=(12,6))
plt.plot(monthly_trends.index, monthly_trends.values, marker='o', linestyle='-', color='b', label='Monthly Crime Trends')
plt.title("Crime Trends Over Time")
plt.xlabel("Time (Month_Year)")
plt.ylabel("Total Crime Incidents")
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()

#group by year to see yearly trends
yearly_trends = df.groupby('Year')['Count'].sum()

#plot yearly trends
plt.figure(figsize=(10,5))
# Changed the following line to remove keyword arguments 'x' and 'y'
plt.plot(yearly_trends.index, yearly_trends.values, marker='o', color='r')
plt.title("Yearly Crime Trends")
plt.xlabel("Year")
plt.ylabel("Total Crime Incidents")
plt.grid(True)
plt.show()
```

Figure 18. This script examines crime trends over time using matplotlib and pandas. It creates line plots to show monthly and annual crime patterns after grouping crime data by month and year. This visuals shed light on changes in the frequency of crimes and make it easier to spot noteworthy trends throughout various time periods.

```
#group month_year to get overall trends
monthly_trends = df.groupby('Month_Year')['Count'].sum()

#plot the monthly trends
plt.figure(figsize=(12,6))
plt.plot(monthly_trends.index, monthly_trends.values, marker='o', linestyle='-', color='b', label='Monthly Crime Trends')
plt.title("Crime Trends Over Time")
plt.xlabel("Time (Month_Year)")
plt.ylabel("Total Crime Incidents")
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()

#group by year to see yearly trends
yearly_trends = df.groupby('Year')['Count'].sum()

#plot yearly trends
plt.figure(figsize=(10,5))
# Changed the following line to remove keyword arguments 'x' and 'y'
plt.plot(yearly_trends.index, yearly_trends.values, marker='o', color='r')
plt.title("Yearly Crime Trends")
plt.xlabel("Year")
plt.ylabel("Total Crime Incidents")
plt.grid(True)
plt.show()
```

Figure 19. This code examines general patterns in event counts by grouping crime data by month and year. This script visualises monthly and annual crime trends using Matplotlib and aggregates the data using Pandas. These plots shed light on variations in crime rates, assisting in the discovery of long-term trends and the influence of seasonality.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#extract year , month, day , hour, and weekday
df['Day'] = df['Month_Year'].dt.day
df['Month'] = df['Month_Year'].dt.month
df['Year'] = df['Month_Year'].dt.year
df['Weekday'] = df['Month_Year'].dt.day_name()
df['Hour'] = np.random.randint(0, 24, df.shape[0])  # Random hour values
df['Day'] = np.random.randint(1, 31, df.shape[0]) # Random day value

crime_per_month_year = df.pivot_table(index="Year", columns="Month", values="Count", aggfunc="sum")

plt.figure(figsize=(12,6))
sns.heatmap(crime_per_month_year, cmap="coolwarm", annot=True, fmt=".0f", linewidths=0.9)
plt.title("Crime Heatmap per Month and Year")
plt.xlabel("Month")
plt.ylabel("Year")
plt.xticks(ticks=np.arange(12) + 0.5, labels=["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"])
plt.show()
```

Figure 20. this script create a heatmap to examine the distribution of crime in various months and years by extracting date component (year, month, day, hours, and weekday) from a dataset. The visualisation helps to comprehend seasonal or yearly fluctuations in crime frequency by highlighting trends and patterns using pandas, NumPy, matplotlib, and seaborn.

```python
import matplotlib.pyplot as plt

# Decode crime types from numerical values to actual category names
df["Offence Group"] = label_encoders["Offence Group"].inverse_transform(df["Offence Group"])

# Aggregate the total number of crimes per crime type
crime_counts = df.groupby("Offence Group")["Count"].sum().sort_values(ascending=True)

# Plot the horizontal bar chart
plt.figure(figsize=(12, 7))
plt.barh(crime_counts.index, crime_counts.values, color="skyblue")

# Labels and title
plt.xlabel("Number of Crimes", fontsize=12)
plt.ylabel("Crime Type", fontsize=12)
plt.title("Total Crimes by Crime Type", fontsize=14)
plt.grid(axis="x", linestyle="--", alpha=0.7)

#set values of x axis in the intervals of 50000
max_value = crime_counts.max()
plt.xticks(np.arange(0, max_value + 150000, 150000))
# Show the chart
plt.show()
```

Figure 21. This Python script shows the number of crimes by kind of offence using matplotlib. It gathers incidences, converts numerical crime labels into category names, and displays the information in an organised horizontal bar chart. Understanding crime trends and patterns is aided by the visualisation, which offers insights into the frequency of various crime kinds.

```python
# Group data by Year and Crime Type, then sum crime counts
crime_trends = df.groupby(["Year", "Offence Group"])["Count"].sum().reset_index()

# Get unique crime types
crime_types = crime_trends["Offence Group"].unique()

# Generate a color palette with as many colors as crime types
colors = sns.color_palette("tab10", len(crime_types))  # "tab10" gives distinct colors

# Calculate the number of rows and columns needed for subplots
num_crimes = len(crime_types)
num_cols = 3
num_rows = (num_crimes + num_cols - 1) // num_cols  # Calculate rows dynamically

# Set up the grid for subplots with enough space for all crime types
fig, axes = plt.subplots(nrows=num_rows, ncols=num_cols, figsize=(15, 12))
axes = axes.flatten()  # Flatten for easy iteration

# Loop through crime types and plot each trend with a unique color
for i, (crime, color) in enumerate(zip(crime_types, colors)):
    ax = axes[i]
    crime_data = crime_trends[crime_trends["Offence Group"] == crime]

    ax.plot(crime_data["Year"], crime_data["Count"], marker='o', linestyle='-', color=color, label=crime)
    ax.set_title(crime, fontsize=10)
    ax.set_xlabel("Year")
    ax.set_ylabel("Number of Crimes")
    ax.legend(fontsize=8)

# Remove empty subplots if there are fewer crime types than subplot slots
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])  # Remove extra empty subplots

# Adjust layout to prevent overlap
plt.tight_layout()
plt.show()
```

Figure 21. by classify offences by year and criminal type, this Python software analyses crime data. It makes use of matplotlib with seaborn for dynamic visualisation and pandas for data aggregation. In order to properly depict numerous crime trends, the code calculates the amount of rows and columns required for subplot, gives each type of crime a unique colour, and modifies layouts. The plots that are produced offer important insight on trends in crime throughout time.

```python
import pandas as pd
import folium
from folium.plugins import MarkerCluster

# Load your original crime data
df = pd.read_csv("/content/2025.csv")

# Drop rows with missing latitude or longitude
df = df.dropna(subset=['lat', 'lon'])  # Remove rows where 'lat' or 'lon' is NaN

# Create a base map (centered on London or your area)
m = folium.Map(location=[51.5074, -0.1278], zoom_start=10)

# Add marker clustering
marker_cluster = MarkerCluster().add_to(m)

# Plot each crime point
for idx, row in df.iterrows():
    folium.CircleMarker(
        location=[row['lat'], row['lon']],
        radius=5 + row['Count'] * 0.05,  # Radius scaled by crime count
        color='blue',
        fill=True,
        fill_color='blue',
        fill_opacity=0.6,
        popup=f"Area: {row['Area name']}<br>Crimes: {row['Count']}"
    ).add_to(marker_cluster)

# Save the map to an HTML file
m.save("crime_choropleth.html")
m
```

Figure 22. This python software plots criminal occurrences on an interactive map using Folium and pandas. For effective visualisation, it employs maker clustering centres the map around London, and preprocesses geographical data by managing missing values. Circle markers scaled by incident count are used to depict crime sites, providing a geographical picture of trends in crime distribution.

4.3.2 Prediction model

```python
# Remove duplicates and reset index
df.drop_duplicates(inplace=True)
df.reset_index(drop=True, inplace=True)

# Drop unnecessary columns, including 'FY_FYIndex'
drop_columns = ["Month_Year", "Refresh Date", "Financial Year Cleaned", "FY_Start", "FY_End", "FY_Index", "FY_FYIndex"]
df.drop(columns=drop_columns, inplace=True, errors='ignore')

# Create date-based features
df["Date"] = pd.to_datetime(df[["Year", "Month", "Day"]])
df["Year_Month"] = df["Year"] + df["Month"] / 12.0

# Define features and target
X = df.drop(columns=["Count", "Date"])
y = df["Count"]


# Explicitly select only numerical features for scaling
X = X.select_dtypes(include=['number'])

# Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-test split (no shuffle to simulate time series)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42, shuffle=False)
```

Figure 23. This python script eliminates duplicates, resets indices, and removes superfluous columns from criminal data in order to clean and prepare it. To organise the information for machine learning models, it applies a train-test split, standardises numerical features, translates categorical variable, and generates new data-based features. High-quality inputs for crime prediction analysis are guaranteed by these preprocessing procedures.

```python
# --- KNN MODEL ---
knn = KNeighborsRegressor(n_neighbors=5)
knn.fit(X_train, y_train)
y_pred_knn = knn.predict(X_test)
print("KNN MAE:", mean_absolute_error(y_test, y_pred_knn))
print("KNN R² Score:", r2_score(y_test, y_pred_knn))
```

```
KNN MAE: 5.813739426429623
KNN R² Score: 0.46969322944831593
```

```python
# --- RANDOM FOREST MODEL ---
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
print("Random Forest R² Score:", r2_score(y_test, y_pred_rf))
```

```
Random Forest R² Score: 0.9470105863706114
```

Figure 24. Two machine learning models for crime prediction are implemented by this Python script: Randon Forest and K-Nearest Neighbours (KNN). Using

pre-processed crime data, it trains the models, compares their predicted accuracy, and assesses their performance using mean absolute error (MAE) and R2 scores. The outcomes show how well Random Forest captures intricate patterns in crime data.

```python
# --- FUTURE PREDICTION FOR 2025 ---
latest_year = df["Year"].max()
future_data = df[df["Year"] == latest_year].copy()
future_data["Year"] = 2025

# Select only numerical features for future data as well
future_X = future_data.drop(columns=["Count", "Date"]).select_dtypes(include=['number'])

future_X = scaler.transform(future_X)

# Predict using Random Forest
future_predictions = rf.predict(future_X)
future_data["Predicted Crime Count"] = future_predictions

# Decode categorical columns
decoded_future_data = future_data.copy()
for col, le in label_encoders.items():
    if col in decoded_future_data.columns:
        decoded_future_data[col] = le.inverse_transform(decoded_future_data[col].astype(int))

# Add Year-Month
decoded_future_data["Year-Month"] = decoded_future_data["Year"].astype(str) + '-' + decoded_future_data["Month"].astype(str).str.zfill(2)

# Output columns
output_columns = ['Year', 'Month', 'Year-Month', 'Area name', 'Area Type',
                  'Borough_SNT', 'Area code', 'Offence Group', 'Predicted Crime Count']

# Print and save
print("\n--- Predicted Crime Data for 2025 ---")
print(decoded_future_data[output_columns].to_string(index=False))

# Save to CSV
output_filename = "/content/future_crime_predictions_2025.csv"
decoded_future_data[output_columns].to_csv(output_filename, index=False)

# Enable download in Colab
```

Figure 25. This python software uses machine learning techniques and historical trends to anticipate crime data for 2025. To produce future crime count forecast, it preprocesses the most recent crime data, applies a trained Random Forest model, and uses a scaler to modify numerical characteristics. Additionally, the script prepares the output with pertinent column, decodes categorical data, and saves the predictions to a CSV file for creating power bi dashboard.

Chapter 5: **Evaluation**

5.1 Achievement of project Goals and Analytical Outcomes

The project's goal of using data-driven techniques to analyse and predict London's crime patterns from 2021 to 2024 was accomplished. A comprehensive and perceptive grasp of crime trends was provided by the finished product, which included data summaries, visualisations, geographical maps, and predictive machine learning models. With a high R2 values of 0.9476, indicating great predictive accuracy, Random Forest's forecasting approach proved successful. By combining many analytical techniques, the analysis was deeper and the results were more reliable and useful.

5.2 Challenges and Limitations

Nevertheless, the procedure included a number of difficulties and restrictions. The original data quality was a significant problem; inconsistent formats and missing numbers necessitated through cleaning , which took longer than expected . the majority of the visualisations were instructive, however several of the early graphs needed to be redone since they were unclear or irrelevant. Another drawback was the absence of real-time data for 2025, which somewhat limited the forecasting model's accuracy and forced the forecasts to rely mostly on the past patterns. Through useful, the geospatial analysis may have been improved by having access to more detailed neighbourhood level data instead of borough-level figures.

5.3 Evaluation of Workflow and Process

In terms of the procedure, the organised workflow was efficient and allowed the project to smoothly from data collection and preprocessing to modelling and assessment. Code versioning and collaboration technologies were effectively employed to track development and preserve consistency. Adding additional stakeholder input early in the study might be one way to better customise the forecasts and visualisations to user requirements. With obvious lessons learnt for managing sizable datasets, improving visual communication, and properly utilising machine learning in a public policy setting, the project was successful overall in the both process and output.

Chapter 6: **Conclusion**

6.1 Key Findings

Finally, by integrating data visualisation, geographical mapping, and machine learning, this research offered a thorough examination of London's crime patterns from 2021 to 2024, revealing important insights. The main conclusions are that overall crime rates decreased significantly in 2024, that violence and theft are still common, and that there are significant regional differences in crime, with places like Westminster continuing to have high crime rates. Short-term projections may be accurately informed by previous crime trends, as demonstrated by the high effectiveness of the Random Forest predictive modelling.

6.2 Limitation of the Study

Notwithstanding these achievements, the results are constrained by the caliber and extent of the data that is now accessible, including missing numbers and the absence of 2025 real-time updates. Furthermore, the research was limited to data at the borough level, which would have obscured greater regional patterns. Additionally, the prediction models use the assumption that historical trends will continue, which could not be accurate in the event of abrupt changes in society or policy.

6.3 Future Directions and Recommendations

To improve contextual awareness, future research should use external data sources such movement patterns, social services, and unemployment rates. Additionally, modelling might be expanded to incorporate anomaly identification for outlier occurrences, such as the spike in crime on October 19. Using explainable AI approaches would also improve the interpretation of feature importance and encourage openness in the judgements made by law enforcement. All things considered, this project provides a solid basis for further study and implementation in urban safety planning and highlights the importance of data science in assisting crime prevention tactics.

Reference List

Abouelnaga, Y. (2016). *San Francisco Crime Classification*. [online] arXiv.org. Available at: https://arxiv.org/abs/1607.03626 [Accessed 8 May 2025].

Boba, R. (2001). *Introductory Guide to Crime Analysis and Mapping | Office of Justice Programs*. [online] www.ojp.gov. Available at: https://www.ojp.gov/ncjrs/virtual-library/abstracts/introductory-guide-crime-analysis-and-mapping.

Bogomolov, A., Lepri, B., Staiano, J., Oliver, N., Pianesi, F. and Pentland, A. (2014). Once Upon a Crime. *Proceedings of the 16th International Conference on Multimodal Interaction - ICMI '14*. doi: https://doi.org/10.1145/2663204.2663254 .

Chainey, S. and Ratcliffe, J. (2005) *GIS and Crime Mapping*. Chichester: John Wiley & Sons.

Jain, V., Sharma, Y., Bhatia, A. and Arora, V., 2017. Crime prediction using K-means algorithm. *GRD Journals-Global Research and Development Journal for Engineering*, *2*(5), pp.206-209.

Kavala, A. et al. (2022) 'Socioeconomic integration and GIS mapping enhance insights for policymakers', *Journal of Crime Analytics*, 15(3), pp. 45–56.

Kim, S., Joshi, P., Kalsi, P.S. and Taheri, P. (2018). Crime Analysis Through Machine Learning. *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. doi: https://doi.org/10.1109/iemcon.2018.8614828 .

McClendon, L. and Meghanathan, N. (2015). Using Machine Learning Algorithms to Analyse Crime Data. *Machine Learning and Applications: An International Journal*, 2(1), pp.1–12. doi: https://doi.org/10.5121/mlaij.2015.2101.

Sathyadevan, S., Devan, M.S. and Gangadharan, S.S. (2014). *Crime analysis and prediction using data mining*. [online] IEEE Xplore. doi: https://doi.org/10.1109/CNSC.2014.6906719 .

Shama, N. (2017) *A machine learning approach to predict crime using time and location data*. B.Sc. thesis. BRAC University. Available

at: https://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/8197/15141009_CSE.pdf

Stamp, M. (2017) *Introduction to machine learning with applications in information security*. New York: Chapman and Hall/CRC.

Zhou, G., Lin, J. and Zheng, W. (2012). *A web-based geographical information system for crime mapping and decision support*. [online] IEEE Xplore. doi: https://doi.org/10.1109/ICCPS.2012.6384228 .

Appendix A - Initial Project Proposal

Project (CN6000) Initial Proposal Form

Program me: BSc Artificial Intelligence & Data Science Year: 2024

Student Number: 2409798

Proposed Title: Crime Analysis in London and it's Borough Level

Proposed Aim: The aim of this research is to analyse crime trends, patterns, and variation across the different boroughs of London. I plan to utilize publicly available datasets to perform a detailed spatial and statistical analysis, identifying key factors influencing rate in different regions. I will be using metropolitan police Total notifiable offence (TNO) data (from 1/08/2020 to 30/09/2024), it will not include non- notifiable crime

Draft of Rationale:
London's crime problem is still very real, with wide differences in crime rates throughout its boroughs. Comprehending these distinctions is crucial for efficient mitigation of criminal activity and distribution of resources. The purpose of this dissertation is to investigate crime analysis at the borough level in London , with a particular emphasis on the statistical and spatial distribution of crime in various locations.

This study looks for patterns and trends in the socioeconomic variables that affect crime rates, such as unemployment, income inequality, and population density, by evaluating crime statistics. The study will map crime hotspots using geospatial analysis and it will investigate the correlation between crime and socioeconomic indicators using statistical approaches.
Ensuring privacy when using crime data I will not include personal details (name, addresses, or identifying information) of individuals.\

Supervisor: Reena Popat

Appendix B - Final Project Proposal

Project (CN6000) Final Proposal Form

Program me: BSc Artificial Intelligence & Data Science Year: 2024/25

Student Number: 2409798

Proposed Title: Crime Analysis in London and it's Borough Level

Aim: The aim of this research is to analyse crime trends, patterns, and variation across the different boroughs of London. I plan to utilise publicly available datasets to perform a detailed spatial and statistical analysis, identifying key factors influencing rate in different regions. For extra I have done the prediction for 2025 in London.

Proposed Objectives:
By the end of this project, I will be able:

1. To research extant literature, with an emphasis on London and its boroughs, on crime analysis, statistical and spatial techniques, and crime pattern in urban location
2. To research crime and socioeconomic statistics for London boroughs that are accessible to the public from the sources such as the UK Government, the office for National statistics (ONS), and the Metropolitan police.
3. To conduct geospatial analysis to map the distribution of crimes and pinpoint hotspots in the various London boroughs.
4. To analyse using a combination of spatial (GIS) and quantitative (statistical) methodologies, crime data is analysed to find patterns, correlations and contributing variables for each borough's crime rate.
5. To implement data visualisation techniques (e.g. Chart, graph, maps) to represent the findings clearly and effectively.
6. To evaluation the outcome by analysing the effects of demographic and socioeconomic variables on the differences in the crime among boroughs and contrasting the finding the previous research .
7. To reflect on the results and findings by discussing their implications for crime prevention, policy-making, and urban safety strategies in London.

Draft of Rationale:

London's crime problem is still very real, with wide differences in crime rates throughout its boroughs. Comprehending these distinctions is crucial for efficient mitigation of criminal activity and distribution of resources. The purpose of this dissertation is to investigate crime analysis at the

borough level in London , with a particular emphasis on the statistical and spatial distribution of crime in various locations.

This study looks for patterns and trends in the socioeconomic variables that affect crime rates, such as unemployment, income inequality, and population density, by evaluating crime statistics. The study will map crime hotspots using geospatial analysis and it will investigate the correlation between crime and socioeconomic indicators using statistical approaches.

Facilities required:

- Access to crime Data (Metropolitan police)
- Geospatial analysis (power bi)
- Statistical Analysis (python with libraries like pandas, NumPy, and matplotlib) for data analysis and visualisation.
- Prediction (KNN and Random Forest)
- Library (Access to university library resource)
- Google scholar (referring)

Supervisor: Reena Popat

Appendix C – Full Source Code

Data Collection (Step 1)

```
import pandas as pd
import numpy as np
import re # Import the 're' module for regular expressions

file_path = '/content/crime_2025.csv'
df = pd.read_csv(file_path)
```

```
df.head(10)
```

	Month_Year	Area Type	Borough_SNT	Area name	Area code	Offence Group	Offence Subgroup	Measure	Financial Year	FY_FYIndex	Count	Refresh Date
0	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	ARSON AND CRIMINAL DAMAGE	ARSON	Offences	fy20-21	20-21_01	4	06/01/2025
1	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	ARSON AND CRIMINAL DAMAGE	ARSON	Positive Outcomes	fy20-21	20-21_01	1	06/01/2025
2	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	DRUG OFFENCES	TRAFFICKING OF DRUGS	Offences	fy20-21	20-21_01	9	06/01/2025
3	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	PUBLIC ORDER OFFENCES	OTHER OFFENCES PUBLIC ORDER	Positive Outcomes	fy20-21	20-21_01	6	06/01/2025
4	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	POSSESSION OF WEAPONS	POSSESSION OF WEAPONS	Offences	fy20-21	20-21_01	11	06/01/2025
5	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	ROBBERY	ROBBERY OF BUSINESS PROPERTY	Positive Outcomes	fy20-21	20-21_01	1	06/01/2025
6	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	BURGLARY	BURGLARY - RESIDENTIAL	Positive Outcomes	fy20-21	20-21_01	6	06/01/2025
7	01/01/2021	Borough	Barking and Dagenham	Barking and Dagenham	E09000002	PUBLIC ORDER OFFENCES	PUBLIC FEAR ALARM OR DISTRESS	Offences	fy20-21	20-21_01	54	06/01/2025

(1)

```
df.tail()
```

	Month_Year	Area Type	Borough_SNT	Area name	Area code	Offence Group	Offence Subgroup	Measure	Financial Year	FY_FYIndex	Count	Refresh Date
787862	01/12/2024	Safer Neighbourhood Teams	Westminster West End	West End	E05013808	VEHICLE OFFENCES	INTERFERING WITH A MOTOR VEHICLE	Offences	fy24-25	24-25_01	6	06/01/2025
787863	01/12/2024	Safer Neighbourhood Teams	Westminster West End	West End	E05013808	VIOLENCE AGAINST THE PERSON	VIOLENCE WITHOUT INJURY	Offences	fy24-25	24-25_01	139	06/01/2025
787864	01/12/2024	Safer Neighbourhood Teams	Westminster West End	West End (NOT ACTIVE)	E05000649	THEFT	OTHER THEFT	Offences	fy24-25	24-25_01	4	06/01/2025
787865	01/12/2024	Safer Neighbourhood Teams	Westminster Westbourne	Westbourne	E05013809	PUBLIC ORDER OFFENCES	OTHER OFFENCES PUBLIC ORDER	Offences	fy24-25	24-25_01	1	06/01/2025
787866	01/12/2024	Safer Neighbourhood Teams	Westminster Westbourne	Westbourne	E05013809	VEHICLE OFFENCES	THEFT OR UNAUTH TAKING OF A MOTOR VEH	Offences	fy24-25	24-25_01	3	06/01/2025

Data Preprocessing (step 2)

to see type of crime

```
# Unique crime types in the dataset
unique_crime_types = df["Offence Group"].unique()
print("Unique Crime Types:\n", unique_crime_types)

# Unique subcategories of crimes
unique_crime_subtypes = df["Offence Subgroup"].unique()
print("\nUnique Crime Subcategories:\n", unique_crime_subtypes)
```

(2)

Data Preprocessing (step 2)

to see type of crime

```
# Unique crime types in the dataset
unique_crime_types = df["Offence Group"].unique()
print("Unique Crime Types:\n", unique_crime_types)

# Unique subcategories of crimes
unique_crime_subtypes = df["Offence Subgroup"].unique()
print("\nUnique Crime Subcategories:\n", unique_crime_subtypes)
```

```
Unique Crime Types:
 ['ARSON AND CRIMINAL DAMAGE' 'DRUG OFFENCES' 'PUBLIC ORDER OFFENCES'
 'POSSESSION OF WEAPONS' 'ROBBERY' 'BURGLARY'
 'MISCELLANEOUS CRIMES AGAINST SOCIETY' 'VEHICLE OFFENCES'
 'SEXUAL OFFENCES' 'THEFT' 'VIOLENCE AGAINST THE PERSON'
 'FRAUD AND FORGERY' 'NFIB FRAUD']

Unique Crime Subcategories:
 ['ARSON' 'TRAFFICKING OF DRUGS' 'OTHER OFFENCES PUBLIC ORDER'
 'POSSESSION OF WEAPONS' 'ROBBERY OF BUSINESS PROPERTY'
 'BURGLARY - RESIDENTIAL' 'PUBLIC FEAR ALARM OR DISTRESS'
 'POSSESSION OF DRUGS' 'MISC CRIMES AGAINST SOCIETY'
 'RACE OR RELIGIOUS AGG PUBLIC FEAR' 'THEFT FROM A VEHICLE'
 'CRIMINAL DAMAGE' 'OTHER SEXUAL OFFENCES'
 'BURGLARY - BUSINESS AND COMMUNITY' 'RAPE' 'BICYCLE THEFT'
 'AGGRAVATED VEHICLE TAKING' 'SHOPLIFTING'
 'INTERFERING WITH A MOTOR VEHICLE'
 'THEFT OR UNAUTH TAKING OF A MOTOR VEH' 'VIOLENCE WITHOUT INJURY'
 'VIOLENCE WITH INJURY' 'VIOLENT DISORDER' 'ROBBERY OF PERSONAL PROPERTY'
 'OTHER THEFT' 'THEFT FROM THE PERSON' 'HOMICIDE' 'FRAUD AND FORGERY'
 'NFIB']
```

(3)

```
#display basic information about the dataset
print("Initial data overview:")
print(df.info())
print(df.head(10))
```

```
Initial data overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 787867 entries, 0 to 787866
Data columns (total 12 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   Month_Year       787867 non-null  object
 1   Area Type        787867 non-null  object
 2   Borough_SNT      787867 non-null  object
 3   Area name        787449 non-null  object
 4   Area code        784800 non-null  object
 5   Offence Group    787867 non-null  object
 6   Offence Subgroup 787867 non-null  object
 7   Measure          787867 non-null  object
 8   Financial Year   787867 non-null  object
 9   FY_FYIndex       787867 non-null  object
 10  Count            787867 non-null  int64
 11  Refresh Date     787867 non-null  object
dtypes: int64(1), object(11)
memory usage: 72.1+ MB
None
   Month_Year Area Type          Borough_SNT            Area name  \
0  01/01/2021   Borough  Barking and Dagenham  Barking and Dagenham
1  01/01/2021   Borough  Barking and Dagenham  Barking and Dagenham
2  01/01/2021   Borough  Barking and Dagenham  Barking and Dagenham
3  01/01/2021   Borough  Barking and Dagenham  Barking and Dagenham
4  01/01/2021   Borough  Barking and Dagenham  Barking and Dagenham
5  01/01/2021   Borough  Barking and Dagenham  Barking and Dagenham
6  01/01/2021   Borough  Barking and Dagenham  Barking and Dagenham
7  01/01/2021   Borough  Barking and Dagenham  Barking and Dagenham
8  01/01/2021   Borough  Barking and Dagenham  Barking and Dagenham
9  01/01/2021   Borough  Barking and Dagenham  Barking and Dagenham

   Area code             Offence Group              Offence Subgroup  \
0  E09000002  ARSON AND CRIMINAL DAMAGE                        ARSON
```

(4)

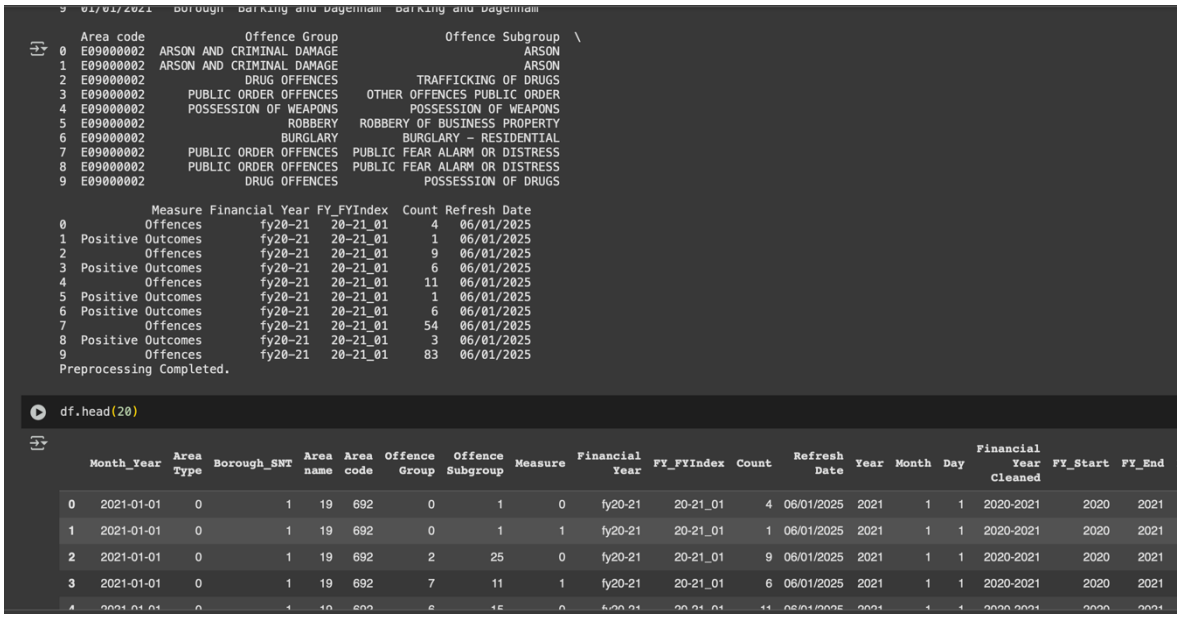
```python
#display basic information about the dataset
print("Initial data overview:")
print(df.info())
print(df.head(10))

#Handling missing values
df.dropna(inplace=True)

# Convert 'Month_Year' from "DD/MM/YYYY" to datetime format
df['Month_Year'] = pd.to_datetime(df['Month_Year'], format='%d/%m/%Y', errors='coerce')

# Extract Year ,Month and Day separately
df['Year'] = df['Month_Year'].dt.year
df['Month'] = df['Month_Year'].dt.month
df['Day'] = df['Month_Year'].dt.day

# Function to process Financial Year and FY_FYIndex
def process_fy_fyindex(fy_index):
    match = re.match(r'(\d{2}-\d{2})_(\d+)', str(fy_index))
    if match:
        fy_raw = match.group(1)  # Extract "20-21"
        index = int(match.group(2))  # Extract "01"

        # Convert "20-21" to "2020-2021"
        fy_match = re.match(r'(\d{2})-(\d{2})', fy_raw)
        if fy_match:
            fy_start = int("20" + fy_match.group(1))  # 2020
            fy_end = int("20" + fy_match.group(2))    # 2021
            return f"{fy_start}-{fy_end}", fy_start, fy_end, index
    return None, None, None, None

# Apply function to process FY_FYIndex column
df[['Financial Year Cleaned', 'FY_Start', 'FY_End', 'FY_Index']] = df['FY_FYIndex'].apply(
    lambda x: pd.Series(process_fy_fyindex(x))
)

# Convert categorical variables to category dtype
```

(5)

```python
)

# Convert categorical variables to category dtype
categorical_columns = ['Area Type', 'Borough_SNT', 'Area name', 'Area code',
                       'Offence Group', 'Offence Subgroup', 'Measure']
for col in categorical_columns:
    df[col] = df[col].astype('category')

# Encode categorical variables using label encoding (if needed)
from sklearn.preprocessing import LabelEncoder
label_encoders = {}
for col in categorical_columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le  # Store encoders for future use

# Ensure 'Count' is numeric
df['Count'] = pd.to_numeric(df['Count'], errors='coerce')

# Remove duplicates
df.drop_duplicates(inplace=True)

# Reset index
df.reset_index(drop=True, inplace=True)


print("Preprocessing Completed. ")
```

```
Initial data overview:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 787867 entries, 0 to 787866
Data columns (total 12 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Month_Year        787867 non-null  object
 1   Area Type         787867 non-null  object
 2   Borough_SNT       787867 non-null  object
```

(6)

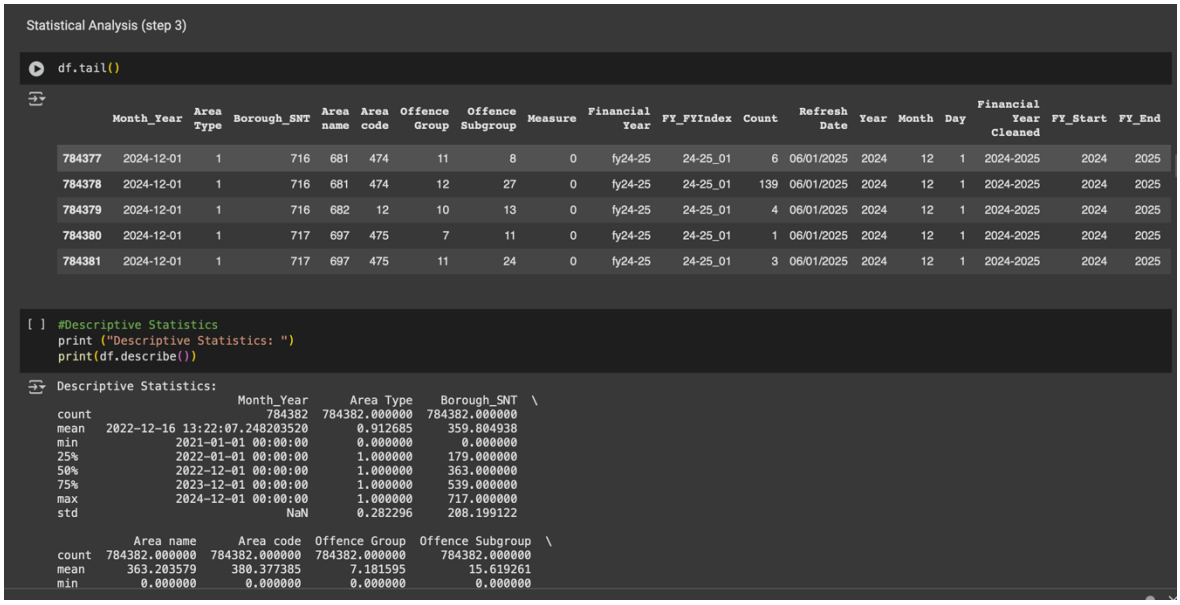


```
9  01/01/2021  Borough  Barking and Dagenham  Barking and Dagenham

    Area code             Offence Group             Offence Subgroup  \
0  E09000002  ARSON AND CRIMINAL DAMAGE                        ARSON
1  E09000002  ARSON AND CRIMINAL DAMAGE                        ARSON
2  E09000002            DRUG OFFENCES          TRAFFICKING OF DRUGS
3  E09000002     PUBLIC ORDER OFFENCES   OTHER OFFENCES PUBLIC ORDER
4  E09000002     POSSESSION OF WEAPONS          POSSESSION OF WEAPONS
5  E09000002                  ROBBERY  ROBBERY OF BUSINESS PROPERTY
6  E09000002                  BURGLARY         BURGLARY - RESIDENTIAL
7  E09000002     PUBLIC ORDER OFFENCES  PUBLIC FEAR ALARM OR DISTRESS
8  E09000002     PUBLIC ORDER OFFENCES  PUBLIC FEAR ALARM OR DISTRESS
9  E09000002            DRUG OFFENCES            POSSESSION OF DRUGS

            Measure Financial Year FY_FYIndex  Count Refresh Date
0          Offences        fy20-21   20-21_01      4   06/01/2025
1  Positive Outcomes        fy20-21   20-21_01      1   06/01/2025
2          Offences        fy20-21   20-21_01      9   06/01/2025
3  Positive Outcomes        fy20-21   20-21_01      6   06/01/2025
4          Offences        fy20-21   20-21_01     11   06/01/2025
5  Positive Outcomes        fy20-21   20-21_01      1   06/01/2025
6  Positive Outcomes        fy20-21   20-21_01      6   06/01/2025
7          Offences        fy20-21   20-21_01     54   06/01/2025
8  Positive Outcomes        fy20-21   20-21_01      3   06/01/2025
9          Offences        fy20-21   20-21_01     83   06/01/2025
Preprocessing Completed.
```

```
df.head(20)
```

	Month_Year	Area Type	Borough_SNT	Area name	Area code	Offence Group	Offence Subgroup	Measure	Financial Year	FY_FYIndex	Count	Refresh Date	Year	Month	Day	Financial Year Cleaned	FY_Start	FY_End
0	2021-01-01	0	1	19	692	0	1	0	fy20-21	20-21_01	4	06/01/2025	2021	1	1	2020-2021	2020	2021
1	2021-01-01	0	1	19	692	0	1	1	fy20-21	20-21_01	1	06/01/2025	2021	1	1	2020-2021	2020	2021
2	2021-01-01	0	1	19	692	2	25	0	fy20-21	20-21_01	9	06/01/2025	2021	1	1	2020-2021	2020	2021
3	2021-01-01	0	1	19	692	7	11	1	fy20-21	20-21_01	6	06/01/2025	2021	1	1	2020-2021	2020	2021

(7)



Statistical Analysis (step 3)

```
df.tail()
```

	Month_Year	Area Type	Borough_SNT	Area name	Area code	Offence Group	Offence Subgroup	Measure	Financial Year	FY_FYIndex	Count	Refresh Date	Year	Month	Day	Financial Year Cleaned	FY_Start	FY_End
784377	2024-12-01	1	716	681	474	11	8	0	fy24-25	24-25_01	6	06/01/2025	2024	12	1	2024-2025	2024	2025
784378	2024-12-01	1	716	681	474	12	27	0	fy24-25	24-25_01	139	06/01/2025	2024	12	1	2024-2025	2024	2025
784379	2024-12-01	1	716	682	12	10	13	0	fy24-25	24-25_01	4	06/01/2025	2024	12	1	2024-2025	2024	2025
784380	2024-12-01	1	717	697	475	7	11	0	fy24-25	24-25_01	1	06/01/2025	2024	12	1	2024-2025	2024	2025
784381	2024-12-01	1	717	697	475	11	24	0	fy24-25	24-25_01	3	06/01/2025	2024	12	1	2024-2025	2024	2025

```
#Descriptive Statistics
print ("Descriptive Statistics: ")
print(df.describe())
```

```
Descriptive Statistics:
                      Month_Year      Area Type    Borough_SNT  \
count                     784382  784382.000000  784382.000000
mean   2022-12-16 13:22:07.248203520       0.912685     359.804938
min              2021-01-01 00:00:00       0.000000       0.000000
25%              2022-01-01 00:00:00       1.000000     179.000000
50%              2022-12-01 00:00:00       1.000000     363.000000
75%              2023-12-01 00:00:00       1.000000     539.000000
max              2024-12-01 00:00:00       1.000000     717.000000
std                              NaN       0.282296     208.199122

          Area name      Area code  Offence Group  Offence Subgroup  \
count  784382.000000  784382.000000  784382.000000     784382.000000
mean      363.203579     380.377385       7.181595         15.619261
min         0.000000       0.000000       0.000000          0.000000
```

(8)

```
50%          0.000000       3.000000    2022.000000       7.000000       1.0
75%          0.000000       7.000000    2023.000000      10.000000       1.0
max          1.000000    4054.000000    2024.000000      12.000000       1.0
std          0.420160      38.703030       1.116032       3.450347       0.0

            FY_Start         FY_End  FY_Index
count  784382.000000  784382.000000  784382.0
mean     2022.249511    2023.249511       1.0
min      2020.000000    2021.000000       1.0
25%      2021.000000    2022.000000       1.0
50%      2022.000000    2023.000000       1.0
75%      2023.000000    2024.000000       1.0
max      2024.000000    2025.000000       1.0
std         1.197838       1.197838       0.0
```

```
#check for unique values in categorical variables
print("\nUnique values in categorical variable:")
for col in ['Offence Group', 'Offence Subgroup', 'Area Type', 'Borough_SNT']:
  print(f"{col} : {df}")
```

```
Unique values in categorical variable:
Offence Group :        Month_Year  Area Type  Borough_SNT  Area name  Area code  \
0       2021-01-01           0            1         19        692
1       2021-01-01           0            1         19        692
2       2021-01-01           0            1         19        692
```

(9)

```
[784382 rows x 19 columns]
Borough_SNT :        Month_Year  Area Type  Borough_SNT  Area name  Area code  \
0       2021-01-01           0            1         19        692
1       2021-01-01           0            1         19        692
2       2021-01-01           0            1         19        692
3       2021-01-01           0            1         19        692
4       2021-01-01           0            1         19        692
...            ...         ...          ...        ...        ...
784377 2024-12-01           1          716        681        474
784378 2024-12-01           1          716        681        474
784379 2024-12-01           1          716        682         12
784380 2024-12-01           1          717        697        475
784381 2024-12-01           1          717        697        475

        Offence Group  Offence Subgroup  Measure Financial Year FY_FYIndex  \
0                   0                 1        0        fy20-21   20-21_01
1                   0                 1        1        fy20-21   20-21_01
2                   2                25        0        fy20-21   20-21_01
3                   7                11        1        fy20-21   20-21_01
4                   6                15        0        fy20-21   20-21_01
...               ...               ...      ...            ...       ...
784377             11                 8        0        fy24-25   24-25_01
784378             12                27        0        fy24-25   24-25_01
784379             10                13        0        fy24-25   24-25_01
784380              7                11        0        fy24-25   24-25_01
784381             11                24        0        fy24-25   24-25_01

        Count Refresh Date  Year  Month  Day Financial Year Cleaned  FY_Start  \
0           4   06/01/2025  2021      1    1              2020-2021      2020
1           1   06/01/2025  2021      1    1              2020-2021      2020
2           9   06/01/2025  2021      1    1              2020-2021      2020
3           6   06/01/2025  2021      1    1              2020-2021      2020
4          11   06/01/2025  2021      1    1              2020-2021      2020
...       ...          ...   ...    ...  ...                    ...       ...
784377      6   06/01/2025  2024     12    1              2024-2025      2024
784378    139   06/01/2025  2024     12    1              2024-2025      2024
784379      4   06/01/2025  2024     12    1              2024-2025      2024
784380      1   06/01/2025  2024     12    1              2024-2025      2024
784381      3   06/01/2025  2024     12    1              2024-2025      2024

        FY_End  FY_Index
0         2021         1
```

(10)

```
[ ] monthly_avg_crime = df.groupby('Day')['Count'].mean()
    print(monthly_avg_crime)

Day
1    9.588889
Name: Count, dtype: float64

    import seaborn as sns
    import matplotlib.pyplot as plt

    #Aggregate crime count per year, month , and day
    yearly_crime = df.groupby('Year')['Count'].sum()
    monthly_crime = df.groupby('Month')['Count'].sum()
    daily_crime = df.groupby('Day')['Count'].sum()

    #statistical summary of crime data
    print("\nCrime Statistical Summary:")
    print(yearly_crime.describe())
    print(monthly_crime.describe())
    print(daily_crime.describe())

    #set up a figure with three boxplots
    fig, axes = plt.subplots(1, 3, figsize=(18,6))

    # Boxplot for yearly crime distribution
    sns.boxplot(y=yearly_crime, color='skyblue', ax=axes[0])
    axes[0].set_title("Crime Distribution Per Year")
    axes[0].set_ylabel("Incident counts per year")
    axes[0].set_xlabel("Years")

    # Boxplot for monthly crime distribution
    sns.boxplot(y=monthly_crime, color='purple', ax=axes[1])
    axes[1].set_title("Crime Distribution Per Month")
    axes[1].set_ylabel("Incident counts per month")
    axes[1].set_xlabel("Months")
```

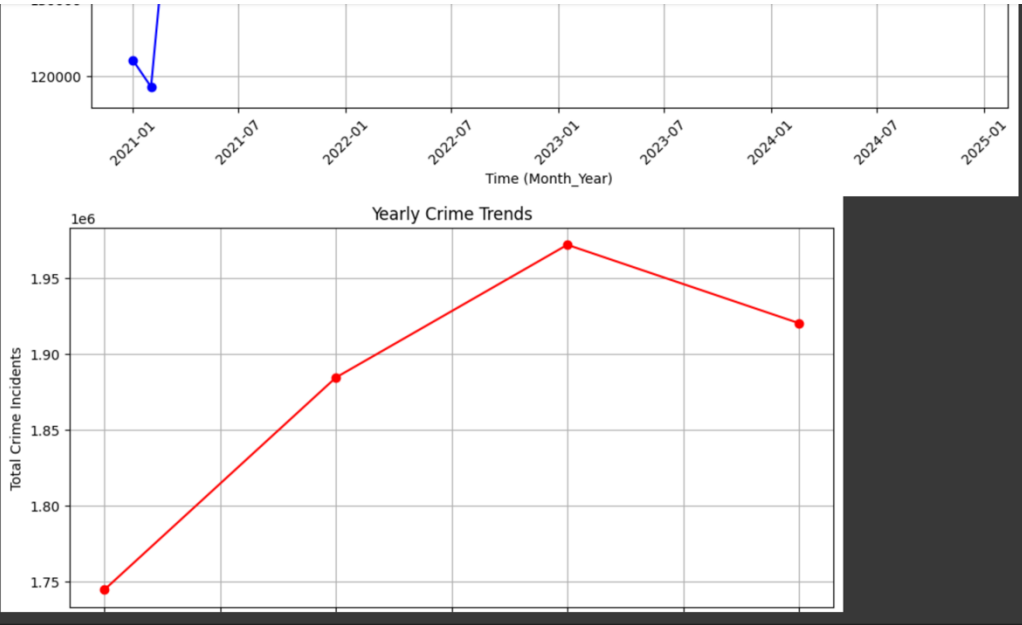
(11)

```
50%      1.902346e+06
75%      1.933138e+06
max      1.971779e+06
Name: Count, dtype: float64
count        12.000000
mean     626779.333333
std       29615.154660
min      574768.000000
25%      613339.500000
50%      628799.000000
75%      644732.750000
max      670238.000000
Name: Count, dtype: float64
count          1.0
mean     7521352.0
std            NaN
min      7521352.0
25%      7521352.0
50%      7521352.0
75%      7521352.0
max      7521352.0
Name: Count, dtype: float64
```

(12)

```python
#group month_year to get overall trends
monthly_trends = df.groupby('Month_Year')['Count'].sum()

#plot the monthly trends
plt.figure(figsize=(12,6))
plt.plot(monthly_trends.index, monthly_trends.values, marker='o', linestyle='-', color='b', label='Monthly Crime Trends')
plt.title("Crime Trends Over Time")
plt.xlabel("Time (Month_Year)")
plt.ylabel("Total Crime Incidents")
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)
plt.show()

#group by year to see yearly trends
yearly_trends = df.groupby('Year')['Count'].sum()

#plot yearly trends
plt.figure(figsize=(10,5))
# Changed the following line to remove keyword arguments 'x' and 'y'
plt.plot(yearly_trends.index, yearly_trends.values, marker='o', color='r')
plt.title("Yearly Crime Trends")
plt.xlabel("Year")
plt.ylabel("Total Crime Incidents")
plt.grid(True)
plt.show()
```

(13)




```python
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

(14)

```python
import statsmodels.api as sm
import statsmodels.formula.api as smf
from scipy.stats import chi2_contingency, ttest_ind, f_oneway, pearsonr



#  **Chi-Square Test (Crime Type vs Borough)**
contingency_table = pd.crosstab(df['Borough_SNT'], df['Offence Group'])
chi2, p, dof, expected = chi2_contingency(contingency_table)

print("\nChi-Square Test Results (Crime Type vs Borough):")
print(f"Chi-Square Statistic: {chi2:.3f}, p-value: {p:.3f}")

if p < 0.05:
    print("There is a significant relationship between crime type and borough.")
else:
    print("No significant relationship found between crime type and borough.")

#  **T-Test (Comparing Crime Count in Two Boroughs)**
boroughs = df['Borough_SNT'].unique()[:2]  # Select first two boroughs for comparison
df_b1 = df[df['Borough_SNT'] == boroughs[0]]['Count']
df_b2 = df[df['Borough_SNT'] == boroughs[1]]['Count']

t_stat, p_value = ttest_ind(df_b1, df_b2, equal_var=False)

print("\nT-Test Results Between Two Boroughs:")
print(f"T-Statistic: {t_stat:.3f}, p-value: {p_value:.3f}")

if p_value < 0.05:
    print(f"Significant difference in crime counts between {boroughs[0]} and {boroughs[1]}.")
else:
    print(f"No significant difference in crime counts between {boroughs[0]} and {boroughs[1]}.")

#  **ANOVA Test (Crime Rates Across Multiple Boroughs)**
borough_groups = [df[df['Borough_SNT'] == borough]['Count'] for borough in df['Borough_SNT'].unique()[:5]]
anova_stat, anova_p = f_oneway(*borough_groups)
```

(15)

```python
[ ]
if anova_p < 0.05:
    print("Crime rates significantly vary across boroughs.")
else:
    print("No significant variation in crime rates across boroughs.")
```

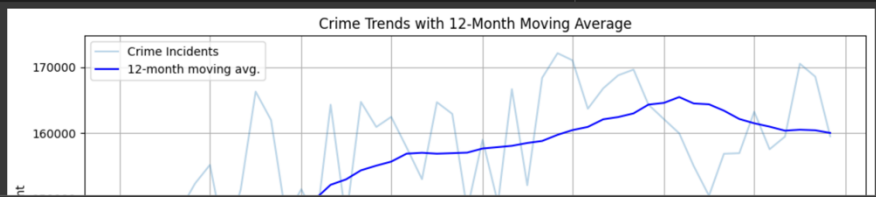
```
Chi-Square Test Results (Crime Type vs Borough):
Chi-Square Statistic: 21546.042, p-value: 0.000
There is a significant relationship between crime type and borough.

T-Test Results Between Two Boroughs:
T-Statistic: -4.879, p-value: 0.000
Significant difference in crime counts between 1 and 21.

ANOVA Test Results (Crime Rates Across Boroughs):
F-Statistic: 32.863, p-value: 0.000
Crime rates significantly vary across boroughs.
```

Crime Trends with 12-Month Moving **Average**

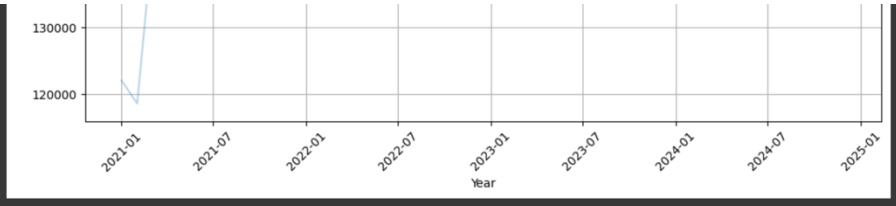
```python
# Convert 'Month_Year' to datetime format
df['Month_Year'] = pd.to_datetime(df['Month_Year'], format='%d/%m/%Y')

# Aggregate crime counts per month
monthly_trend = df.groupby('Month_Year')['Count'].sum().reset_index()

# Calculate 12-month moving average
monthly_trend['Moving_Avg'] = monthly_trend['Count'].rolling(window=12).mean()

# Plot Time Series Chart
plt.figure(figsize=(12, 6))
sns.lineplot(x=monthly_trend['Month_Year'], y=monthly_trend['Count'], label="Crime Incidents", alpha=0.3)
sns.lineplot(x=monthly_trend['Month_Year'], y=monthly_trend['Moving_Avg'], label="12-month moving avg.", color='blue')
```

(16)

```
df['Month_Year'] = pd.to_datetime(df['Month_Year'], format='%d/%m/%Y')

# Aggregate crime counts per month
monthly_trend = df.groupby('Month_Year')['Count'].sum().reset_index()

# Calculate 12-month moving average
monthly_trend['Moving_Avg'] = monthly_trend['Count'].rolling(window=12).mean()

# Plot Time Series Chart
plt.figure(figsize=(12, 6))
sns.lineplot(x=monthly_trend['Month_Year'], y=monthly_trend['Count'], label="Crime Incidents", alpha=0.3)
sns.lineplot(x=monthly_trend['Month_Year'], y=monthly_trend['Moving_Avg'], label="12-month moving avg.", color='blue')

# Formatting
plt.title("Crime Trends with 12-Month Moving Average")
plt.xlabel("Year")
plt.ylabel("Incidents Count")
plt.xticks(rotation=45)
plt.legend()
plt.grid(True)

# Show Plot
plt.show()
```



(17)

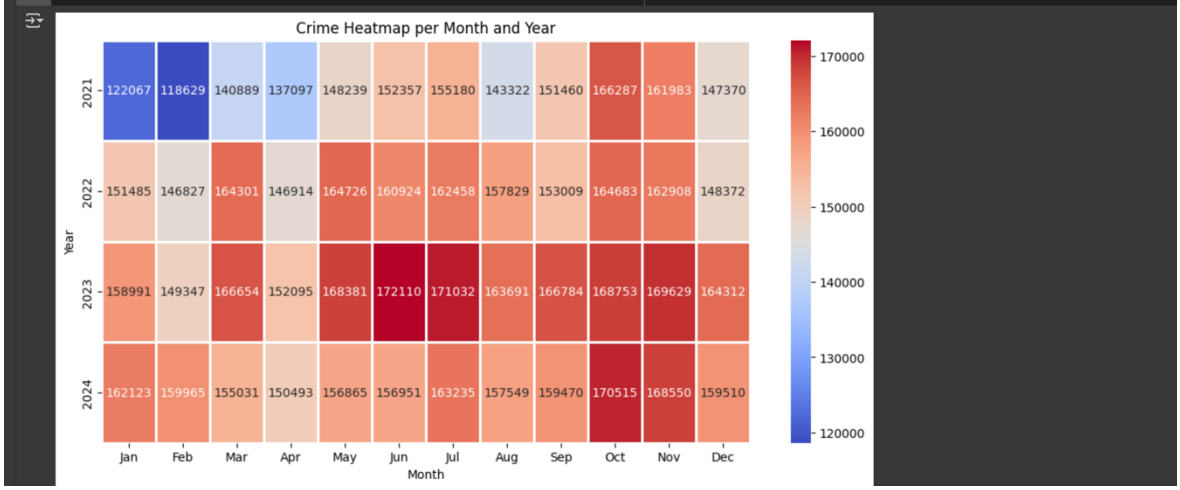


heatmap

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#extract year , month, day , hour, and weekday
df['Day'] = df['Month_Year'].dt.day
df['Month'] = df['Month_Year'].dt.month
df['Year'] = df['Month_Year'].dt.year
df['Weekday'] = df['Month_Year'].dt.day_name()
df['Hour'] = np.random.randint(0, 24, df.shape[0])  # Random hour values
df['Day'] = np.random.randint(1, 31, df.shape[0]) # Random day value

crime_per_month_year = df.pivot_table(index="Year", columns="Month", values="Count", aggfunc="sum")

plt.figure(figsize=(12,6))
sns.heatmap(crime_per_month_year, cmap="coolwarm", annot=True, fmt=".0f", linewidths=0.9)
plt.title("Crime Heatmap per Month and Year")
plt.xlabel("Month")
plt.ylabel("Year")
```

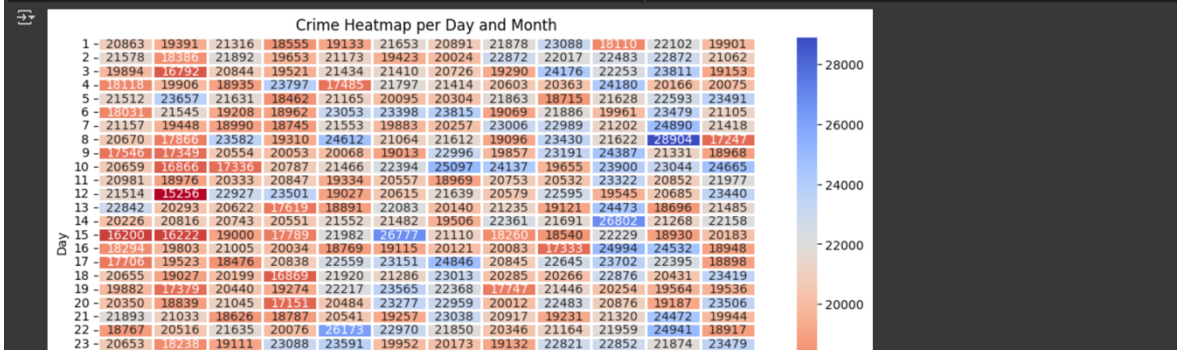
(18)

```
sns.heatmap(crime_per_month_year, cmap="coolwarm", annot=True, fmt=".0f", linewidths=0.9)
plt.title("Crime Heatmap per Month and Year")
plt.xlabel("Month")
plt.ylabel("Year")
plt.xticks(ticks=np.arange(12) + 0.5, labels=["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"])
plt.show()
```



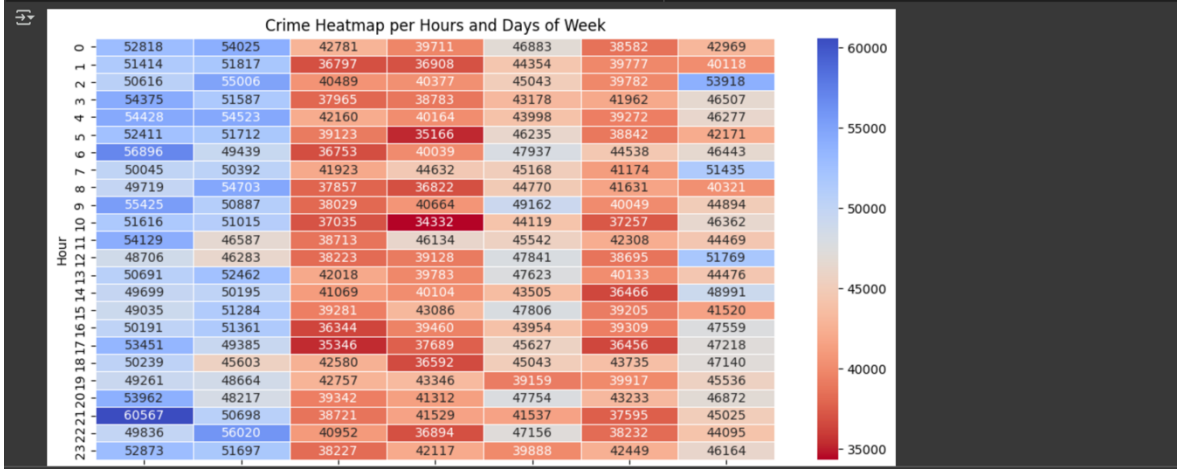
(19)

```
#  (a) Crime Heatmap per Day and Month
crime_per_day_month = df.pivot_table(index="Day", columns="Month", values="Count", aggfunc="sum")

plt.figure(figsize=(12, 6))
sns.heatmap(crime_per_day_month, cmap="coolwarm_r", annot=True, fmt=".0f", linewidths=0.9)
plt.title("Crime Heatmap per Day and Month")
plt.xlabel("Month")
plt.ylabel("Day")
plt.xticks(ticks=np.arange(12) + 0.5, labels=["Jan", "Feb", "Mar", "Apr", "May", "Jun",
                                              "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"])
plt.show()
```



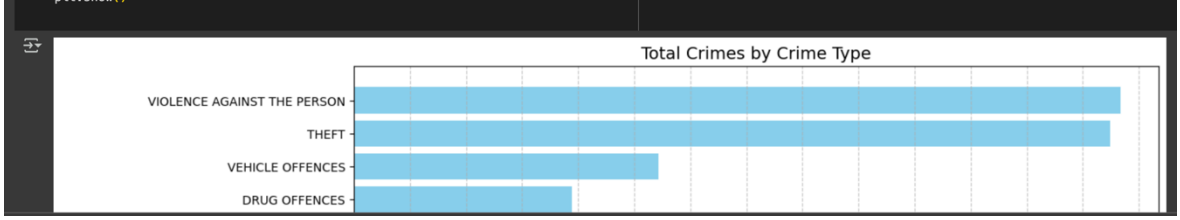
(20)

```
[ ] # (b) Crime Heatmap per Hour and Days of the Week
    crime_per_hour_weekday = df.pivot_table(index="Hour", columns="Weekday", values="Count", aggfunc="sum")

    plt.figure(figsize=(12, 6))
    sns.heatmap(crime_per_hour_weekday, cmap="coolwarm_r", annot=True, fmt=".0f", linewidths=0.5)
    plt.title("Crime Heatmap per Hours and Days of Week")
    plt.xlabel("Weekday")
    plt.ylabel("Hour")
    plt.show()
```



(21)

```
    import matplotlib.pyplot as plt

    # Decode crime types from numerical values to actual category names
    df["Offence Group"] = label_encoders["Offence Group"].inverse_transform(df["Offence Group"])

    # Aggregate the total number of crimes per crime type
    crime_counts = df.groupby("Offence Group")["Count"].sum().sort_values(ascending=True)

    # Plot the horizontal bar chart
    plt.figure(figsize=(12, 7))
    plt.barh(crime_counts.index, crime_counts.values, color="skyblue")

    # Labels and title
    plt.xlabel("Number of Crimes", fontsize=12)
    plt.ylabel("Crime Type", fontsize=12)
    plt.title("Total Crimes by Crime Type", fontsize=14)
    plt.grid(axis="x", linestyle="--", alpha=0.7)

    #set values of x axis in the intervals of 50000
    max_value = crime_counts.max()
    plt.xticks(np.arange(0, max_value + 150000, 150000))
    # Show the chart
    plt.show()
```

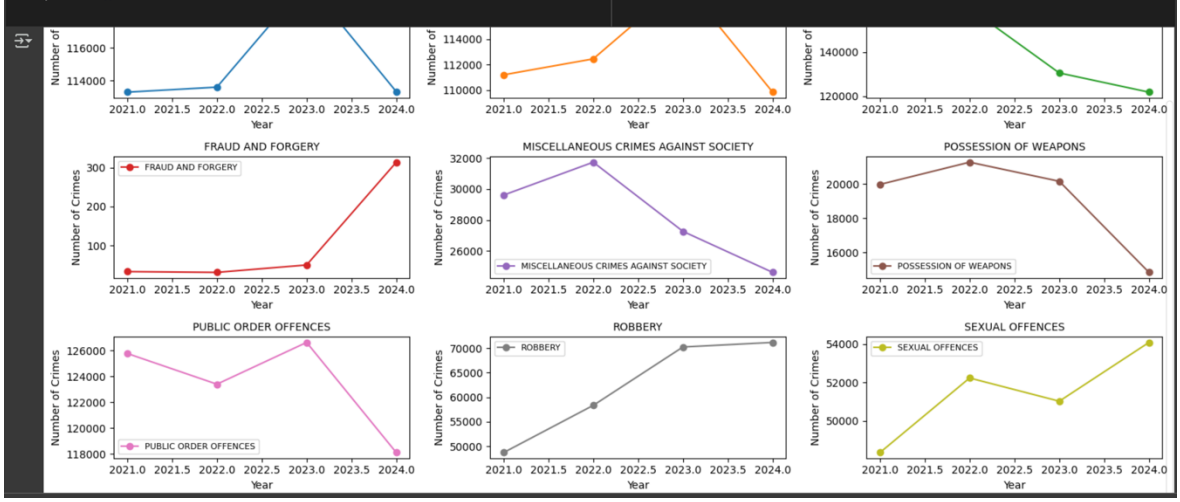


(22)

```python
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Group data by Year and Crime Type, then sum crime counts
crime_trends = df.groupby(["Year", "Offence Group"])["Count"].sum().reset_index()

# Get unique crime types
crime_types = crime_trends["Offence Group"].unique()

# Generate a color palette with as many colors as crime types
colors = sns.color_palette("tab10", len(crime_types))  # "tab10" gives distinct colors

# Calculate the number of rows and columns needed for subplots
num_crimes = len(crime_types)
num_cols = 3
num_rows = (num_crimes + num_cols - 1) // num_cols  # Calculate rows dynamically

# Set up the grid for subplots with enough space for all crime types
fig, axes = plt.subplots(nrows=num_rows, ncols=num_cols, figsize=(15, 12))
axes = axes.flatten()  # Flatten for easy iteration

# Loop through crime types and plot each trend with a unique color
for i, (crime, color) in enumerate(zip(crime_types, colors)):
    ax = axes[i]
    crime_data = crime_trends[crime_trends["Offence Group"] == crime]

    ax.plot(crime_data["Year"], crime_data["Count"], marker='o', linestyle='-', color=color, label=crime)
    ax.set_title(crime, fontsize=10)
    ax.set_xlabel("Year")
    ax.set_ylabel("Number of Crimes")
    ax.legend(fontsize=8)

# Remove empty subplots if there are fewer crime types than subplot slots
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])  # Remove extra empty subplots
```

(23)

```python
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])  # Remove extra empty subplots

# Adjust layout to prevent overlap
plt.tight_layout()
plt.show()
```



(24)

Geospatial Analysis (step 5)

```
[ ] import pandas as pd

    # Load crime data (Ensure it has 'Borough' and 'Crime Count')
    df = pd.read_csv("/content/2025.csv")

    # Group by borough to get total crime counts
    crime_by_borough = df.groupby("Borough_SNT")["Count"].sum().reset_index()

    # Preview data
    print(crime_by_borough.head())
```

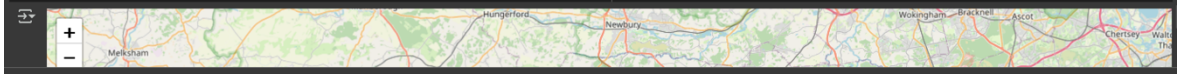
```
                            Borough_SNT  Count
0                      Aviation Security  10638
1                   Barking and Dagenham  88975
2             Barking and Dagenham Abbey   7023
3            Barking and Dagenham Alibon   3424
4  Barking and Dagenham Barking Riverside   4598
```

```
import geopandas as gpd

geojson_path = "/content/london.geojson"

gdf = gpd.read_file(geojson_path)

print(gdf.head())
```

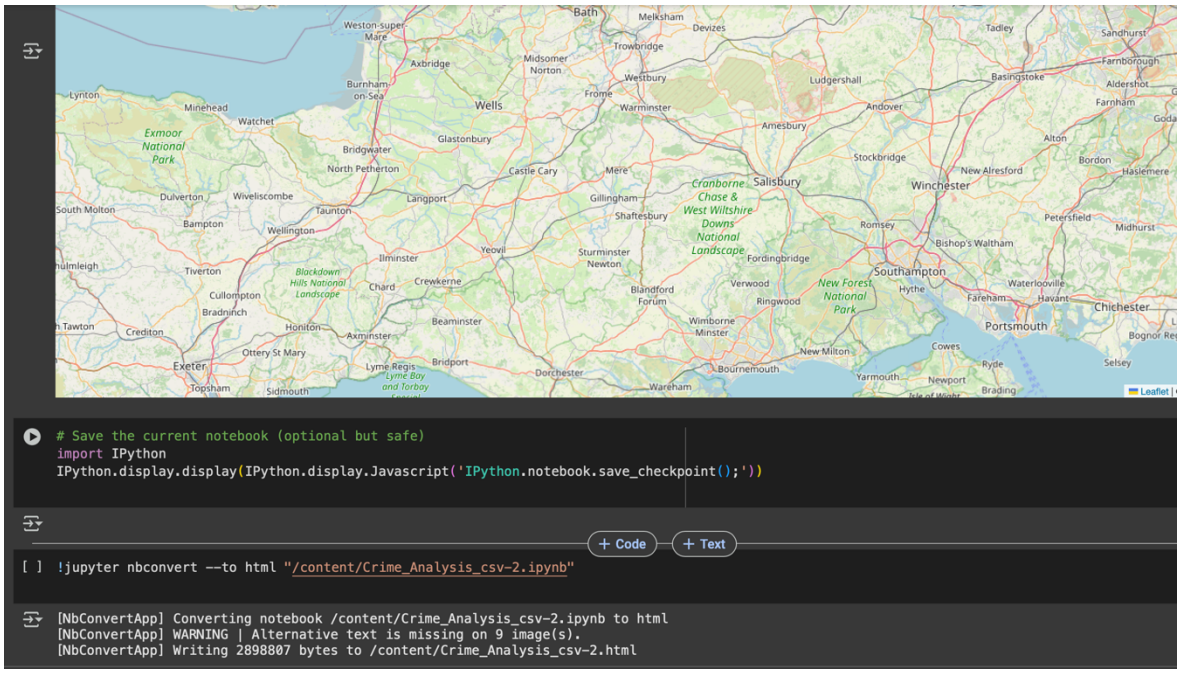
```
       type     id                                               tags  \
0  relation  58447  { "ISO3166-2": "GB-ENG", "admin_level": "4", "...

  relations meta                                           geometry
0       [ ]  { }  POLYGON ((-3.08279 52.77756, -3.08333 52.77768...
```

(25)

```
import pandas as pd
import folium
from folium.plugins import MarkerCluster

# Load your original crime data
df = pd.read_csv("/content/2025.csv")

# Drop rows with missing latitude or longitude
df = df.dropna(subset=['lat', 'lon'])  # Remove rows where 'lat' or 'lon' is NaN

# Create a base map (centered on London or your area)
m = folium.Map(location=[51.5074, -0.1278], zoom_start=10)

# Add marker clustering
marker_cluster = MarkerCluster().add_to(m)

# Plot each crime point
for idx, row in df.iterrows():
    folium.CircleMarker(
        location=[row['lat'], row['lon']],
        radius=5 + row['Count'] * 0.05,  # Radius scaled by crime count
        color='blue',
        fill=True,
        fill_color='blue',
        fill_opacity=0.6,
        popup=f"Area: {row['Area name']}<br>Crimes: {row['Count']}"
    ).add_to(marker_cluster)

# Save the map to an HTML file
m.save("crime_choropleth.html")
m

m
```



(26)




```
# Save the current notebook (optional but safe)
import IPython
IPython.display.display(IPython.display.Javascript('IPython.notebook.save_checkpoint();'))
```

+ Code + Text

```
[ ] !jupyter nbconvert --to html "/content/Crime_Analysis_csv-2.ipynb"
```

```
[NbConvertApp] Converting notebook /content/Crime_Analysis_csv-2.ipynb to html
[NbConvertApp] WARNING | Alternative text is missing on 9 image(s).
[NbConvertApp] Writing 2898807 bytes to /content/Crime_Analysis_csv-2.html
```

(27)

Prediction

```
import pandas as pd
import numpy as np
import re
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score
```

```
# Load data
file_path = '/content/crime_2025.csv'
df = pd.read_csv(file_path)

# Drop rows with missing values initially
df.dropna(inplace=True)
```

```
[ ] # Convert Month_Year to datetime and extract components
df['Month_Year'] = pd.to_datetime(df['Month_Year'], format='%d/%m/%Y', errors='coerce')
df['Year'] = df['Month_Year'].dt.year
df['Month'] = df['Month_Year'].dt.month
df['Day'] = df['Month_Year'].dt.day

# Function to process Financial Year and FY Index
def process_fy_fyindex(fy_index):
    match = re.match(r'(\d{2}-\d{2})_(\d+)', str(fy_index))
    if match:
        fy_raw = match.group(1)
        index = int(match.group(2))
        fy_match = re.match(r'(\d{2})-(\d{2})', fy_raw)
        if fy_match:
            fy_start = int("20" + fy_match.group(1))
            fy_end = int("20" + fy_match.group(2))
```

(28)

```
[ ] # Apply FY processing
    df[['Financial Year Cleaned', 'FY_Start', 'FY_End', 'FY_Index']] = df['FY_FYIndex'].apply(
        lambda x: pd.Series(process_fy_fyindex(x))
    )

    # Convert categorical columns
    categorical_columns = ['Area Type', 'Borough_SNT', 'Area name', 'Area code',
                           'Offence Group', 'Offence Subgroup', 'Measure']
    for col in categorical_columns:
        df[col] = df[col].astype(str)  # Ensure type is consistent for encoding

    # Store encoders for decoding later
    label_encoders = {}
    for col in categorical_columns:
        le = LabelEncoder()
        df[col] = le.fit_transform(df[col])
        label_encoders[col] = le

    # Ensure Count is numeric
    df['Count'] = pd.to_numeric(df['Count'], errors='coerce')
    df.dropna(subset=['Count'], inplace=True)  # Drop if Count couldn't be converted
```

```
    # Remove duplicates and reset index
    df.drop_duplicates(inplace=True)
    df.reset_index(drop=True, inplace=True)

    # Drop unnecessary columns, including 'FY_FYIndex'
    drop_columns = ["Month_Year", "Refresh Date", "Financial Year Cleaned", "FY_Start", "FY_End", "FY_Index", "FY_FYIndex"]
    df.drop(columns=drop_columns, inplace=True, errors='ignore')

    # Create date-based features
    df["Date"] = pd.to_datetime(df[["Year", "Month", "Day"]])
    df["Year_Month"] = df["Year"] + df["Month"] / 12.0

    # Define features and target
    X = df.drop(columns=["Count", "Date"])
```

(29)

```
[ ] # Explicitly select only numerical features for scaling
    X = X.select_dtypes(include=['number'])

    # Scale features
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # Train-test split (no shuffle to simulate time series)
    X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42, shuffle=False)
```

```
[ ] # --- KNN MODEL ---
    knn = KNeighborsRegressor(n_neighbors=5)
    knn.fit(X_train, y_train)
    y_pred_knn = knn.predict(X_test)
    print("KNN MAE:", mean_absolute_error(y_test, y_pred_knn))
    print("KNN R² Score:", r2_score(y_test, y_pred_knn))
```

```
KNN MAE: 5.813739426429623
KNN R² Score: 0.46969322944831593
```

```
[ ] # --- RANDOM FOREST MODEL ---
    rf = RandomForestRegressor(n_estimators=100, random_state=42)
    rf.fit(X_train, y_train)
    y_pred_rf = rf.predict(X_test)
    print("Random Forest R² Score:", r2_score(y_test, y_pred_rf))
```

```
Random Forest R² Score: 0.9470105863706114
```

```
    # --- FUTURE PREDICTION FOR 2025 ---
    latest_year = df["Year"].max()
    future_data = df[df["Year"] == latest_year].copy()
    future_data["Year"] = 2025
```

(30)

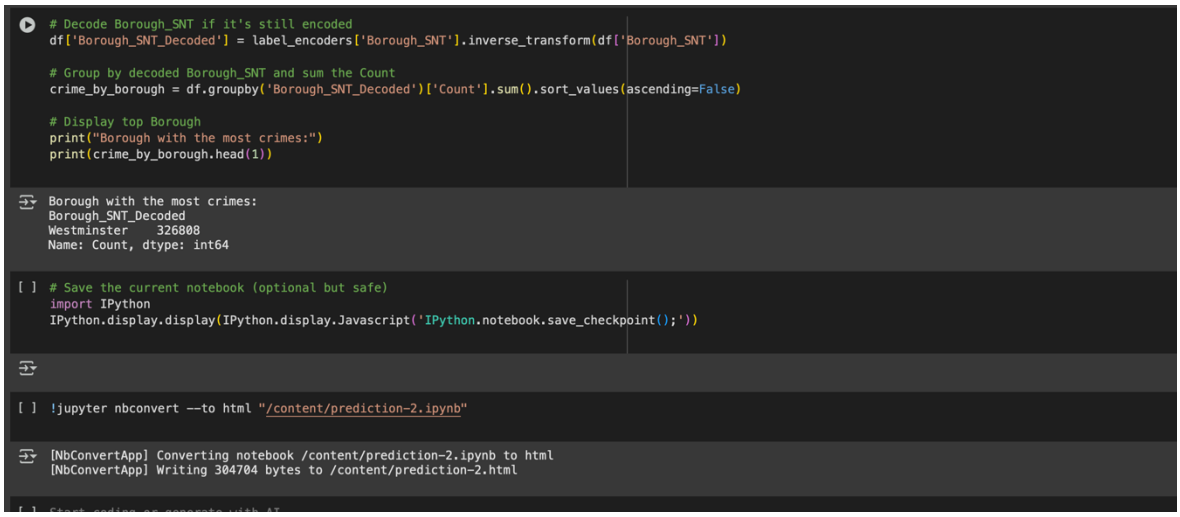
```python
# --- FUTURE PREDICTION FOR 2025 ---
latest_year = df["Year"].max()
future_data = df[df["Year"] == latest_year].copy()
future_data["Year"] = 2025

# Select only numerical features for future data as well
future_X = future_data.drop(columns=["Count", "Date"]).select_dtypes(include=['number'])

future_X = scaler.transform(future_X)

# Predict using Random Forest
future_predictions = rf.predict(future_X)
future_data["Predicted Crime Count"] = future_predictions

# Decode categorical columns
decoded_future_data = future_data.copy()
for col, le in label_encoders.items():
    if col in decoded_future_data.columns:
        decoded_future_data[col] = le.inverse_transform(decoded_future_data[col].astype(int))

# Add Year-Month
decoded_future_data["Year-Month"] = decoded_future_data["Year"].astype(str) + '-' + decoded_future_data["Month"].astype(str).str.zfill(2)

# Output columns
output_columns = ['Year', 'Month', 'Year-Month', 'Area name', 'Area Type',
                  'Borough_SNT', 'Area code', 'Offence Group', 'Predicted Crime Count']

# Print and save
print("\n--- Predicted Crime Data for 2025 ---")
print(decoded_future_data[output_columns].to_string(index=False))

# Save to CSV
output_filename = "/content/future_crime_predictions_2025.csv"
decoded_future_data[output_columns].to_csv(output_filename, index=False)

# Enable download in Colab
from google.colab import files
```

(31)

```python
print(decoded_future_data[output_columns].to_string(index=False))

# Save to CSV
output_filename = "/content/future_crime_predictions_2025.csv"
decoded_future_data[output_columns].to_csv(output_filename, index=False)

# Enable download in Colab
from google.colab import files
files.download(output_filename)
```

```
Streaming output truncated to the last 5000 lines.
2025  11  2025-11  Crystal Palace & Anerley Safer Neighbourhood Teams  Bromley Crystal Palace & Anerley  E05013995
2025  11  2025-11  Crystal Palace & Anerley Safer Neighbourhood Teams  Bromley Crystal Palace & Anerley  E05013995
2025  11  2025-11  Darwin Safer Neighbourhood Teams  Bromley Darwin  E05013996
2025  11  2025-11  Darwin Safer Neighbourhood Teams  Bromley Darwin  E05013996
2025  11  2025-11  Darwin Safer Neighbourhood Teams  Bromley Darwin  E05013996
2025  11  2025-11  Farnborough & Crofton Safer Neighbourhood Teams  Bromley Farnborough & Crofton  E05013997
2025  11  2025-11  Farnborough & Crofton Safer Neighbourhood Teams  Bromley Farnborough & Crofton  E05013997
2025  11  2025-11  Farnborough & Crofton Safer Neighbourhood Teams  Bromley Farnborough & Crofton  E05013997
2025  11  2025-11  Hayes & Coney Hall Safer Neighbourhood Teams  Bromley Hayes & Coney Hall  E05013998
2025  11  2025-11  Hayes & Coney Hall Safer Neighbourhood Teams  Bromley Hayes & Coney Hall  E05013998
2025  11  2025-11  Kelsey & Eden Park Safer Neighbourhood Teams  Bromley Kelsey & Eden Park  E05013999
2025  11  2025-11  Kelsey & Eden Park Safer Neighbourhood Teams  Bromley Kelsey & Eden Park  E05013999
2025  11  2025-11  Kelsey & Eden Park Safer Neighbourhood Teams  Bromley Kelsey & Eden Park  E05013999
2025  11  2025-11  Kelsey & Eden Park Safer Neighbourhood Teams  Bromley Kelsey & Eden Park  E05013999
2025  11  2025-11  Kelsey & Eden Park Safer Neighbourhood Teams  Bromley Kelsey & Eden Park  E05013999
2025  11  2025-11  Mottingham Safer Neighbourhood Teams  Bromley Mottingham  E05014000
2025  11  2025-11  Mottingham Safer Neighbourhood Teams  Bromley Mottingham  E05014000
2025  11  2025-11  Mottingham Safer Neighbourhood Teams  Bromley Mottingham  E05014000
2025  11  2025-11  Mottingham Safer Neighbourhood Teams  Bromley Mottingham  E05014000
2025  11  2025-11  Mottingham Safer Neighbourhood Teams  Bromley Mottingham  E05014000
2025  11  2025-11  Mottingham Safer Neighbourhood Teams  Bromley Mottingham  E05014000
2025  11  2025-11  Mottingham Safer Neighbourhood Teams  Bromley Mottingham  E05014000
2025  11  2025-11  Orpington Safer Neighbourhood Teams  Bromley Orpington  E05014001
2025  11  2025-11  Orpington Safer Neighbourhood Teams  Bromley Orpington  E05014001
2025  11  2025-11  Orpington Safer Neighbourhood Teams  Bromley Orpington  E05014001
2025  11  2025-11  Penge & Cator Safer Neighbourhood Teams  Bromley Penge & Cator  E05014002
2025  11  2025-11  Penge & Cator Safer Neighbourhood Teams  Bromley Penge & Cator  E05014002
2025  11  2025-11  Penge & Cator Safer Neighbourhood Teams  Bromley Penge & Cator  E05014002
```

(32)



```
# Decode Borough_SNT if it's still encoded
df['Borough_SNT_Decoded'] = label_encoders['Borough_SNT'].inverse_transform(df['Borough_SNT'])

# Group by decoded Borough_SNT and sum the Count
crime_by_borough = df.groupby('Borough_SNT_Decoded')['Count'].sum().sort_values(ascending=False)

# Display top Borough
print("Borough with the most crimes:")
print(crime_by_borough.head(1))
```

```
Borough with the most crimes:
Borough_SNT_Decoded
Westminster    326808
Name: Count, dtype: int64
```

```
# Save the current notebook (optional but safe)
import IPython
IPython.display.display(IPython.display.Javascript('IPython.notebook.save_checkpoint();'))
```

```
!jupyter nbconvert --to html "/content/prediction-2.ipynb"
```

```
[NbConvertApp] Converting notebook /content/prediction-2.ipynb to html
[NbConvertApp] Writing 304704 bytes to /content/prediction-2.html
```

(33)