To what extent can socioeconomic factors explain the spatial variation in crime rates at the LSOA level in London?

Preparation

- https://github.com/Yingxi-Z/casa06datascience
- Number of words: 2032
- Runtime: 8 minutes (*Memory 10 GB, CPU Intel i7-10700 CPU @2.90GHz*)
- Coding environment: Python Anaconda 3
-

Table of contents

1. Introduction

[go back to the top]

The spatial variation evolution of crime rates have long been central topics in urban studies, social policy, and criminology. In London, there are significant differences in crime rates across different areas. These spatial disparities may reflect variations in socioeconomic conditions, population structure, and environmental factors within communities(Alves,2018). Understanding the spatial

dimensions is crucial for formulating sound public safety policies, optimizing resource allocation, and informing social interventions.

Based on this, the study aims to address the question: To what extent can socioeconomic factors explain the spatial variation in crime rates at the LSOA level in London?

the study uses crime data from 2021, applying models such as linear regression, random forest, and XGBoost to examine socioeconomic factors (e.g., unemployment rate, median income, housing conditions). The aim is to identify the most significant variables associated with crime rates at the LSOA level.

2. Literature Review

[go back to the top]

The relationship between socioeconomic factors and crime rates has long been a central focus in criminology. The Social Disorganization Theory proposed by Shaw and McKay (1942) provides a foundational framework, suggesting that poverty, residential mobility, and ethnic heterogeneity undermine community cohesion, thereby increasing the likelihood of crime. Subsequent studies have validated and extended this theory, identifying strong correlations between crime rates and variables such as unemployment (Raphael & Winter-Ebmer, 2001), income inequality (Kelly, 2000), and low educational attainment (Lochner & Moretti, 2004). Moreover, urban spatial characteristics (Bernasco & Block, 2011) and community stability (Hipp, 2007) have also been shown to influence the spatial distribution of crime, indicating the multidimensional nature of its causes.

Although previous research has provided valuable insights into the spatial dimensions of crime, traditional regression models often fail to capture complex non-linear relationships, and the long-term potential of micro-level geographies, such as LSOAs, has not yet been fully explored. To address these gaps, this study employs machine learning to analyze crime data at the LSOA level in London, aiming to develop a more comprehensive framework for understanding crime dynamics over space.

3. Data

[go back to the top]

3.1 Data Source

This study relies on multiple data sources to analyze crime patterns from spatial perspective. For the cross-sectional analysis, we integrate crime statistics with socioeconomic indicators at the LSOA level in 2021.

1. **Crime Data**: Crime data were obtained from the UK Government open data portal (London Datastore), which provides crime records collected by the Metropolitan Police Service at the LSOA level. The dataset includes information on crime type, location, and date.

2. **Socioeconomic Data**: Socioeconomic data were sourced from the UK Office for National Statistics (ONS) 2021 Census and the Consumer Data Research Centre (CDRC) data portal. These sources offer comprehensive information on demographic characteristics, economic conditions, housing, education, and other relevant variables at the LSOA level.

3. **Boundary Data**: The LSOA geographic boundary files were retrieved from the ONS Open Geography Portal. These shapefiles support spatial analysis and visualization of crime patterns across different areas.

3.2 Data propressing and key variables

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid")

# ========== read crime data ==========
crime_url = "https://raw.githubusercontent.com/Yingxi-Z/casa06datascience/refs/h
crime_df = pd.read_csv(crime_url)
crime_df.columns = crime_df.columns.astype(str)

# choose 2021
month_cols = [col for col in crime_df.columns if col.startswith("2021")]
crime_df["Total Crime 2021"] = crime_df[month_cols].sum(axis=1)
crime_df = crime_df[["LSOA Code", "Total Crime 2021","Borough"]].rename(columns=

# ========== read socialeconomic data and calculate varibles ==========
def load_health():
    df = pd.read_csv("https://raw.githubusercontent.com/Yingxi-Z/casa06datascien
    df.columns = df.columns.str.strip()
    df["healthy rate (%)"] = (
        (df["General health: Good health"] + df["General health: Very good healt
        df["General health: Total: All usual residents"]
    ) * 100
    return df[["geography code", "healthy rate (%)"]]

def load_birth():
    df = pd.read_csv("https://raw.githubusercontent.com/Yingxi-Z/casa06datascien
    df.columns = df.columns.str.strip()
    df["Born-UK rate (%)"] = (
        df["Country of birth: Europe: United Kingdom; measures: Value"] /
        df["Country of birth: Total; measures: Value"]
    ) * 100
    df = df.rename(columns={
        "Country of birth: Total; measures: Value": "population"
    })
    return df[["geography code", "Born-UK rate (%)","population"]]
```

```
def load_youth():
    df = pd.read_csv("https://raw.githubusercontent.com/Yingxi-Z/casa06datascien
    df.columns = df.columns.str.strip()
    df["Youth rate (%)"] = (
        (df["Age: Aged 15 to 19 years"] + df["Age: Aged 20 to 24 years"]) /
        df["Age: Total"]
    ) * 100
    return df[["geography code", "Youth rate (%)"]]

def load_qualification_unemp():
    qual = pd.read_csv("https://raw.githubusercontent.com/Yingxi-Z/casa06datasci
    qual["Low qualification rate (%)"] = (
        qual["Highest level of qualification: No qualifications"] /
        qual["Highest level of qualification: Total: All usual residents aged 16
    ) * 100
    qual["High qualification rate (%)"] = (
        qual["Highest level of qualification: Level 4 qualifications and above"]
        qual["Highest level of qualification: Total: All usual residents aged 16
    ) * 100


    unemp = pd.read_csv("https://raw.githubusercontent.com/Yingxi-Z/casa06datasc
    qual = qual.merge(unemp[["geography code",
        "Unemployment history: Total: All usual residents aged 16 years and over
        on="geography code", how="left")

    qual["Unemployment rate (%)"] = (
        qual["Unemployment history: Total: All usual residents aged 16 years and
        qual["Highest level of qualification: Total: All usual residents aged 16
    ) * 100

    return qual[["geography code", "Low qualification rate (%)", "High qualifica
    ]]

def load_pop_density():
    df = pd.read_csv("https://raw.githubusercontent.com/Yingxi-Z/casa06datascien
    df = df.rename(columns={
        "Population Density: Persons per square kilometre; measures: Value": "Pc
    })
    return df[["geography code", "Population density"]]


def load_house_price():
    df = pd.read_csv("https://raw.githubusercontent.com/Yingxi-Z/casa06datascien
    df["house price"] = df[["hpmd202003", "hpmd202103"]].mean(axis=1)
    return df[["geography code", "house price"]]

def load_IMD():
    df=pd.read_csv("https://raw.githubusercontent.com/Yingxi-Z/casa06datascience
    df = df.rename(columns={
        "Index of Multiple Deprivation (IMD) Score": "IMD",
        "Income Score (rate)": "income",
        "Employment Score (rate)": "employment rate",
        "Education, Skills and Training Score": "education score",
        "Health Deprivation and Disability Score": "health score",
        "Barriers to Housing and Services Score": "housing and services score",
        "Living Environment Score": "living score"
    })
    return df[["geography code", "IMD","income","employment rate","education sco
               "health score","housing and services score","living score"]]
```

```
def load_PTAL():
    df = pd.read_csv("https://raw.githubusercontent.com/Yingxi-Z/casa06datascien
    return df[["geography code", "AvPTAI2015"]]

# ========== merge all data ==========
dfs = [
    load_health(),
    load_birth(),
    load_youth(),
    load_qualification_unemp(),
    load_pop_density(),
    load_house_price(),
    load_IMD(),
    load_PTAL(),
    crime_df
]

from functools import reduce
final_df = reduce(lambda left, right: pd.merge(left, right, on="geography code",

import numpy as np

# log crime rate
#final_df['Log_Crime'] = np.log(final_df['Total Crime 2021'] + 1)
final_df['house price'] = np.log(final_df['house price'] + 1)
final_df['crime rate'] = final_df['Total Crime 2021'] / final_df['population']*1
final_df['log crime rate'] = np.log(final_df['crime rate'] + 1)

final_df = final_df.dropna().reset_index(drop=True)
```

To prepare the data for the cross-sectional analysis, we performed standard preprocessing steps including merging datasets by LSOA codes, handling missing values, and applying normalization where necessary. The following table summarizes the key variables used in the analysis, along with their descriptions and respective data sources:

Variable Name	Description	Type	Source
Crime rate	Total number of recorded crimes per 100 people in each LSOA area (2021)	Count (Numeric)	gov.uk
log crime rate	Log-transformed version of crime rate	Continuous	Calculated from Crime Rate
Youth rate (%)	Percentage of youth population (age 16-24) in each LSOA area	Ratio (%)	ONS Census 2021
Population density	Total population density in each LSOA area (2021)	Count (Numeric)	ONS Census 2021
IMD (Index of Multiple Deprivation)	Index of Multiple Deprivation score (2019)	Continuous	CDRC

Variable Name	Description	Type	Source
AvPTAI2015 (Average Public Transport Accessibility Index 2015)	Accessibility to public transport in each LSOA (2015)	Continuous	gov.uk
Education score	Score reflecting education attainment level in each LSOA area (2019)	Continuous	CDRC
Housing and services score	Score reflecting quality of housing and public services in each LSOA (2019)	Continuous	CDRC
Living score	Score reflecting living standards in each LSOA area (2019)	Continuous	CDRC

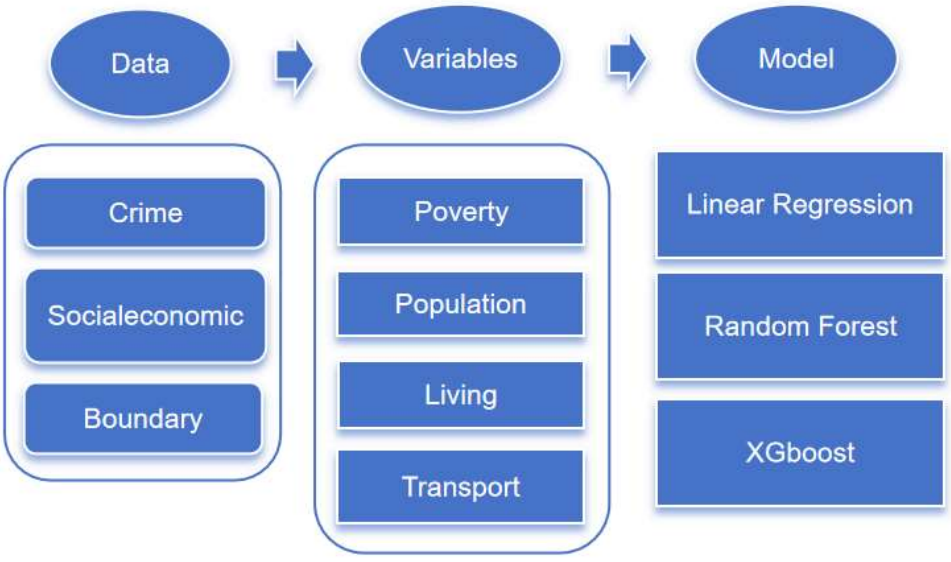
3.3 Data Limitation

Although efforts have been made to include a wide range of socioeconomic variables, the selection is ultimately constrained by data availability and public access. As a result, the current indicators may not fully capture the complex social structure and living conditions of each LSOA. For example, important subjective factors such as residents' mental health, community cohesion, or perceived safety are not reflected in the available data.

4 Methodology

[go back to the top]

This study adopts a approach of spatial cross-sectional analysis to explore the explanatory power of socioeconomic factors on crime rates at the LSOA level in London. The flow chart is as follows.



4.1 Correlation Analysis

To assess the appropriateness of the selected variables, Pearson's correlation coefficient is first used to test the linear relationship between independent variables (socioeconomic indicators) and the target variable (crime rates at the LSOA level)(Han,2017). By calculating the correlation matrix and visualizing it as a heatmap, we can identify potentially strong correlations and uncover issues of multicollinearity between predictor variables, providing a foundation for the subsequent modeling process.

4.2 Variable Importance Analysis and Predictive Modeling

Three regression models are employed to model crime rates and evaluate the importance of socioeconomic variables(Lin,2017):

Multiple Linear Regression (MLR): This baseline model is used to assess the explanatory power of variables under a linear relationship and evaluate the overall model performance.

Random Forest Regression: A Bagging-based ensemble learning method that constructs multiple decision trees and averages their predictions to reduce the risk of overfitting. The importance of variables is measured by the Mean Decrease in Impurity (MDI).

XGBoost Regression: An enhanced gradient boosting algorithm based on decision trees, known for its superior feature selection and generalization capabilities. Feature importance is measured by "Gain" (the contribution of each variable to model performance), which is effective for handling high-dimensional and nonlinear data.

To ensure the robustness of model performance evaluation, all models are validated using fold cross-validation, which involves repeated training and validation to obtain generalization performance metrics (such as RMSE and R^2). The prediction accuracy and explanatory power of the models are then compared.

5. Results and discussion

[go back to the top]

5.1 Exploratory Data Analysis

As shown in **Figure 1 and 2**, the distribution of the selected variables reveals several key patterns. The crime rate displays a highly skewed distribution, with a significant gap between the minimum (0.97) and maximum (281.41) values. This wide range is due to a small number of LSOAs experiencing exceptionally high crime levels, resulting in the presence of outliers. To mitigate the impact of these

extreme values on subsequent modeling and improve the normality of the distribution, the logarithmic transformation of the crime rate is applied.

Most socioeconomic variables exhibit approximately normal distributions. The youth rate, population density, Index of Multiple Deprivation (IMD), Public Transport Accessibility Level (PTAL), education score, and living environment score all show bell-shaped curves with relatively symmetric distributions, as supported by their summary statistics. These patterns suggest that the majority of LSOAs fall within a typical range, with only a few extreme observations.

In [18]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# EDA
features = [
    "crime rate", "log crime rate", "Youth rate (%)",
    "Population density", "IMD",
    "AvPTAI2015", "education score",
    "housing and services score", "living score"
]

stats_summary = final_df[features].agg(['min', 'max', 'mean', 'median', 'std']).
stats_summary.columns = ['Min', 'Max', 'Mean', 'Median', 'Std Dev']

print("Summary Statistics:")
display(stats_summary)

# Distribution + KDE
plt.figure(figsize=(16, 16))
for i, feature in enumerate(features):
    plt.subplot(3, 3, i + 1)
    sns.histplot(data=final_df, x=feature, kde=True, bins=30, color='skyblue')
    plt.title(f"Distribution of {feature}")
    plt.xlabel(feature)
    plt.ylabel("Count")
plt.tight_layout()
plt.figtext(0.5, -0.01, "Figure 1: Distribution of Key Variables.",
            ha="center", fontsize=20)
plt.show()

# Boxplots
plt.figure(figsize=(16, 16))
for i, feature in enumerate(features):
    plt.subplot(3, 3, i + 1)
    sns.boxplot(data=final_df, y=feature, color='lightgreen')
    plt.title(f"Boxplot of {feature}")
    plt.ylabel(feature)
    plt.grid(True, axis='y', linestyle='--', alpha=0.5)
plt.tight_layout()
plt.figtext(0.5, -0.01,"Figure 2: Boxplots of Key Variables", ha="center", fonts
plt.show()
```

```
Summary Statistics:
```

	Min	Max	Mean	Median	Std Dev
crime rate	0.969426	281.409118	8.037775	6.349206	8.965081
log crime rate	0.677742	5.643357	2.038519	1.994592	0.504339
Youth rate (%)	3.813281	59.331652	11.831323	11.361142	3.567840
Population density	122.600000	49139.300000	9722.739251	8637.800000	5867.949362
IMD	2.300000	59.000000	20.667764	19.400000	10.560249
AvPTAI2015	0.191141	121.887000	12.902421	9.272710	11.976617
education score	0.000000	59.000000	12.490661	10.600000	9.709221
housing and services score	6.900000	65.000000	30.982292	30.200000	9.506583
living score	5.500000	91.600000	28.899165	28.300000	10.727845

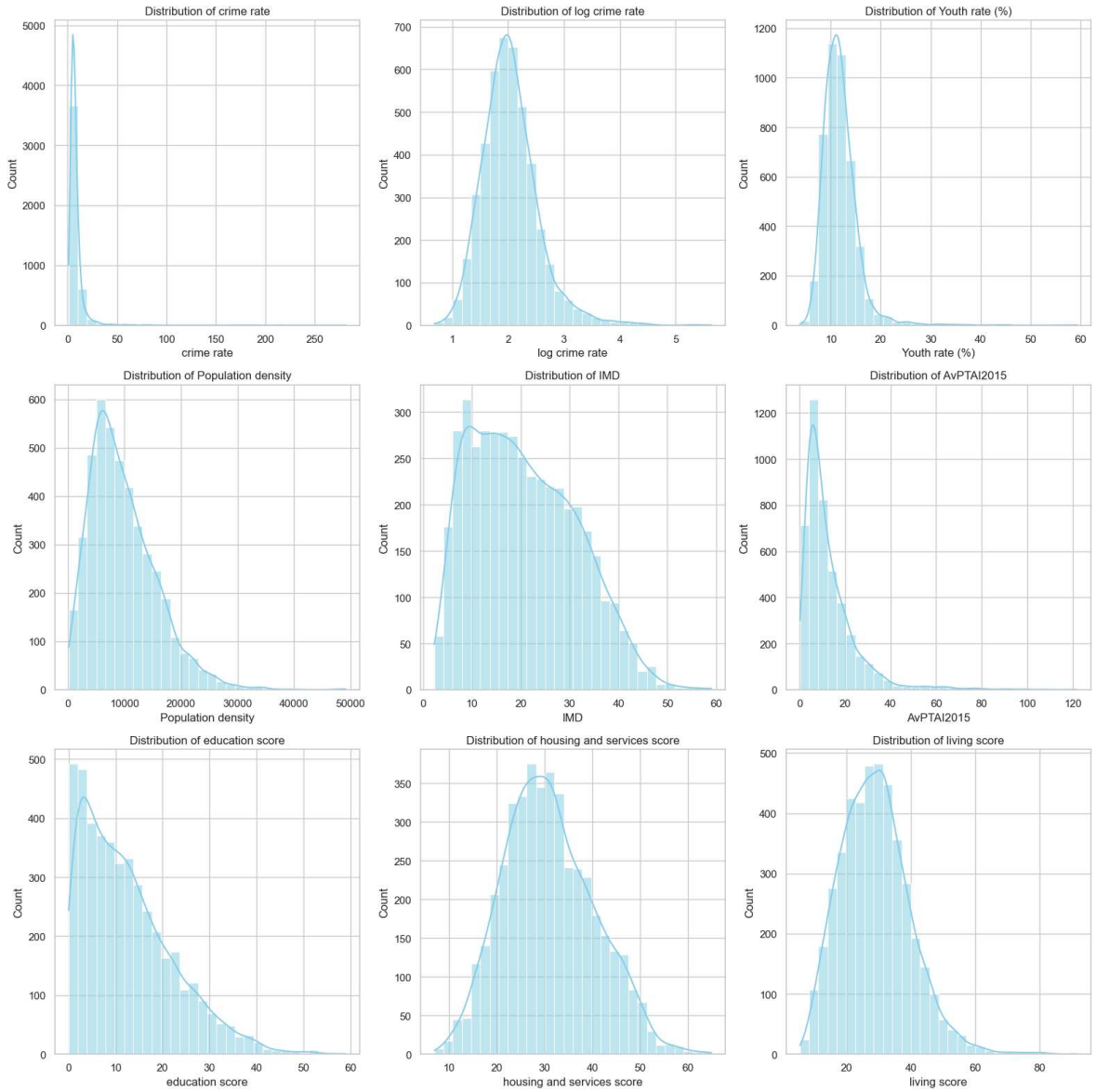


Figure 1: Distribution of Key Variables.

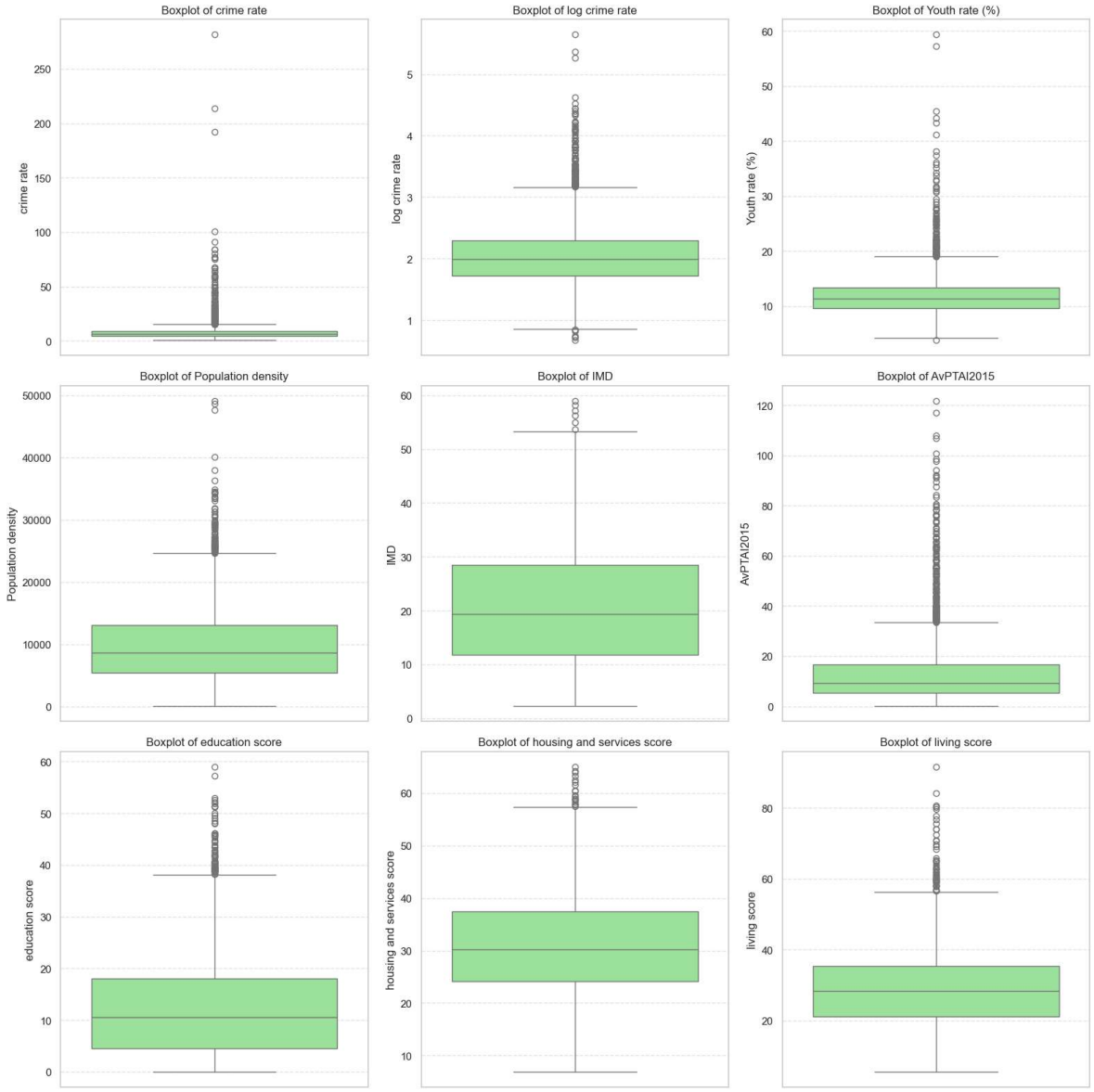


Figure 2: Boxplots of Key Variables

5.2 Spatial Distribution of crime rate in LSOA

As shown in **Figure 3**, the spatial distribution of the log-transformed crime rate across London LSOAs in 2021 reveals a clear central-peripheral pattern. High crime rates are predominantly concentrated in the central areas of the city and gradually decrease as one moves toward the outskirts. The blank or white areas on the map indicate missing values. This spatial trend reflects the typical urban crime concentration phenomenon, where densely populated and socioeconomically diverse central zones may tend to experience higher crime levels(Han,2017).

In [17]:
```
import geopandas as gpd

lsoa_gdf = gpd.read_file("https://raw.githubusercontent.com/Yingxi-Z/casa06datas
#print(lsoa_gdf.columns)
merged = lsoa_gdf.merge(final_df, left_on="LSOA11CD", right_on="geography code")

columns_to_plot = [ "log crime rate"]

for column in columns_to_plot:
```

```
    fig, ax = plt.subplots(1, 1, figsize=(10, 8))
    merged.plot(
        column=column,
        cmap='viridis',
        linewidth=0.1,
        ax=ax,
        edgecolor='0.8',
        legend=True,
        legend_kwds={
            'label': f"{column}",
            'orientation': "horizontal"
        }
    )
    plt.tight_layout()
    plt.figtext(0.5, -0.01,"Figure 3: Spatial Distribution of crime rate", ha="c
    plt.show()
```

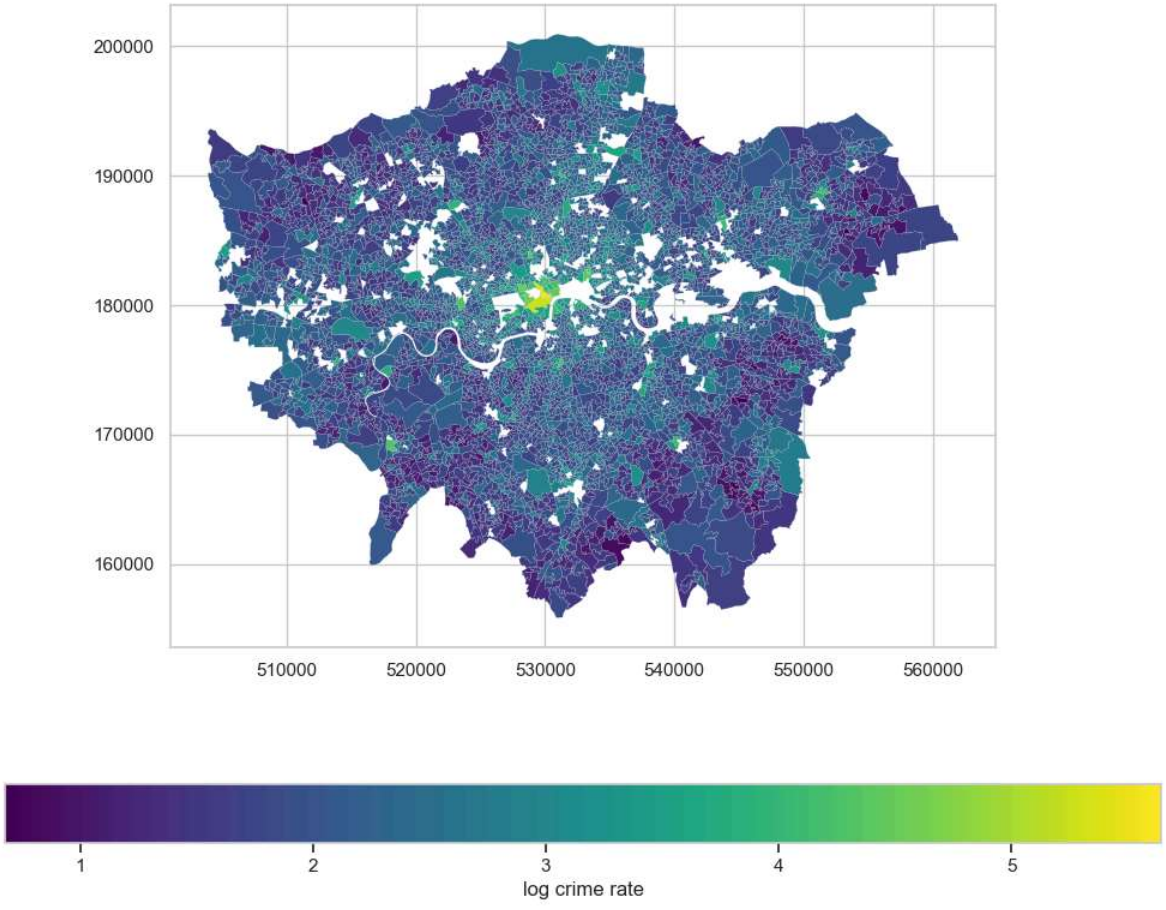


Figure 3: Spatial Distribution of crime rate

5.3 Correlation Analysis

In [16]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
import scipy.stats as stats
import pandas as pd
from statsmodels.tools import add_constant
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.tools.tools import add_constant
```

```
variables = [
    "crime rate", "log crime rate", "Youth rate (%)",
    "Population density", "IMD",
    "AvPTAI2015", "education score",
    "housing and services score", "living score"
]

corr_df = final_df[variables].dropna()


corr_matrix = corr_df.corr()

plt.figure(figsize=(12, 10))

sns.heatmap(
    corr_matrix,
    annot=True,
    fmt=".2f",
    cmap="coolwarm",
    linewidths=0.5,
    square=True,
    cbar_kws={"shrink": .8}
)
plt.xticks(rotation=45, ha="right")
plt.yticks(rotation=0)
plt.tight_layout()


plt.figtext(0.5, -0.01, "Figure 4: Pearson Correlation Matrix of Key Variables",
plt.show()
```

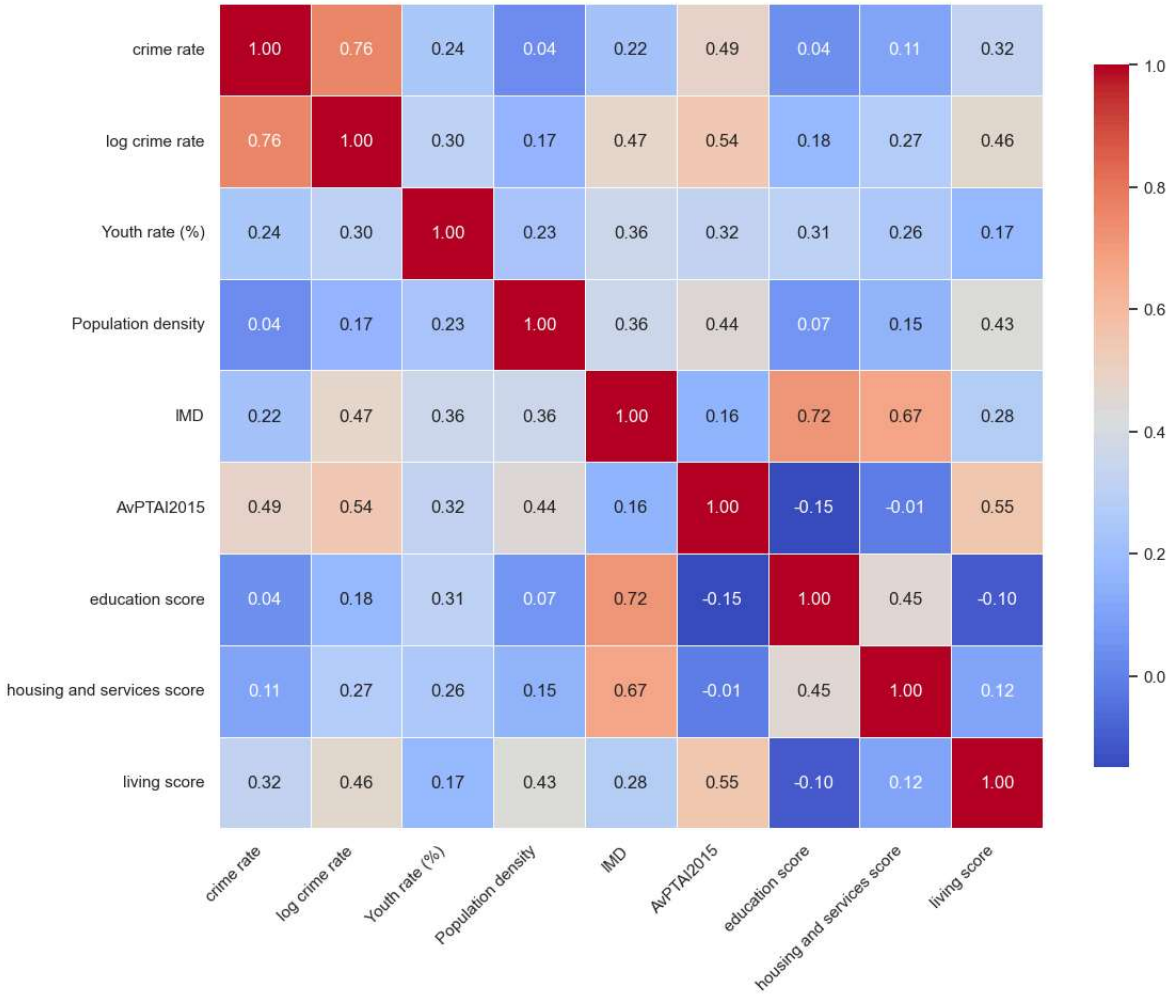


Figure 4: Pearson Correlation Matrix of Key Variables

As shown in **Figure 4**, the heatmap of the correlation matrix indicates a significant relationship between socialeconomic factors and crime rates. Specifically, areas with better transport accessibility (higher PTAL), higher poverty levels (higher IMD), and better living conditions (higher living score) tend to have higher crime rates. This suggests that areas with better connectivity and improved living conditions are more prone to criminal activity. These areas often have higher population densities and more complex socio-economic conditions, which may influence crime dynamics. Better transport accessibility may lead to more population movement, increasing the opportunities for crime, while higher poverty levels and lower social welfare (indicated by higher IMD) may contribute to social instability and increase crime incentives(Cook,2013). Additionally, higher living scores may be associated with more prosperous and resource-rich areas, which are more likely to attract criminal behavior(Pratt,2005).

5.4 Machine Learning

5.4.1 Linear Regression

In [5]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import LinearRegression
```

```python
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import scipy.stats as stats
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import pandas as pd

features = [
    "Youth rate (%)",
    "Population density", "IMD",
    "AvPTAI2015", "education score",
    "living score",  "housing and services score"
]

X = final_df[features]
y = final_df["log crime rate"]

X_clean = X.dropna()
y_clean = y[X_clean.index]

# VIF
X_with_const = add_constant(X)
vif_data = pd.DataFrame()

vif_data["features"] = X_with_const.columns
vif_data["VIF"] = [variance_inflation_factor(X_with_const.values, i) for i in ra
vif_data = vif_data[vif_data["features"] != "const"]

print(vif_data)


# train LinearRegression model
X_train, X_test, y_train, y_test = train_test_split(X_clean, y_clean, test_size=

lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)

# predict
y_pred = lin_reg.predict(X_test)

# measure
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error (MSE): {mse:.4f}")
print(f"R²: {r2:.4f}")

# residuals
residuals = y_test - y_pred

print(f"mean residuals: {np.mean(residuals):.4f}")
print(f"std residuals: {np.std(residuals):.4f}")
```

```
                     features       VIF
1              Youth rate (%)  1.323890
2          Population density  1.472901
3                         IMD  4.394657
4                   AvPTAI2015  1.797751
5              education score  2.865491
6                 living score  1.737265
7  housing and services score  1.921382
Mean Squared Error (MSE): 0.1140
R²: 0.4845
mean residuals: -0.0044
std residuals: 0.3376
```

In [6]:
```python
# After fitting the LinearRegression model
# Get the coefficients and the intercept
coefficients = lin_reg.coef_
intercept = lin_reg.intercept_

# Create the formula
formula = f"y = {intercept:.4f}"

# Add each coefficient and feature name to the formula
for coef, feature in zip(coefficients, X_clean.columns):
    formula += f" + ({coef:.4f}) * {feature}"

# Print the formula
print("Prediction Formula: ")
print(formula)
```

```
Prediction Formula:
y = 1.3043 + (0.0047) * Youth rate (%) + (-0.0000) * Population density + (0.024
5) * IMD + (0.0208) * AvPTAI2015 + (-0.0044) * education score + (0.0079) * livin
g score + (-0.0005) * housing and services score
```

In [15]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import numpy as np

# Residuals
residuals_lr = y_test - y_pred

fig, axes = plt.subplots(1, 3, figsize=(20, 6))

# === Actual vs Predicted ===
sns.scatterplot(x=y_test, y=y_pred, ax=axes[0], color='blue', alpha=0.6)
axes[0].plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
axes[0].set_xlabel('Actual Values')
axes[0].set_ylabel('Predicted Values')
axes[0].set_title('Actual vs Predicted (Linear Regression)')

# === Residuals vs Predicted===
sns.scatterplot(x=y_pred, y=residuals_lr, ax=axes[1], color='darkorange', alpha=
axes[1].axhline(0, color='red', linestyle='--')
axes[1].set_xlabel('Predicted Values')
axes[1].set_ylabel('Residuals')
axes[1].set_title('Residuals vs Predicted (Linear Regression)')

# === Q-Q plot ===
stats.probplot(residuals_lr, dist="norm", plot=axes[2])
```

```
axes[2].set_title("Q-Q Plot of Residuals (Linear Regression)")

plt.figtext(0.5, -0.03, "Figure 5: Evaluation Plots for Linear Regression", ha="
plt.tight_layout()
plt.show()
```

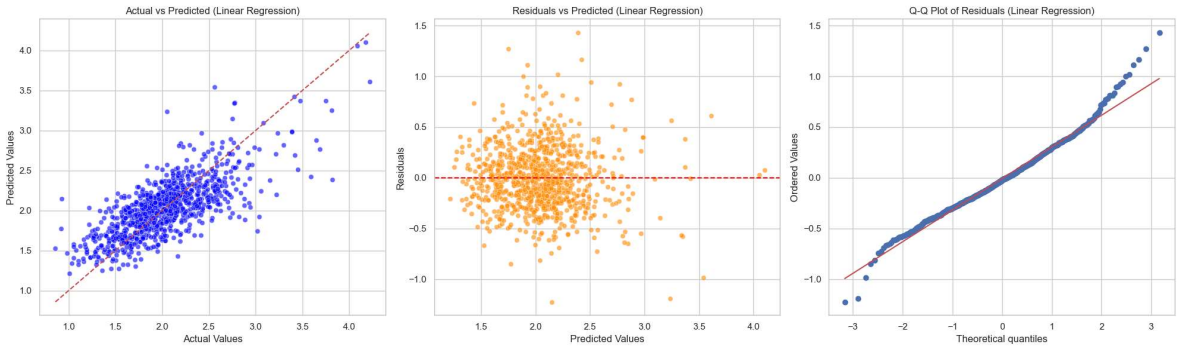


Figure 5: Evaluation Plots for Linear Regression

By splitting the data into 80% training and 20% testing sets, and performing linear regression modeling using machine learning, the R-squared value of the model was found to be 0.4845. This indicates that the model only can explain less than 50% of the variance in the data. Upon examining the residual plot and QQ plot in **Figure 5**, it was observed that the predicted values are symmetrically distributed around the residual line, and the residuals follow a roughly normal distribution.

Prediction Formula:

$$\log \text{crime rate} = 1.3043 + (0.0047) \times \text{Youth rate } (\%) + (-0.0000) \times \text{Populat}$$
$$\times \text{ education score} + (-0.0005) \times \text{housing and}$$

This formula suggests that, among the variables included, "IMD" (Index of Multiple Deprivation) and "AvPTAI2015" (Average Public Transport Accessibility Index for 2015) have the most significant positive contributions to the log crime rate. Conversely, the "education score" and "housing and services score" have very minor negative impacts, with coefficients close to zero.

5.4.2 Random Forest model

In [12]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
from statsmodels.tools.tools import add_constant
from sklearn.model_selection import GridSearchCV

# RandomForestRegression
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
rf_model = RandomForestRegressor(random_state=42)

# GridSearchCV
param_grid = {
    'max_depth': [3, 5, 10, 15, 20],
    'n_estimators': [100, 200, 300],
    'random_state': [42]
}
```

```
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=5, n_jc
grid_search.fit(X_train, y_train)

print(f"best params: {grid_search.best_params_}")

best_rf_model = grid_search.best_estimator_

y_pred = best_rf_model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"MSE: {mse:.4f}")
print(f"Best model - R²: {r2:.4f}")

# feature importance
feature_importances = pd.DataFrame({
    'feature': features,
    'importance': best_rf_model.feature_importances_
}).sort_values(by='importance', ascending=False)

print("\nFeature Importances:")
print(feature_importances)

depths = [3, 5, 10, 15, 20]
train_errors = []
test_errors = []

for depth in depths:
    rf_model = RandomForestRegressor(max_depth=depth, random_state=42)
    rf_model.fit(X_train, y_train)

    train_pred = rf_model.predict(X_train)
    test_pred = rf_model.predict(X_test)

    train_errors.append(mean_squared_error(y_train, train_pred))
    test_errors.append(mean_squared_error(y_test, test_pred))

plt.plot(depths, train_errors, label="Training Error")
plt.plot(depths, test_errors, label="Test Error")
plt.xlabel('Tree Depth')
plt.ylabel('Mean Squared Error')
plt.legend()
plt.figtext(0.5, -0.1, "Figure 6: Learning Curve: Error vs Tree Depth", ha="cent
plt.show()
```

```
Fitting 5 folds for each of 15 candidates, totalling 75 fits
best params: {'max_depth': 10, 'n_estimators': 300, 'random_state': 42}
MSE: 0.0983
Best model - R²: 0.5554

Feature Importances:
                      feature  importance
3                   AvPTAI2015    0.365070
2                          IMD    0.279782
1          Population density    0.160885
5                living score    0.056561
0              Youth rate (%)    0.052381
6  housing and services score    0.044167
4              education score    0.041155
```

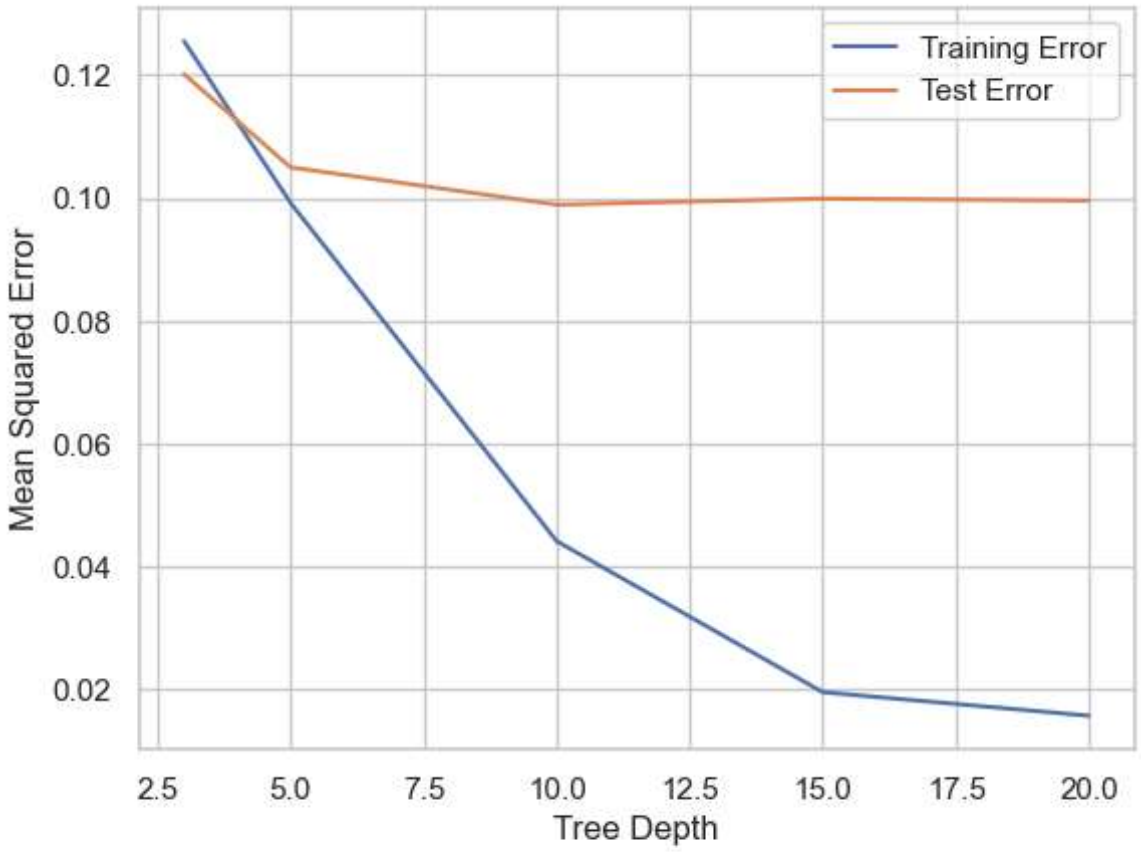


Figure 6: Learning Curve: Error vs Tree Depth

In [96]:
```
from sklearn.model_selection import cross_val_score
import numpy as np

# croscrs_val
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

cv_scores_r2 = cross_val_score(rf_model, X, y, cv=5, scoring='r2')
cv_scores_mse = cross_val_score(rf_model, X, y, cv=5, scoring='neg_mean_squared_

print(f"cross_val R²: {cv_scores_r2.mean():.4f} ± {cv_scores_r2.std():.4f}")
print(f"cross_val MSE: {-cv_scores_mse.mean():.4f} ± {cv_scores_mse.std():.4f}")
```

```
cross_val R²: 0.5393 ± 0.0555
cross_val MSE: 0.1125 ± 0.0133
```

In [14]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats

residuals_rf = y_test - y_pred

fig, axes = plt.subplots(1, 3, figsize=(20, 6))

# === Actual vs Predicted ===
sns.scatterplot(x=y_test, y=y_pred, ax=axes[0], color='blue', alpha=0.6)
axes[0].plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
axes[0].set_xlabel('Actual Values')
axes[0].set_ylabel('Predicted Values')
axes[0].set_title('Actual vs Predicted (Random Forest)')
```

```
# === Residuals vs Predicted ===
sns.scatterplot(x=y_pred, y=residuals_rf, ax=axes[1], color='darkorange', alpha=
axes[1].axhline(0, color='red', linestyle='--')
axes[1].set_xlabel('Predicted Values')
axes[1].set_ylabel('Residuals')
axes[1].set_title('Residuals vs Predicted (Random Forest)')

# === Q-Q Plot of Residuals ===
stats.probplot(residuals_rf, dist="norm", plot=axes[2])
axes[2].set_title("Q-Q Plot of Residuals (Random Forest)")

plt.figtext(0.5, -0.03, "Figure 7: Evaluation Plots for Random Forest Regression
plt.tight_layout()
plt.show()
```

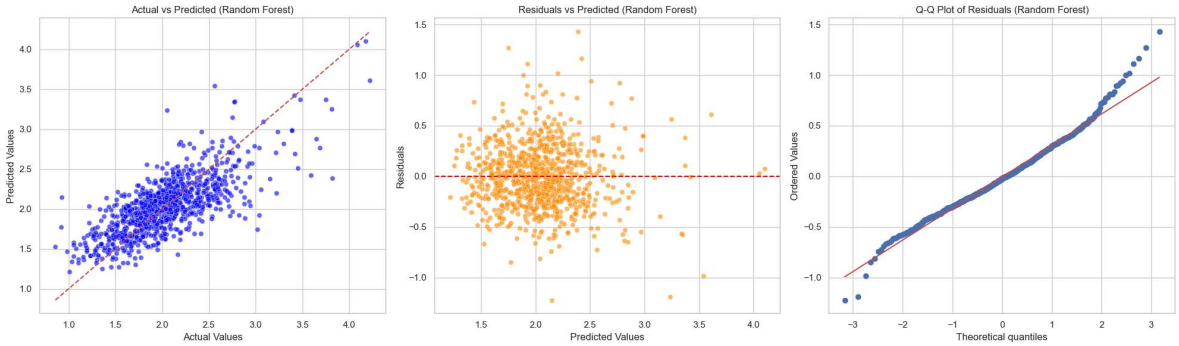


Figure 7: Evaluation Plots for Random Forest Regression

As shown in **Figure 7**, the learning curve suggests that the model performs optimally when the tree depth is set to 10. The best hyperparameters, determined through cross-validation, are n_estimators=300, and random_state=42. These parameters ensure a balance between underfitting and overfitting, optimizing the model's performance.

The best model achieves a Mean Squared Error of 0.0983, which indicates that the model's errors are relatively small and the predictions are fairly accurate. As shown in Figure 8, the actual value is close to the predicted curve. R^2 value of 0.5554 implies that the model explains about 55.5% of the variability in the log crime rate, which suggests a moderate fit to the data. The results from cross-validation further validate that the model is relatively stable and not overly sensitive to different subsets of the data, with an average R^2 of 0.5393 ± 0.0555 and an MSE of 0.1125 ± 0.0133.

Looking at the feature importance in **Figure 8**, Average Public Transport Accessibility Index has the largest contribution, accounting for 36.5% of the model's predictive power. This suggests that areas with better public transportation accessibility tend to have higher crime rates, potentially due to factors like higher foot traffic or easier access to urban areas, which could facilitate criminal activity(Saeed,2023).

The Index of Multiple Deprivation follows closely, with 27.98% of the total importance. This indicates a significant relationship between areas with higher levels of deprivation and increased crime rates. Population density, which accounts for 16.1% of the importance, also plays a critical role. This aligns with the idea that densely populated areas are more likely to experience higher crime

rates, likely due to factors such as increased anonymity, greater social inequality, and limited access to resources(Lin,2017).

Other variables, such as living score (5.66%), Youth rate (5.24%), housing and services score*(4.42%), and education score (4.12%), contribute less but still have some predictive power. For example, higher youth rates might correlate with higher crime, while better housing and services could mitigate crime.

5.4.3 XGboost model

In [10]:
```
import pandas as pd
import xgboost as xgb
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split, GridSearchCV
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats

X = final_df[["Youth rate (%)",
    "Population density", "IMD",
    "AvPTAI2015", "education score",
    "living score",  "housing and services score"]]
y = final_df["log crime rate"]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_

# XGBoostRegressor
xgb_reg = xgb.XGBRegressor(objective='reg:squarederror',
                           n_estimators=100,
                           learning_rate=0.1,
                           max_depth=5,
                           random_state=42)

xgb_reg.fit(X_train, y_train)

param_grid = {
    'n_estimators': [100, 200, 300],
    'learning_rate': [0.01, 0.05, 0.1],
    'max_depth': [3, 5, 7]
}

grid_search = GridSearchCV(estimator=xgb_reg, param_grid=param_grid, cv=5, scori
grid_search.fit(X_train, y_train)

print(f"best_params: {grid_search.best_params_}")
best_model = grid_search.best_estimator_

y_pred_best = best_model.predict(X_test)
mse_best = mean_squared_error(y_test, y_pred_best)
r2_best = r2_score(y_test, y_pred_best)

print(f"Best model - MSE: {mse_best:.4f}")
print(f"Best model - R²: {r2_best:.4f}")

xgb.plot_importance(best_model, importance_type='weight', max_num_features=10)
plt.figtext(0.5, -0.1, "Figure 8: Best Model Feature Importance", ha="center", f
plt.show()
```

```
residuals_best = y_test - y_pred_best
```

```
best_params: {'learning_rate': 0.05, 'max_depth': 3, 'n_estimators': 200}
Best model - MSE: 0.0985
Best model - R²: 0.5547
```

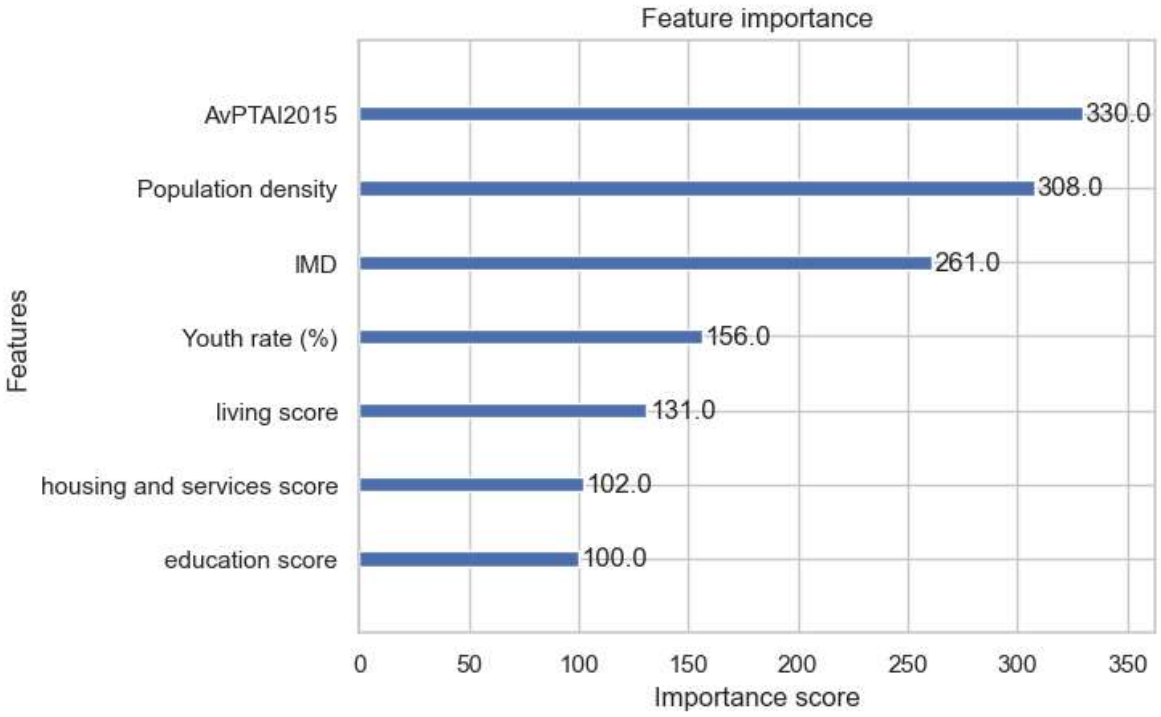


Figure 8: Best Model Feature Importance

In [13]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats

residuals_best = y_test - y_pred_best

fig, axes = plt.subplots(1, 3, figsize=(20, 6))

# === Actual vs Predicted ===
sns.scatterplot(x=y_test, y=y_pred_best, ax=axes[0], color="dodgerblue", alpha=0
axes[0].plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
axes[0].set_title("Actual vs Predicted")
axes[0].set_xlabel("Actual Values")
axes[0].set_ylabel("Predicted Values")

# === Residuals vs Predicted ===
sns.scatterplot(x=y_pred_best, y=residuals_best, ax=axes[1], color="forestgreen"
axes[1].axhline(0, color='red', linestyle='--')
axes[1].set_title("Residuals vs Predicted")
axes[1].set_xlabel("Predicted Values")
axes[1].set_ylabel("Residuals")

# === Q-Q Plot of Residuals ===
stats.probplot(residuals_best, dist="norm", plot=axes[2])
axes[2].set_title("Q-Q Plot of Residuals")
axes[2].set_xlabel("Theoretical Quantiles")
axes[2].set_ylabel("Sample Quantiles")

plt.figtext(0.5, -0.03, "Figure 9: Model Evaluation Plots (XGB)", ha="center", f
```

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

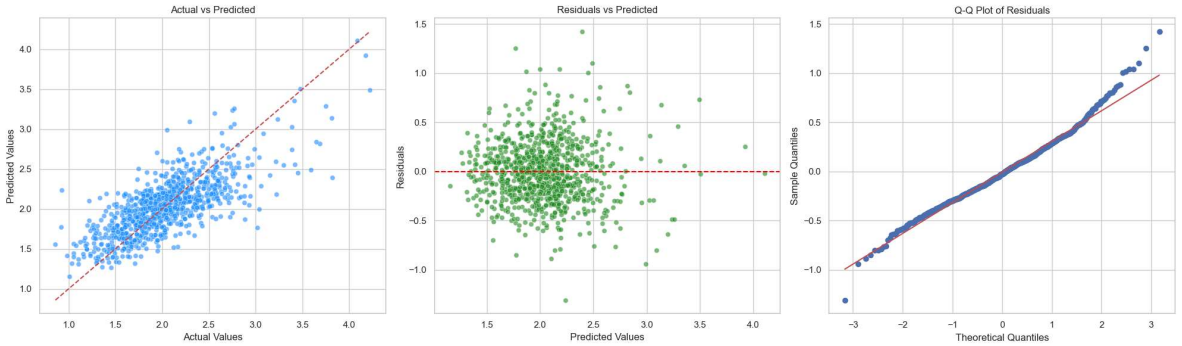


Figure 9: Model Evaluation Plots (XGB)

The crime rate prediction model constructed based on the XGBoost algorithm yielded results with a Mean Squared Error (MSE) of 0.0985 and a coefficient of determination (R^2) of 0.5547. This indicates that the model can explain 55.47% of the variance in crime rates, demonstrating good predictive capability, while approximately 45% of the variance remains unexplained, potentially due to other underlying influencing factors not included in the model.

Feature importance analysis revealed that the Public Transport Accessibility Index (AvPTAI2015) ranked highest with an importance score of 330.0, consistent with the crime opportunity theory and indicating that areas with better transportation access are more prone to becoming crime hotspots. Population density followed closely with an importance score of 308.0, supporting the characteristic of weaker social control in high-density areas. The Index of Multiple Deprivation (IMD) showed an importance score of 261.0, directly reflecting the positive correlation between socioeconomic disadvantage and crime rates. The youth population proportion had an importance score of 156.0, aligning with classical criminological theories regarding the relationship between age and crime. Additionally, community resource indicators such as living environment scores, housing service scores, and education scores also demonstrated significant influence.

Analysis of the residual plot and QQ plot in **Figure 9** shows that the predicted values are symmetrically distributed around the residual line and closely approximate the actual observed values, indicating high predictive accuracy of the model. Furthermore, the residuals largely conform to a normal distribution, validating that the model meets the fundamental assumptions of linear regression and demonstrating the strong applicability and reliability of the XGBoost algorithm for crime rate prediction.

Conclusion

[go back to the top]

The analysis of London's LSOAs shows that socioeconomic factors moderately explain the spatial variation in crime. Linear regression achieved an R^2 of 0.48,

identifying IMD and public transport accessibility (PTAI) as significant positive contributors. However, Random Forest and XGBoost performed better, both reaching R^2 values above 0.55, highlighting potential nonlinear relationships between predictors and crime.

Across all models, PTAI, IMD, and population density consistently emerged as the most influential factors. This consistency increases confidence in their importance. The spatial distribution of crime also showed a central–peripheral pattern, with higher crime rates in central areas characterized by greater connectivity and complex socioeconomic dynamics.

Although all models identified similar patterns, their explanatory power varied. The moderate R^2 values suggest that while socioeconomic variables are important, they do not fully explain crime patterns. Unmeasured factors such as policing strategies, social cohesion, or urban design, may also play significant roles.

In conclusion, while machine learning models improve predictive accuracy, fully understanding urban crime requires moving beyond traditional socioeconomic data. Future studies should incorporate additional contextual and behavioral variables to uncover the broader mechanisms behind crime distribution in London(Saeed,2023).

References

[go back to the top]

Bernasco, W., & Block, R. (2011). Robberies in Chicago: A block-level analysis of the influence of crime generators, crime attractors, and offender anchor points. Journal of Research in Crime and Delinquency, 48(1), 33-57.

Cook, S. and Winfield, T. (2013) ‘Crime across the States: are US crime rates converging?’, Urban Studies, 50(9). 1724–41.

Han, L., Bandyopadhyay, S. and Bhattacharya, S. (2013) ‘Understanding the determinants of property and violent crime in England and Wales’, Social Science Research Network.

Hipp, J. R. (2007). Income inequality, race, and place: Does the distribution of race and class within neighborhoods affect crime rates? Criminology, 45(3), 665-697.

Kelly, M. (2000). Inequality and crime. The Review of Economics and Statistics, 82(4), 530-539.

L. Lin, T. -Y. Chen and L. -C. Yu.(2017). "Using Machine Learning to Assist Crime Prevention," 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Hamamatsu, Japan, 1029-1030

Lochner, L., & Moretti, E. (2004). The effect of education on crime: Evidence from prison inmates, arrests, and self-reports. American Economic Review, 94(1), 155-189.

Luiz G.A. Alves, Haroldo V. Ribeiro, Francisco A. Rodrigues.(2018) Crime prediction through urban metrics and statistical learning,Physica A: Statistical Mechanics and its Applications, 435-443.

Pratt, T.C. and Cullen, F.T. (2005) ' Assessing macro-level predictors and theories of crime', in: M. Tonry (Ed.), Crime and Justice: A Review of Research, Chicago, IL.: University of Chicago Press.

Raphael, S., & Winter-Ebmer, R. (2001). Identifying the effect of unemployment on crime. The Journal of Law and Economics, 44(1), 259-283.

Saeed, Ruaa Mohammed and Abdulmohsin, Husam Ali. (2023) "A study on predicting crime rates through machine learning and data mining using text" Journal of Intelligent Systems.

Sampson, R. J., Raudenbush, S. W., & Earls, F. (1997). Neighborhoods and violent crime: A multilevel study of collective efficacy. Science, 277(5328), 918–924.