LOK JAGRUTI KENDRA UNIVERSITY

Master of Computer Applications (Integrated)

7th Semester

Research Project
(050123707)

CRIME RATE PREDICTION AND TREND ANALYSIS

Group No:

IMCA_7_26

Internal Guide:

Prof. Khushali Vala

Project By:

Faizan Memon	**D-31 (21018501210031)**
Naved Memon	**D-33 (21018501210033)**
Mo. Aasim Saiyed	**D-55 (2101801210055)**

CERTIFICATE

This is to certify that Mr. ***Faizan Memon*** *with enrollment number* ***21018501210031*** *Semester* ***7th*** *has successfully completed his* ***Research Project*** *in the* ***Machine Learning*** *from the department of* ***Master of Computer Applications (Integrated)*** *during the academic year 2024 - 2025.*

Date of Submission:

Internal Guide Sign:

Internal Guide: ***Prof. Khushali Vala***

CERTIFICATE

This is to certify that Mr. ***Naved Memon*** *with enrollment number* ***21018501210033*** *Semester* ***7th*** *has successfully completed his* ***Research Project*** *in the* ***Machine Learning*** *from the department of* ***Master of Computer Applications (Integrated)*** *during the academic year 2024 - 2025.*

Date of Submission:

Internal Guide Sign:

Internal Guide: ***Prof. Khushali Vala***

CERTIFICATE

This is to certify that Mr. ***Mo. Aasim Saiyed*** *with enrollment number* ***21018501210055*** *Semester* ***7th*** *has successfully completed his* ***Research Project*** *in the* ***Machine Learning*** *from the department of* ***Master of Computer Applications (Integrated)*** *during the academic year 2024 - 2025.*

Date of Submission:

Internal Guide Sign:

Internal Guide: ***Prof. Khushali Vala***

TABLE OF CONTENT

1. Introduction

Crime rates have been increasing, creating major challenges for law enforcement agencies around the world. In India, many cities are facing issues with rising crime, and it has become clear that traditional methods of crime prevention are not enough. However, with the large amount of data now available, there is an opportunity to use this information to better understand crime and take action before it happens. By analysing crime data, police and authorities can gain important insights into where and when crimes are more likely to occur, allowing them to respond more effectively and prevent criminal activities.[1]

Even though there is a lot of crime data available, it's not always easy to turn it into useful information for predicting future crimes. This creates a big challenge for law enforcement, as they often have to wait for a crime to happen before they can respond, rather than being able to prevent it. Without the right tools to predict crime, it's difficult to identify areas where crimes are likely to occur, which leads to less efficient use of resources. Policymakers also struggle to develop strategies that can effectively reduce crime because they don't have enough data-driven insights to guide their decisions.[2]

The aim of this project is to predict crime rates and study crime trends in different Indian cities by looking at past crime data. By analysing this data, the project seeks to find patterns that can help law enforcement better understand when and where crimes are likely to happen. These insights can then be used to improve resource allocation, develop crime prevention strategies, and guide policy-making. This research will help law enforcement and policymakers make smarter, data-driven decisions to keep cities safer and reduce crime.[3]

1.1 Abstract

This project looks at crime patterns in India by studying past crime data. By analyzing trends and using predictive models, it aims to find the main factors that affect crime rates. It uses different types of data, like the kind of crime, the weapon used, the location, details about the victim, and how the police responded, to predict future crime trends. The main goal is to predict crime rates for the next few years and understand crime trends over time.[4]

1.2 Project Profile

Project Title	Crime rate prediction and trend analysis
Project Developer	Memon Faizan (Eno. 21018501210031) Memon Naved (Eno. 21018501210033) Saiyed Mo. Aasim (Eno. 21018501210055)
Hardware Require	Pentium 4 Micro Processor or Above RAM 1 GB or Above Hard Disk 60 GB or Above
Software Require	Window 7 or above
Application Use	Jupyter Notebook
Other Tools	MS Word 365 (Documentation) MS PowerPoint (Presentation)
Project Internal Guide	Prof. Khushali Vala
Duration	2 Months

1.3 Problem statement

As crime rates in India continue to grow, predicting crime patterns has become harder due to increasing population and the complexity of cities. Law enforcement agencies often struggle to plan and use their resources effectively because they lack accurate ways to predict where and when crimes might happen. Existing methods do not use past crime data well enough to forecast future trends. This project aims to solve that by analyzing historical data to predict future crime rates and trends, helping authorities better prepare and improve public safety.

1.4 Objective

- Study how crime rates have changed over time in India.
- Find out the key reasons why some areas have higher crime rates.
- Create a model that can predict future crime rates.
- Understand crime patterns better and offer useful insights to help law enforcement prepare and respond more effectively.

2. Organization of Data

2.1.Data Source

The dataset used for this project was collected from public crime records reported across various cities in India. The dataset includes details from 2020 onwards, encompassing multiple types of crimes, cities, and law enforcement responses. Sources include official government crime data portals, police departments, and public records.[5]

2.2.Data Category

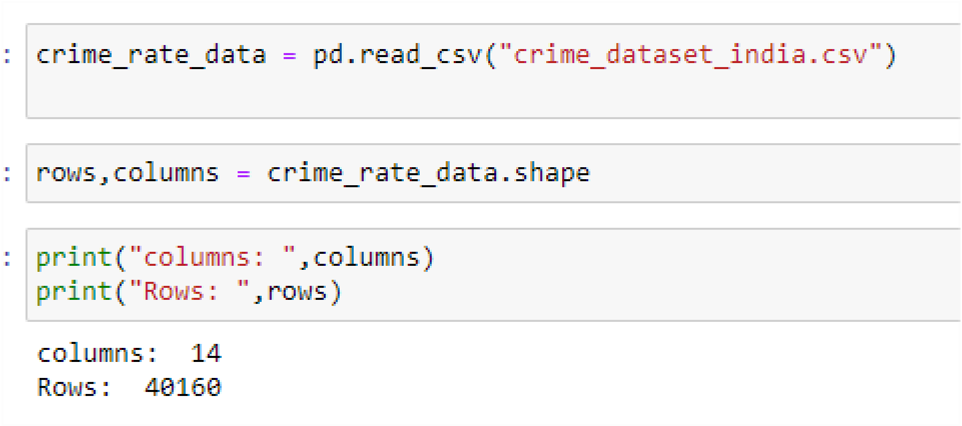
- **Crime Type**: Groups crimes into different categories like identity theft, homicide, kidnapping, burglary, and vandalism.
- **Geographic Data**: Shows city-based data to highlight where crimes happen in different areas.
- **Victim Information**: Includes details about victims' age and gender to understand which groups are more vulnerable.
- **Weapon Involvement**: Lists the types of weapons used in crimes, which helps determine the seriousness of the offenses.

2.3.Data Description

Columns Name	Description
Date Reported	The date when the crime was initially reported to the authorities.
City	The city where the crime occurred.
Crime Code	A unique identifier or code assigned to categorize the type of crime.
Crime Description	A detailed description of the crime that was committed.
Victim Age	The age of the victim at the time of the crime.
Victim Gender	The gender of the victim (e.g., Male, Female, Other).
Weapon Used	The type of weapon (if any) that was used to commit the crime.
Crime Domain	The specific domain or category under which the crime falls (e.g., violent, cyber).
Police Deployed	The number of police officers deployed in response to the crime.
Case Closed	Indicates whether the case is closed (Yes/No).
Date Case Closed	The date when the case was officially closed.

2.4.Data Volume

This image shows a code snippet from the "Crime Rate Prediction and Trend Analysis" project. It loads a dataset called "crime_dataset_india.csv" using pandas. The code checks the size of the dataset, which has 40,160 rows and 14 columns. This dataset will be used to train machine learning models for predicting crime trends based on various crime-related factors.



```
: crime_rate_data = pd.read_csv("crime_dataset_india.csv")

: rows,columns = crime_rate_data.shape

: print("columns: ",columns)
  print("Rows: ",rows)

  columns:  14
  Rows:  40160
```

2.5.Tools Techniques

- **Tools:**

1. **Jupyter Notebook**: For coding, data exploration, and visualization.
2. **PowerPoint**: For presenting findings visually.
3. **Excel**: For data analysis and quick visualizations.
4. **Word**: For writing reports and documentation.

- **Techniques:**

1 **NumPy**: A library for numerical computing, used for handling large arrays and performing mathematical operations.

2 **Pandas**: Ideal for data manipulation, providing easy-to-use tools for data cleaning and analysis.

3 **Matplotlib**: Used for creating static charts like line plots and bar graphs to visualize data.

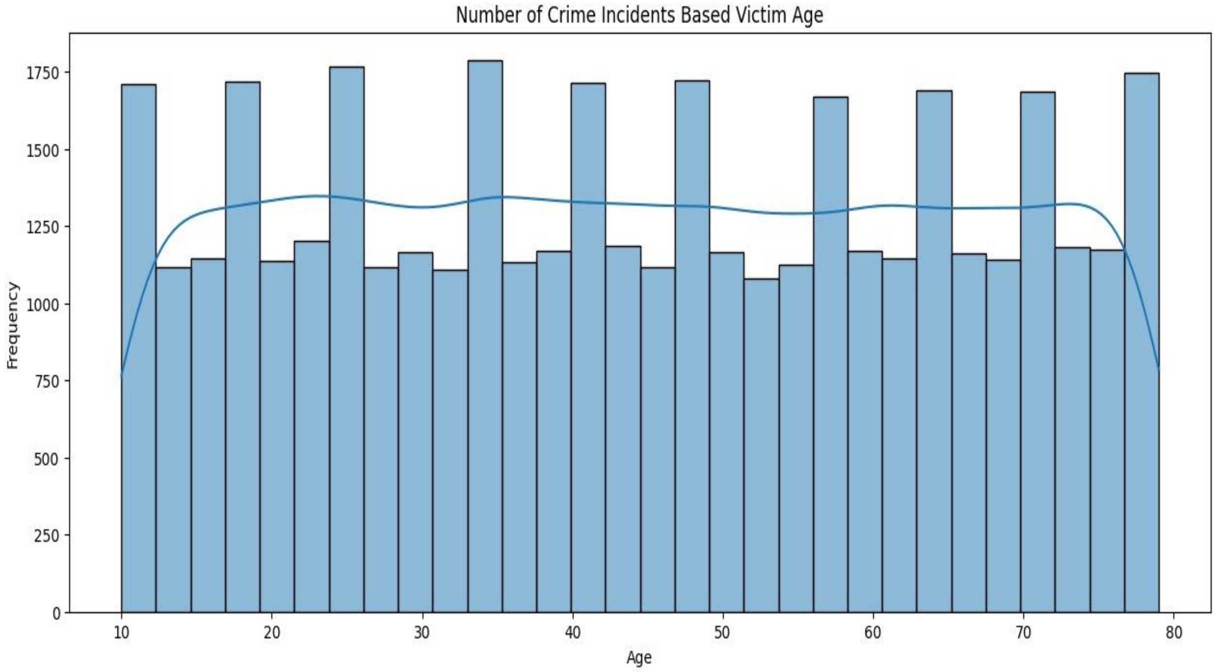
4 **Seaborn**: Simplifies creating attractive statistical visualizations, such as heatmaps and violin plots.[6]

5 **Scikit-learn**: A machine learning library for implementing models like classification and regression. [7]

6 **Plotly Express**: Creates interactive charts, allowing detailed data exploration.

3. Data Visualization(Original Data)

3.1.Distributed Graph



Crime Incidents Based on Victim Age:

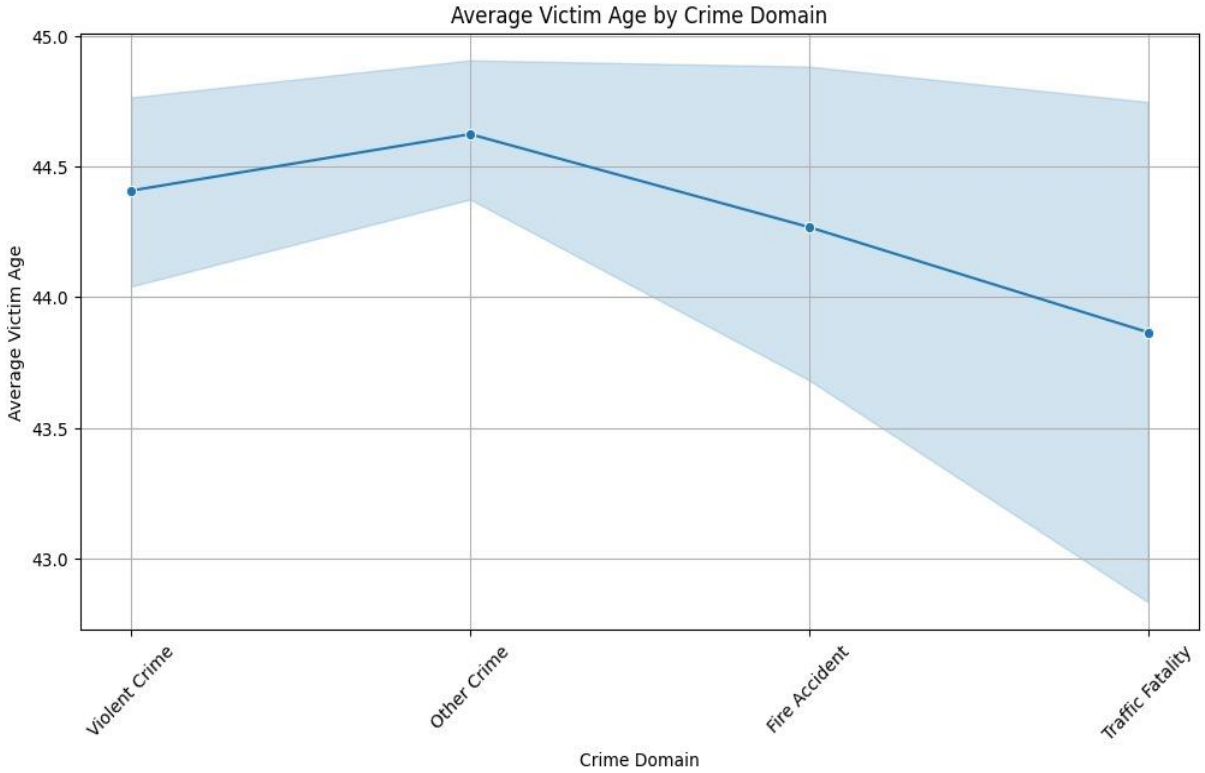
The bar graph above represents the distribution of crime incidents across different age groups of victims. The x-axis shows the age range, while the y-axis displays the frequency of incidents. Each bar indicates the number of crime incidents reported for a specific age group, with the blue line representing the trend of crime frequency.

From the graph, we observe that crime incidents are evenly distributed across all age groups, with slight fluctuations. The age groups around 10 years and 80 years show fewer incidents compared to middle-aged groups. However, the overall trend suggests that individuals from children to the elderly are equally susceptible to crimes.

The highest frequencies of crime incidents are observed around the ages of 10 and 80, indicating that minors and the elderly may be more vulnerable. Conversely, the age groups between 30 and 60 exhibit a stable but lower number of reported incidents, showing that working-age adults may experience fewer crimes.

This analysis points to the need for tailored crime prevention strategies across all age groups, with special attention to protecting the youngest and oldest segments of the population, who appear to be at higher risk.

3.2.Line Graph



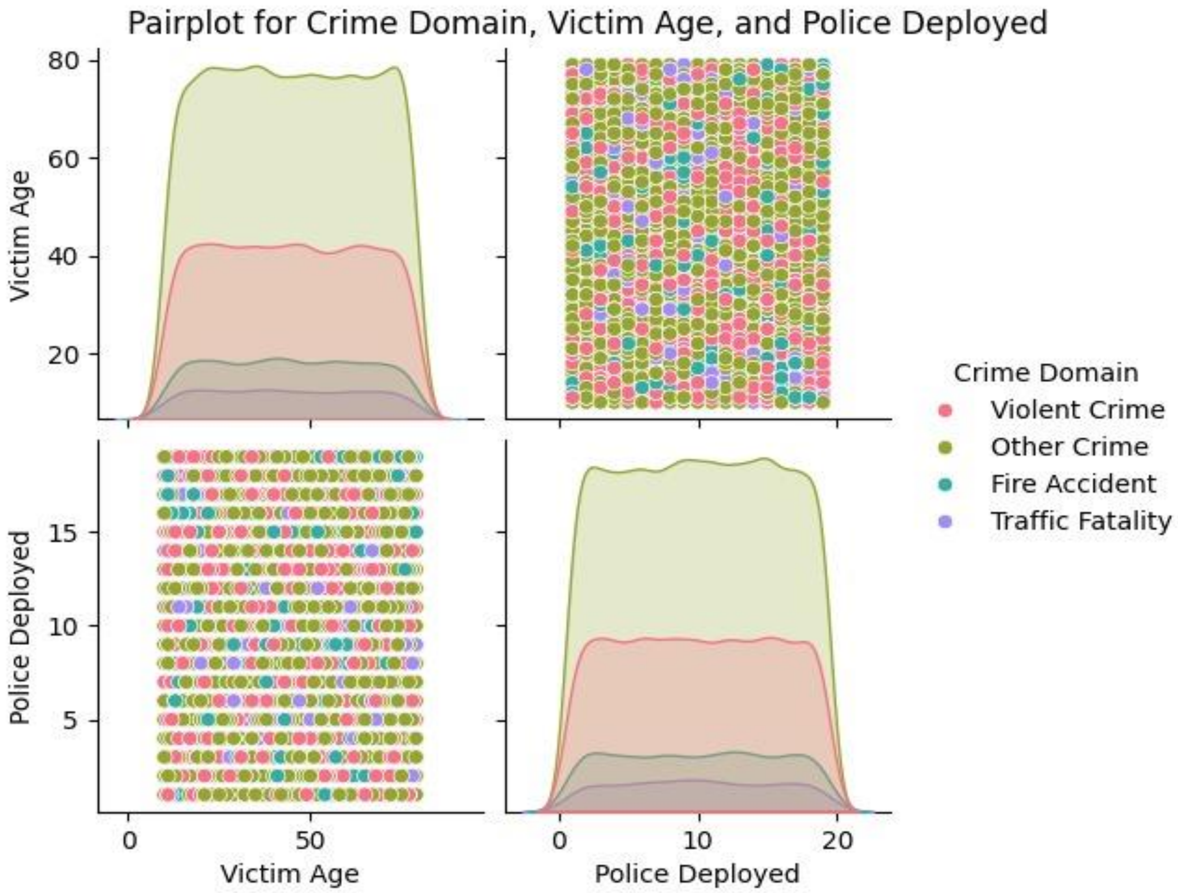
Average Victim Age by Crime Domain:

The line graph above illustrates the relationship between the average age of victims and various crime domains. The x-axis categorizes incidents, including Violent Crime, Other Crime, Fire Accident, and Traffic Fatality, while the y-axis represents the average age of victims. The blue line shows the trend in average victim age, with a shaded area indicating variability around the average values.

From the graph, we see that Other Crime involves the oldest victims, with an average age near 45 years, while Traffic Fatality victims are younger, with an average age below 44 years. Violent Crime and Fire Accident categories fall between these extremes, with average ages around 44.5 and 44 years. The overall trend shows a decline in average age from "Other Crime" to "Traffic Fatality."

The shaded area around the trend line suggests more variability in Traffic Fatality incidents, indicating a wider age range of victims. This analysis highlights the need for targeted crime prevention policies, with special focus on traffic fatalities, where younger victims may be disproportionately affected.

3.3.Pair Plot



Crime Domains Based on Victim Age and Police Deployment:

The pair plot above shows the relationship between crime domain, victim age, and police deployment. Each point represents an incident, color-coded by crime type, such as Violent Crime, Other Crime, Fire Accident, and Traffic Fatality. The diagonal density plots reveal the distribution of victim ages and police deployment across crime types.

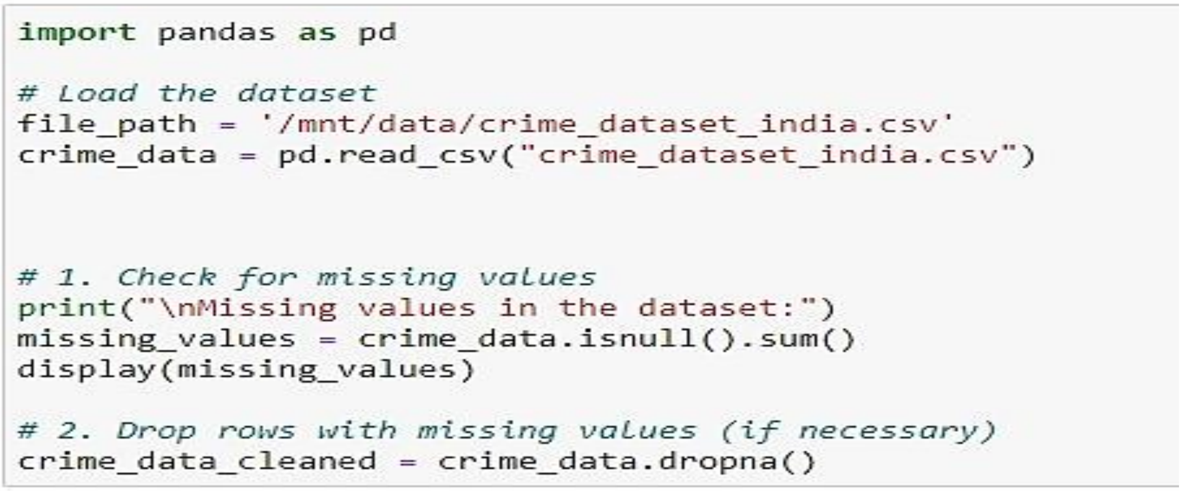
From the plot, we can see that Other Crime and Violent Crime impact a wide range of ages, while Traffic Fatality and Fire Accident affect mostly middle-aged to older individuals. Victims aged between 20 and 70 are involved in the majority of cases.

Police deployment varies by crime type, with Other Crime requiring the most resources, ranging from 2 to over 20 officers. Fire Accident and Traffic Fatality generally involve fewer officers.

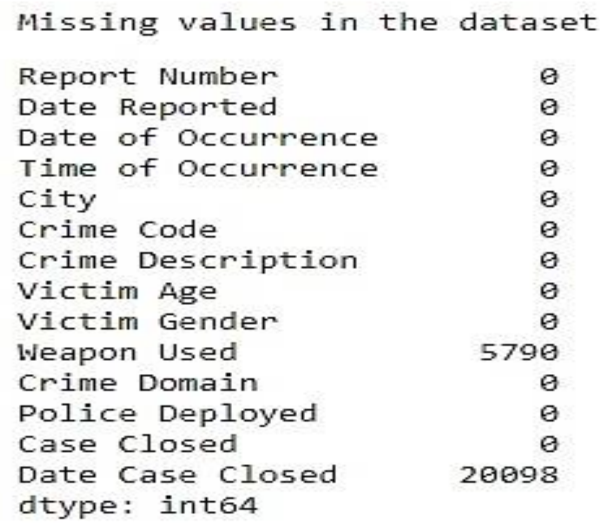
Overall, there is no direct correlation between victim age and the number of officers deployed. The deployment depends more on the crime's nature, not the victim's age, suggesting that resource allocation must remain adaptable across different crime domains.

4. Data Cleaning

- **Checking for null Values**



```
import pandas as pd

# Load the dataset
file_path = '/mnt/data/crime_dataset_india.csv'
crime_data = pd.read_csv("crime_dataset_india.csv")



# 1. Check for missing values
print("\nMissing values in the dataset:")
missing_values = crime_data.isnull().sum()
display(missing_values)

# 2. Drop rows with missing values (if necessary)
crime_data_cleaned = crime_data.dropna()
```

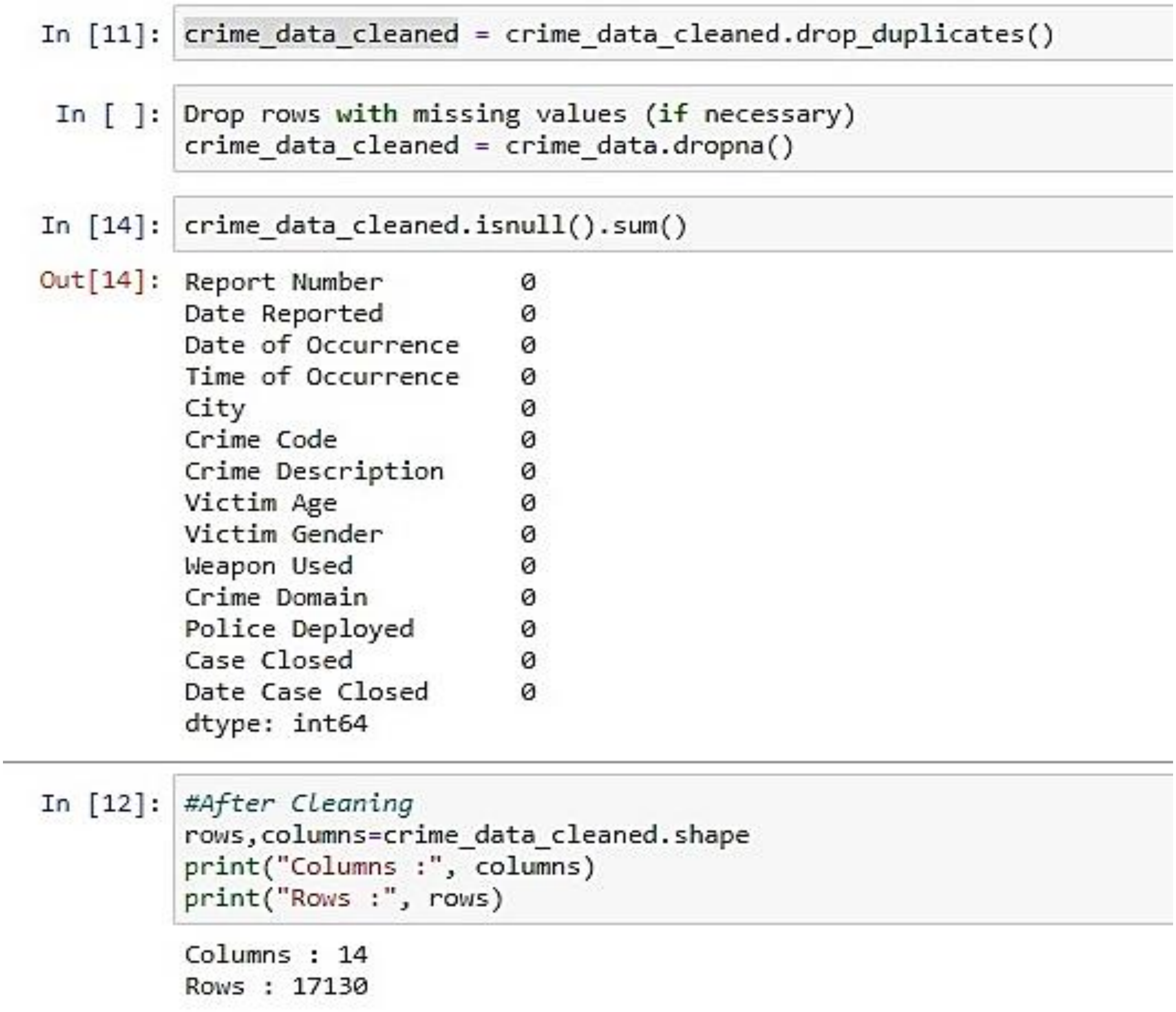


```
Missing values in the dataset:

Report Number              0
Date Reported              0
Date of Occurrence         0
Time of Occurrence         0
City                       0
Crime Code                 0
Crime Description          0
Victim Age                 0
Victim Gender              0
Weapon Used             5790
Crime Domain               0
Police Deployed            0
Case Closed                0
Date Case Closed       20098
dtype: int64
```

In the data preprocessing stage, we first loaded the crime dataset from a CSV file and checked for missing values across all columns. It was found that two columns, "Weapon Used" and "Date Case Closed," contained missing values, with 5790 and 20098 missing entries, respectively, while the rest of the columns had complete data. To handle these missing values, we used the dropna() function to remove rows with incomplete data, resulting in a cleaned dataset.

This cleaning process is a critical step to ensure the dataset is free of inconsistencies, which could otherwise affect the accuracy of the analysis. By eliminating rows with missing values, we prepare the data for further stages of the project, ensuring reliable inputs for machine learning model training.

- **Remove Unnecessary Columns & Null Values**



```
In [11]: crime_data_cleaned = crime_data_cleaned.drop_duplicates()

In [ ]:  Drop rows with missing values (if necessary)
         crime_data_cleaned = crime_data.dropna()

In [14]: crime_data_cleaned.isnull().sum()

Out[14]: Report Number          0
         Date Reported          0
         Date of Occurrence     0
         Time of Occurrence     0
         City                   0
         Crime Code             0
         Crime Description      0
         Victim Age             0
         Victim Gender          0
         Weapon Used            0
         Crime Domain           0
         Police Deployed        0
         Case Closed            0
         Date Case Closed       0
         dtype: int64

In [12]: #After Cleaning
         rows,columns=crime_data_cleaned.shape
         print("Columns :", columns)
         print("Rows :", rows)

         Columns : 14
         Rows : 17130
```

In the data preprocessing phase, we began by removing any duplicate records using the drop_duplicates() function to ensure the dataset contained only unique entries. Following this, we addressed missing values by applying the dropna() function, which eliminated any rows with incomplete data. After cleaning, we verified that no missing values remained in the dataset by using the isnull().sum() function, which returned zero missing entries across all columns.

Finally, we checked the shape of the cleaned dataset, which contained 14 columns and 17,130 rows. This cleaned dataset is now ready for further analysis, ensuring the integrity and reliability of the data used for training the machine learning model.

5. Hypothesis

5.1.Hypothesis Formation

1. **Crime Incidents Based on Victim Gender**:
   Hypothesis: Female victims are more frequently targeted in certain types of crimes (e.g., assault, harassment) compared to male victims, while male victims experience higher rates of violent crimes (e.g., robbery, homicide). City with the Highest Crime Rate.

2. **Crime Incidents Across Indian Cities**:
   Hypothesis: Cities with higher urbanization, such as Delhi and Mumbai, report significantly higher crime rates compared to smaller cities like Faridabad and Nasik.

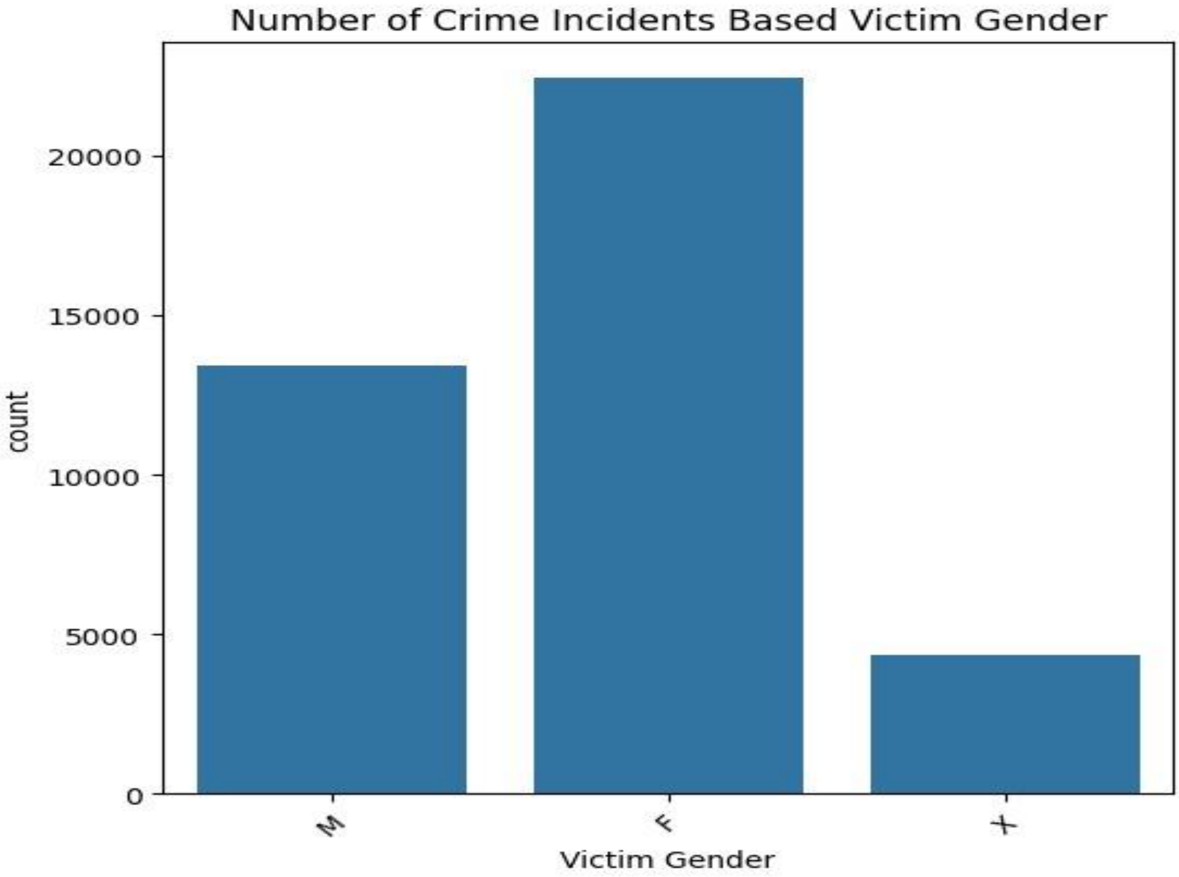
3. **Crime Occurrence by City and Crime Domain**:
   Hypothesis: Non-violent crimes (e.g., theft, fraud) are more relevant in larger cities like Mumbai and Delhi, while violent crimes (e.g., assault, homicide) are more evenly distributed across both large and small cities.

4. **Police Deployment Based on Crime Domain**:
   Hypothesis: Violent crimes and traffic fatalities require larger police deployments compared to non-violent crimes, with cities like Delhi and Bengaluru (Bangalore) allocating the most resources to address these crimes.

5. **Crime Distribution Across Indian Cities**:
   Hypothesis: Crime is more concentrated in metropolitan areas, with cities like Delhi and Mumbai experiencing the highest number of crime incidents, while smaller cities like Ahmedabad and Surat have significantly lower crime rates.

5.2. Implementation of Hypothesis

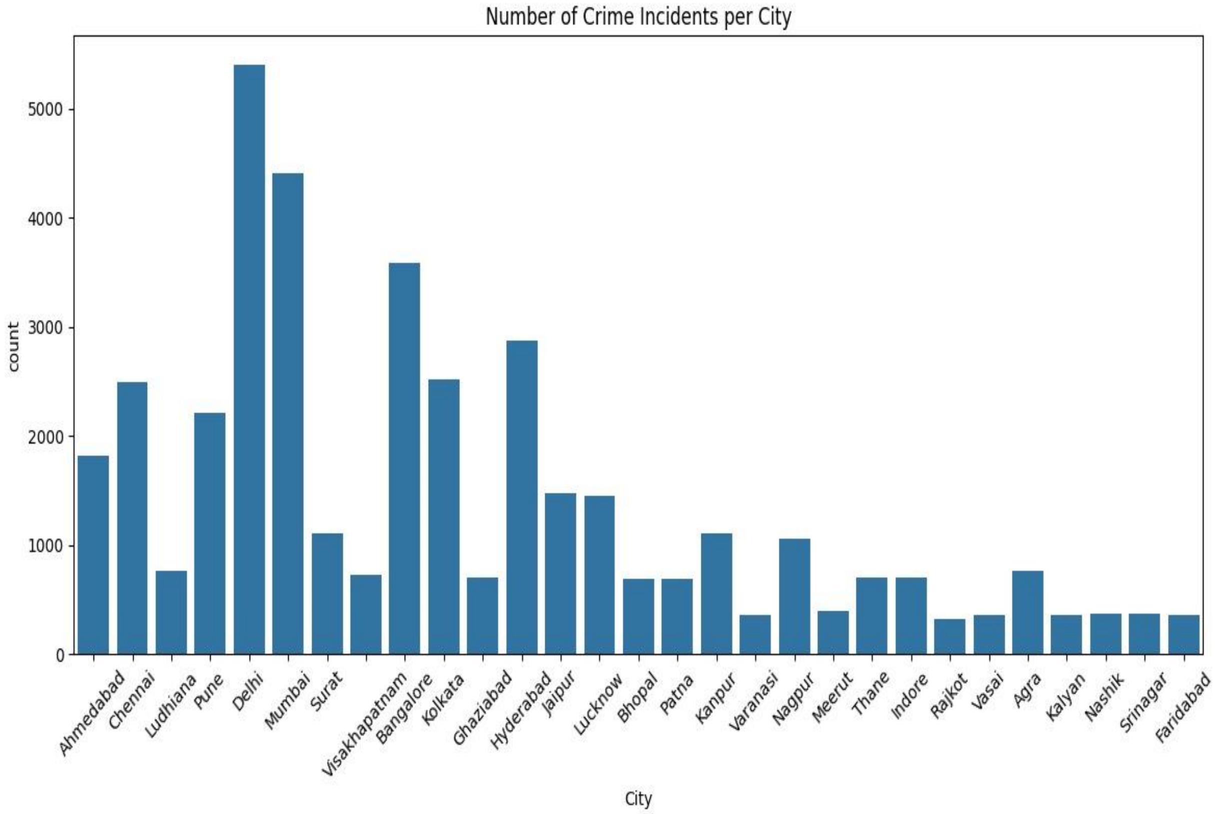
- **Crime Incidents Based on Victim Gender**



The bar graph above represents the distribution of crime incidents based on the gender of the victims. The dataset categorizes victims into three gender groups: Male (M), Female (F), and Other (+). As observed from the graph, a significant portion of crime victims are female, with over 20,000 reported incidents, which is the highest among all groups. Male victims account for approximately 15,000 incidents, while the 'Other' category shows fewer than 5,000 cases, indicating that non-binary or unspecified genders experience a lower number of reported crime incidents.

This trend suggests that females may be more vulnerable to certain types of crimes, or that crimes against females are more frequently reported. The disparity in crime incidents between genders highlights the need for targeted crime prevention strategies, especially focusing on areas and crime types where women are more frequently affected. Furthermore, the low number of cases in the 'Other' category could suggest underreporting or a lack of comprehensive data collection on gender non-binary individuals, an area that should be addressed in future crime data gathering efforts.

- **Crime Incidents Across Indian Cities**



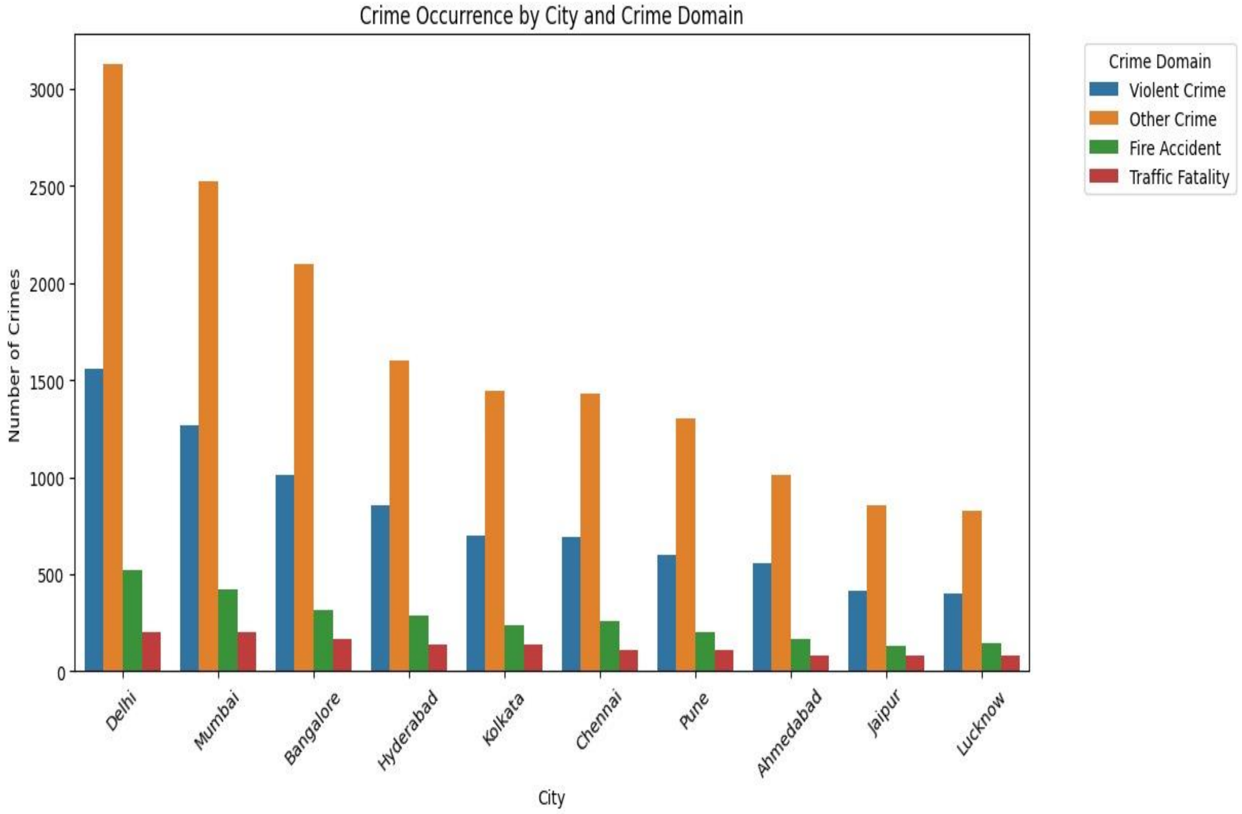
The bar graph above illustrates the number of crime incidents reported in various cities across India. The data shows a significant variation in crime rates, with some cities reporting a notably higher number of incidents compared to others.

Delhi stands out with the highest number of reported crimes, exceeding 5,000 incidents, followed by Mumbai with over 3,000 cases. This highlights these metropolitan areas as high-crime zones, likely due to their large populations and higher urbanization levels. Other cities such as Kolkata, Bhopal, and Visakhapatnam also display relatively high crime rates, with incident counts between 2,000 and 3,000.

In contrast, cities like Faridabad, Srinagar, and Nasik report fewer than 500 incidents, indicating lower crime rates in these regions. This disparity in crime incidents across cities could be attributed to factors such as population density, law enforcement efficiency, socio-economic conditions, and regional crime trends.

The data also reflects potential patterns in urban crime, suggesting that larger cities with dense populations tend to experience more crime. These findings can aid policymakers and law enforcement agencies in focusing resources and implementing crime prevention strategies in cities with higher reported incidents.

- **Crime Occurrence by City and Crime Domain**



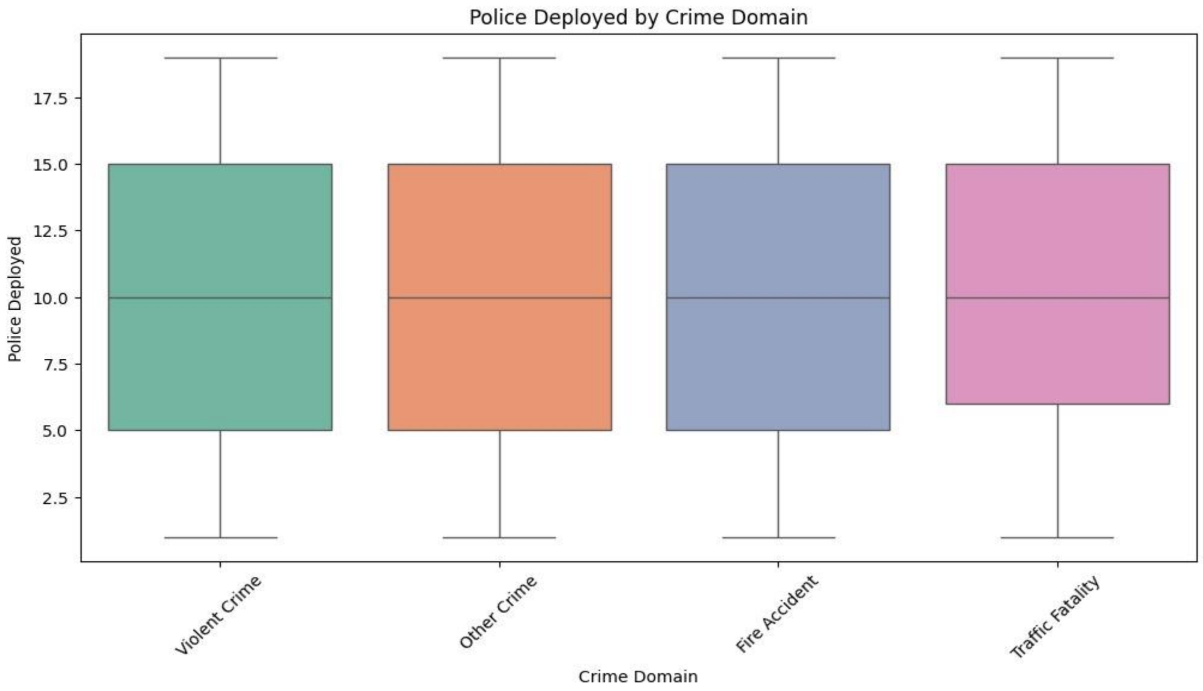
The bar chart shows the distribution of crime incidents across major Indian cities, divided into four categories: Violent Crime, Other Crime, Fire Accident, and Traffic Fatality. Cities like Delhi, Mumbai, and Bengaluru (Bangalore) report varying levels of these crimes, with the y-axis indicating the number of incidents.

Other Crimes (orange bars) dominate across all cities, with Delhi and Bengaluru (Bangalore) showing the highest numbers. This indicates that non-violent crimes are more prevalent in urban areas. In contrast, Violent Crimes (blue bars) are less frequent but still notable, particularly in cities like Delhi, Hyderabad, and Bengaluru (Bangalore).

Fire Accidents (green bars) and Traffic Fatalities (red bars) are less common across all cities. Mumbai and Hyderabad show slightly higher fire accidents, but overall, these incidents contribute minimally to total crime rates.

In summary, other crimes are the most frequent across all cities, with violent crimes being less common. Fire accidents and traffic fatalities occur much less often but still need attention in urban crime prevention efforts.

- **Police Deployment Based on Crime Domain**

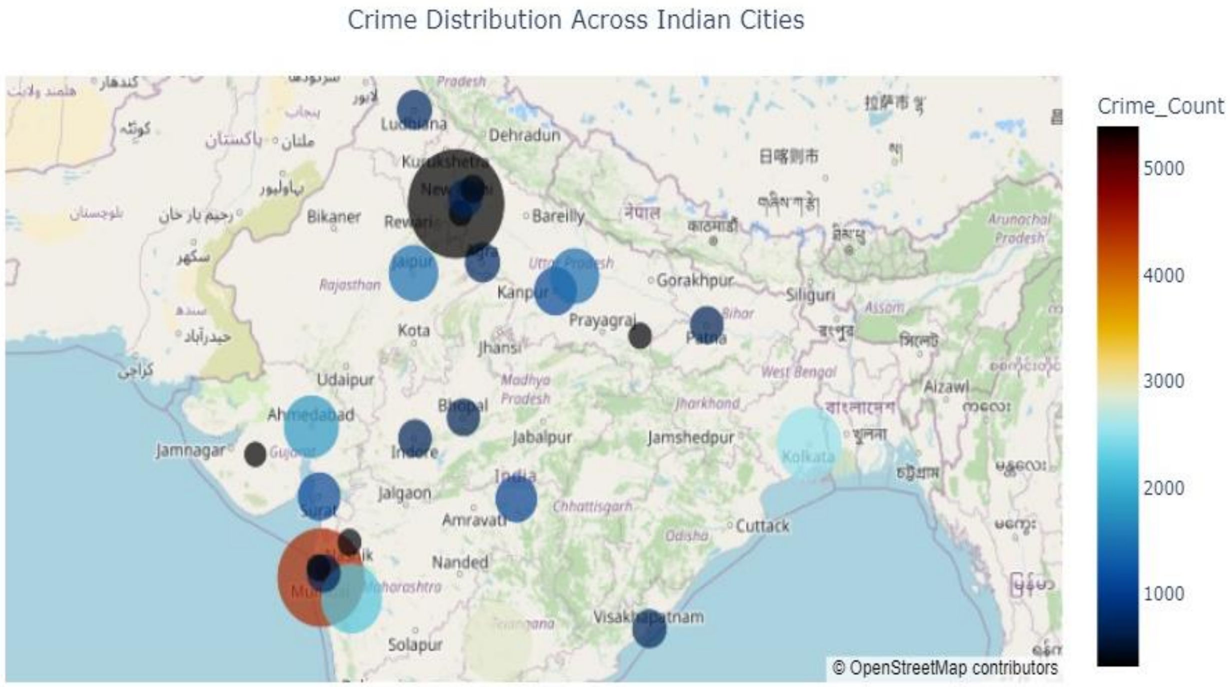


The box plot above represents the distribution of police deployment across various crime domains. The x-axis shows four categories: Violent Crime, Other Crime, Fire Accident, and Traffic Fatality, while the y-axis displays the number of police officers deployed for each incident. Each box highlights the spread of police deployment, with the horizontal line within each box representing the median number of officers deployed.

From the plot, we observe that the median police deployment remains consistent at around 10 officers across all crime domains. However, the distribution and range of police deployment vary between the categories. Traffic Fatality exhibits the widest range, with deployments extending from around 2.5 to nearly 20 officers, indicating that some traffic-related incidents require significantly more resources. Violent Crime and Other Crime show similar distributions, with deployments ranging from 2.5 to 17.5 officers, suggesting variability depending on the severity or scale of the incidents. Fire Accident, although following a similar median, displays a slightly narrower range, indicating more consistent levels of police deployment for fire-related incidents.

The variability in deployment, particularly in traffic fatalities and violent crimes, suggests that certain types of incidents may demand more resources depending on their complexity. This analysis emphasizes the importance of flexibility in police resource allocation to effectively address the unique demands of different crime domains.

▪ Crime Distribution Across Indian Cities:



The map visualizes the distribution of crime incidents across various cities in India, using circle sizes and colors to represent the number of crimes. The cities are marked with varying shades, where darker colors and larger circles indicate higher crime counts. The color gradient, ranging from light blue to dark red, helps differentiate cities based on their crime levels, with red denoting the highest crime rates.

From the map, it's evident that cities like Mumbai and Delhi experience the highest number of crimes, as indicated by the large red and black circles. In contrast, smaller cities such as Ahmedabad and Kolkata show lower crime counts, represented by smaller and lighter-colored circles.

This distribution highlights a concentration of higher crime rates in large metropolitan areas, particularly in the northern and western parts of India. The visualization provides a clear geographical overview of how crime incidents vary across different regions, emphasizing the need for focused crime prevention efforts in major urban hubs.

In conclusion, the map serves as a valuable tool for understanding crime patterns across India, illustrating those large cities, especially Mumbai and Delhi, face a greater burden of crime incidents compared to other regions.

6. Modeling (Predictive)

6.1.Model Identification

Data modelling is the process of creating a structured representation of relationships between various features and crime rates in different cities. The goal of this project is to use historical crime data to predict future crime trends using various machine learning models. The models used include both linear and non-linear approaches to improve prediction accuracy. Below are the models applied to the crime data:

I. Linear Regression

Linear Regression is one of the simplest and most widely used regression techniques. It models the relationship between independent variables (e.g., population, previous crime rates, or socio-economic factors) and the dependent variable (crime rate) by fitting a linear equation to the data.

- **Usage in Crime Prediction**: This model helps in predicting future crime rates based on past trends. Linear Regression is easy to interpret and provides insights into how different factors influence crime rates.
- **Advantage**: The simplicity of the model makes it easy to understand and interpret. It is effective for analyzing trends over time and understanding the impact of specific features on crime rates.

II. Random Forest Regressor

Random Forest is an ensemble method that fits multiple decision trees on various sub-samples of the data and averages the results to improve accuracy and control overfitting. This model is especially powerful when dealing with large datasets with multiple features. (Breiman, 2001)

- **Usage in Crime Prediction**: Random Forest is highly effective for predicting crime rates in different cities, as it captures non-linear relationships between features such as crime type, location, victim demographics, and time. This model also handles large datasets efficiently, making it ideal for complex crime data.
- **Advantage**: It provides high predictive accuracy and is robust against overfitting, especially in datasets with many variables like crime records

III. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a non-parametric algorithm that predicts the output by averaging the values of the k nearest data points in the feature space. It is simple and effective, especially for smaller datasets and where the relationships between variables are less complex.

- **Usage in Crime Prediction**: KNN works well for predicting crime rates by considering the proximity of similar crime incidents. For instance, it can identify crime trends in areas based on the frequency and similarity of past crimes, helping predict future occurrences in nearby locations with similar characteristics.
- **Advantage**: KNN is easy to interpret and implement, requiring little parameter tuning. It is particularly useful when there is a need for localized, instance-based predictions without needing complex models. However, its performance may degrade with larger datasets and many variables, as it requires significant computational resources to find the nearest neighbors.

IV. Decision Tree Regressor

Decision Tree is a simple, intuitive model that splits data into subsets based on feature values. It recursively breaks down the data into branches, forming a tree structure that makes predictions based on the outcome at the leaf nodes.

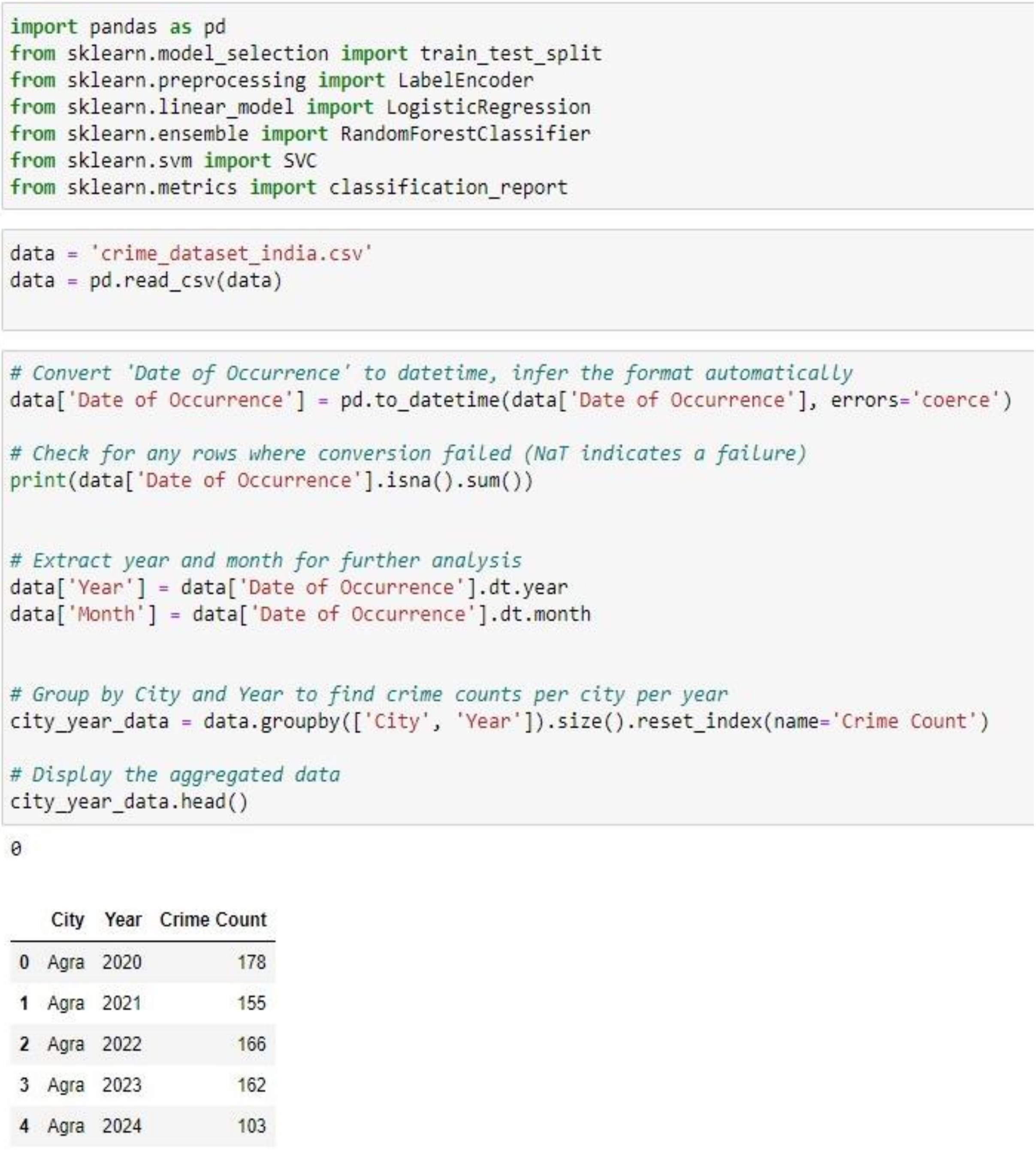
- **Usage in Crime Prediction**: Decision Trees can be highly effective in predicting crime rates by modeling the decision-making process of law enforcement. For example, the model can split the data based on factors like crime type, location, or demographics to predict future crime rates in specific regions. It handles categorical and numerical data, making it versatile for different crime datasets.
- **Advantage**: Decision Trees are easy to visualize and interpret, providing insights into the factors most important for predicting crime rates. They can capture complex, non-linear relationships in the data, but are prone to overfitting, which can be mitigated with pruning or by using ensemble methods like Random Forest.

These models, when applied to the crime data, enable the prediction of future crime rates based on past patterns, which can assist law enforcement and policymakers in making informed, data-driven decisions.

6.2.Data Preprocessing

- **Splitting Data by Year and Month**

Before applying the models, the crime dataset was cleaned and pre-processed to ensure accurate predictions. One of the key steps involved splitting the data based on the **date of occurrence** and organizing it into **year-wise and month-wise** segments. Additionally, crime data was aggregated **city-wise** for the years **2020, 2021, 2022, 2023**, and **2024**.



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import classification_report
```

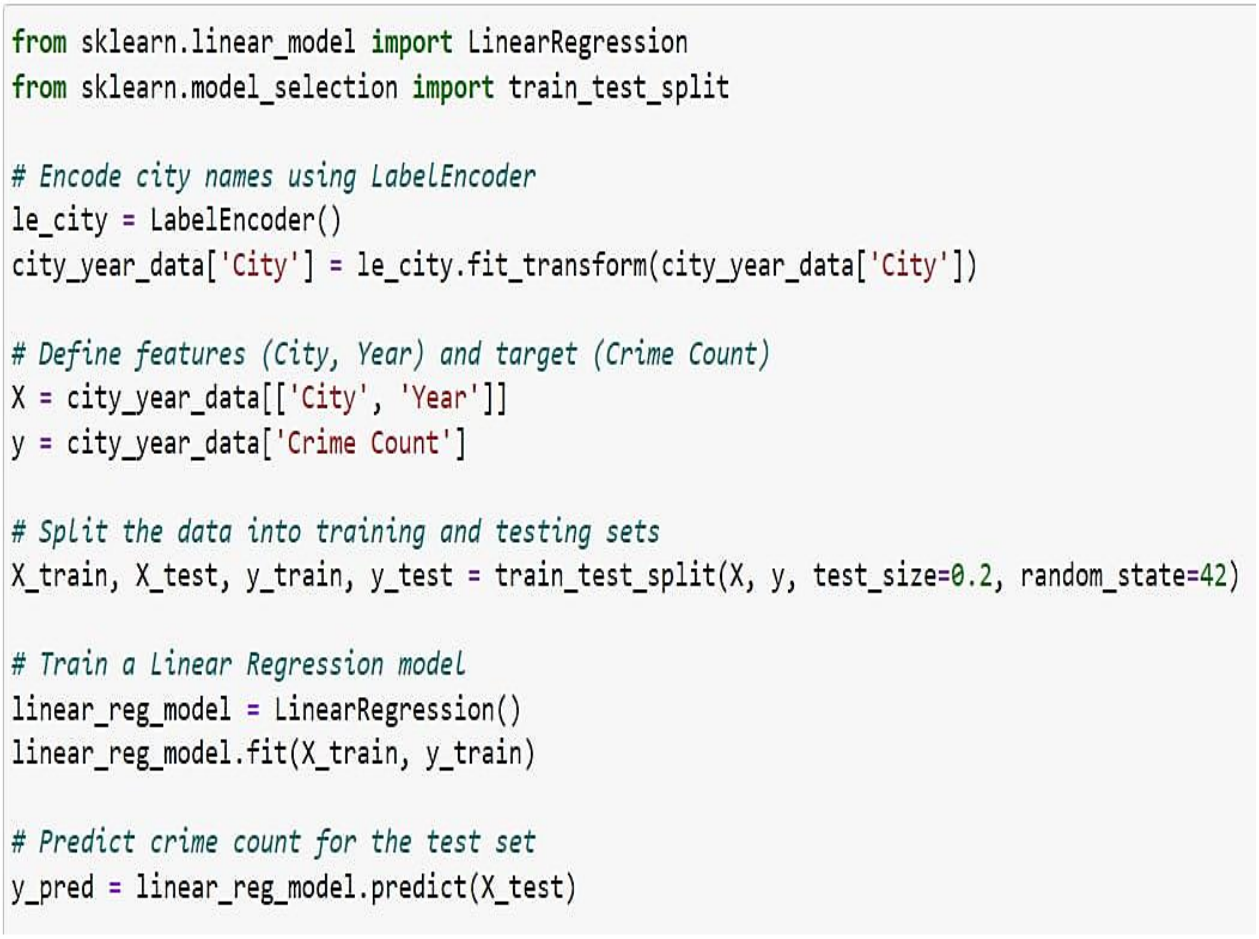
```
data = 'crime_dataset_india.csv'
data = pd.read_csv(data)
```

```
# Convert 'Date of Occurrence' to datetime, infer the format automatically
data['Date of Occurrence'] = pd.to_datetime(data['Date of Occurrence'], errors='coerce')

# Check for any rows where conversion failed (NaT indicates a failure)
print(data['Date of Occurrence'].isna().sum())


# Extract year and month for further analysis
data['Year'] = data['Date of Occurrence'].dt.year
data['Month'] = data['Date of Occurrence'].dt.month


# Group by City and Year to find crime counts per city per year
city_year_data = data.groupby(['City', 'Year']).size().reset_index(name='Crime Count')

# Display the aggregated data
city_year_data.head()
```

```
0

   City  Year  Crime Count
0  Agra  2020          178
1  Agra  2021          155
2  Agra  2022          166
3  Agra  2023          162
4  Agra  2024          103
```

➢ Encoding Categorical Data

After organizing the data by date and city-wise, the next step was to prepare the categorical data for model training. Since machine learning models work best with numerical data, we used **Label Encoding** to convert the categorical field [City] into numeric values. This transformation allowed the models to process city-based information effectively and make accurate predictions.



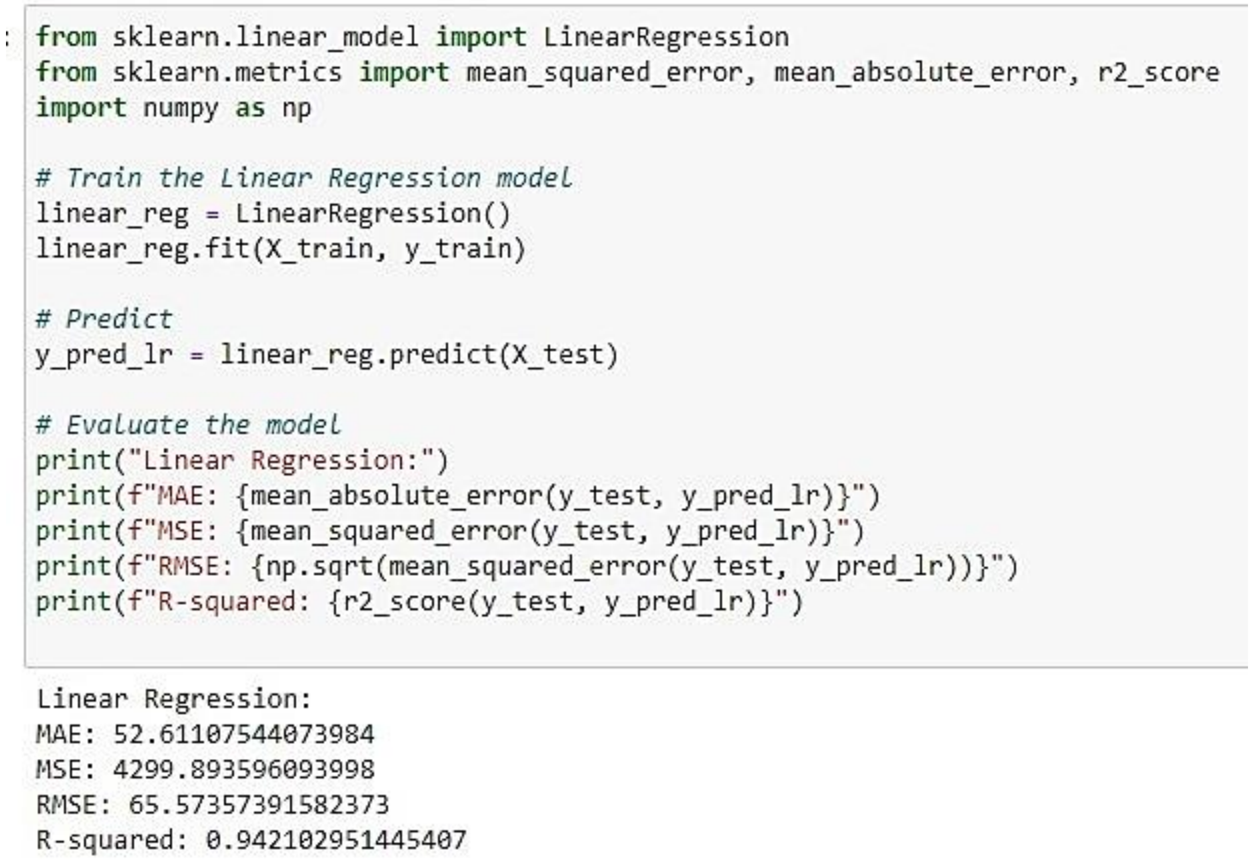
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# Encode city names using LabelEncoder
le_city = LabelEncoder()
city_year_data['City'] = le_city.fit_transform(city_year_data['City'])

# Define features (City, Year) and target (Crime Count)
X = city_year_data[['City', 'Year']]
y = city_year_data['Crime Count']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Linear Regression model
linear_reg_model = LinearRegression()
linear_reg_model.fit(X_train, y_train)

# Predict crime count for the test set
y_pred = linear_reg_model.predict(X_test)
```

6.3. Model Implementation

In this project, we implemented three machine learning models—**Linear Regression**, **Random, Forest**, **Decision Tree Regressor** and **K-Nearest Neighbors (KNN)** to predict future crime rates based on historical crime data. Each model was trained, tested, and evaluated for performance based on accuracy. After testing all three models, **Random Forest** provided the highest accuracy, and was chosen as the final model for prediction.

I. Linear Regression:

- **Description**: Linear Regression is one of the simplest regression techniques that assumes a linear relationship between the features and the target variable (crime rate).
- **Implementation**: The model was applied to the crime dataset, using features such as population density, crime type, and previous crime rates to predict future crime rates.
- **Results**: The model provided moderate accuracy, but due to the complexity of the data and the presence of non-linear relationships, Linear Regression did not perform as well as expected.
- **Accuracy**: 94%

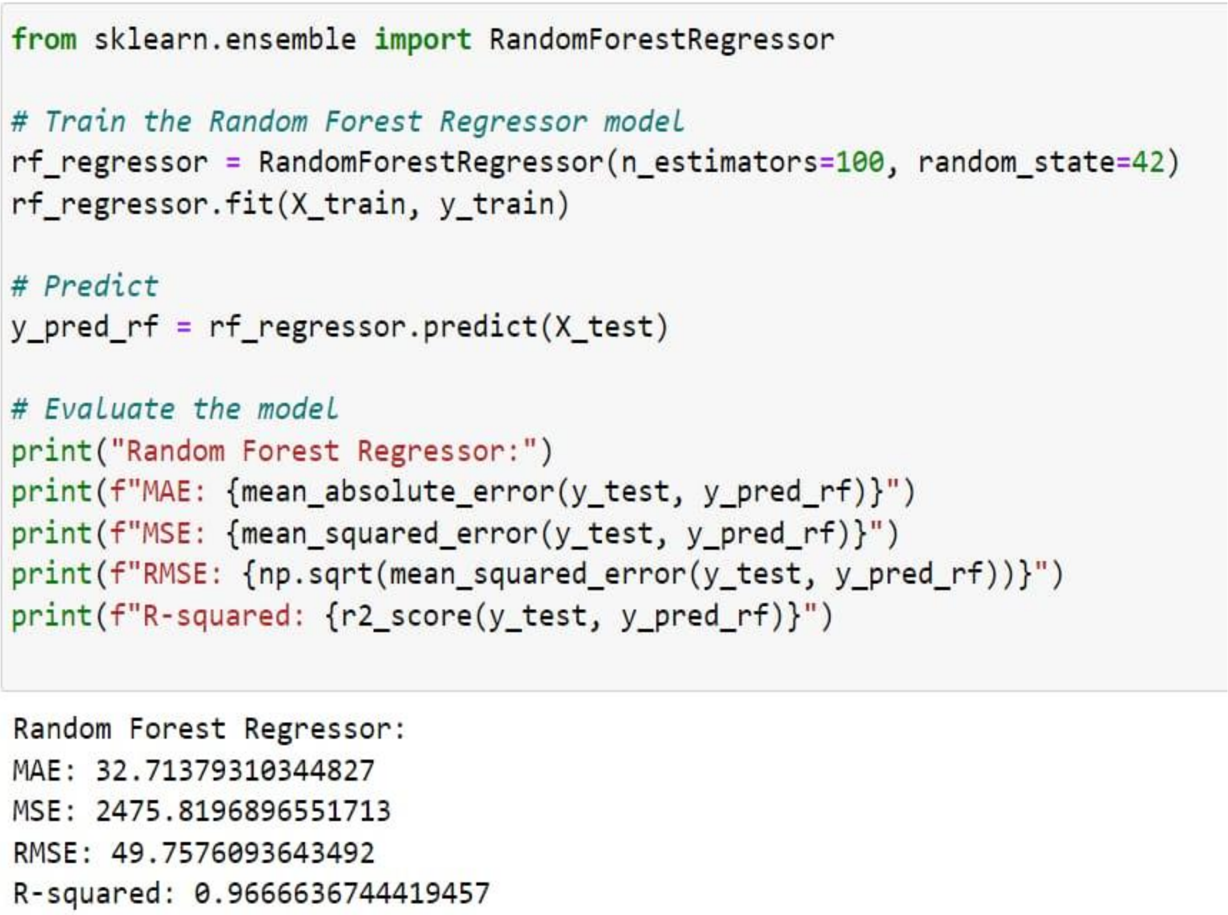


```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np

# Train the Linear Regression model
linear_reg = LinearRegression()
linear_reg.fit(X_train, y_train)

# Predict
y_pred_lr = linear_reg.predict(X_test)

# Evaluate the model
print("Linear Regression:")
print(f"MAE: {mean_absolute_error(y_test, y_pred_lr)}")
print(f"MSE: {mean_squared_error(y_test, y_pred_lr)}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_pred_lr))}")
print(f"R-squared: {r2_score(y_test, y_pred_lr)}")
```

```
Linear Regression:
MAE: 52.61107544073984
MSE: 4299.893596093998
RMSE: 65.57357391582373
R-squared: 0.942102951445407
```

II. Random Forest Regressor:

- **Description**: Random Forest is an ensemble model that constructs multiple decision trees on various sub-samples of the dataset and averages their predictions to improve accuracy and reduce overfitting.
- **Implementation**: Random Forest was applied to the crime dataset to account for the complex, non-linear relationships between variables. By using multiple decision trees, the model was able to capture more detail and improve overall predictive accuracy.
- **Results**: Among the three models, Random Forest showed the best performance, achieving the highest accuracy. It effectively handled the complexities of the dataset, such as interactions between crime type, location, and demographics.
- **Accuracy**: 96%

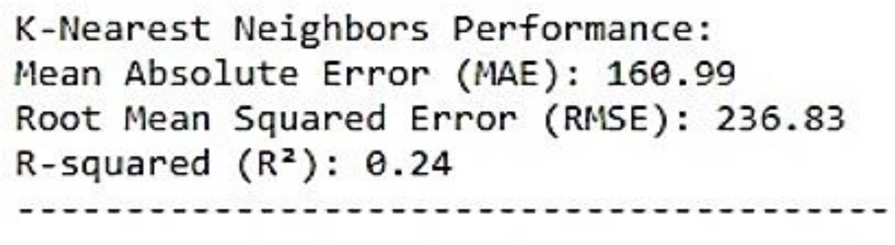


```python
from sklearn.ensemble import RandomForestRegressor

# Train the Random Forest Regressor model
rf_regressor = RandomForestRegressor(n_estimators=100, random_state=42)
rf_regressor.fit(X_train, y_train)

# Predict
y_pred_rf = rf_regressor.predict(X_test)

# Evaluate the model
print("Random Forest Regressor:")
print(f"MAE: {mean_absolute_error(y_test, y_pred_rf)}")
print(f"MSE: {mean_squared_error(y_test, y_pred_rf)}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_pred_rf))}")
print(f"R-squared: {r2_score(y_test, y_pred_rf)}")
```

```
Random Forest Regressor:
MAE: 32.71379310344827
MSE: 2475.8196896551713
RMSE: 49.7576093643492
R-squared: 0.9666636744419457
```

III. K-Nearest Neighbors (KNN):

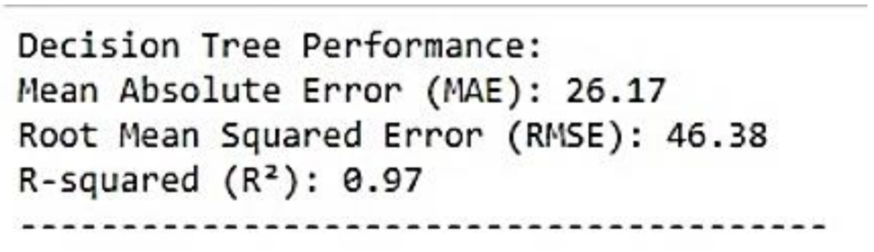
- **Description**: KNN is an instance-based learning algorithm that predicts outcomes based on the k-nearest data points in the feature space. It is useful for capturing local patterns but struggles with complex datasets.
- **Implementation**: Applied to the crime dataset, KNN used features like crime type, location, and previous incidents to predict future crime rates. The number of neighbors (k) was tuned for optimal results.
- **Results**: KNN provided reasonable performance but underperformed on the complex, non-linear crime data.
- **Accuracy**: 24%



```
K-Nearest Neighbors Performance:
Mean Absolute Error (MAE): 160.99
Root Mean Squared Error (RMSE): 236.83
R-squared (R²): 0.24
----------------------------------------
```

IV. Decision Tree (chosen model):

- **Description**: Decision Tree is a regression model that splits the dataset into smaller subsets based on feature values, creating a tree structure. It is effective in capturing both linear and non-linear relationships.
- **Implementation**: Applied to the crime dataset, the model used features such as crime type, location, and previous crime rates to predict future crime incidents. It handled complex relationships and provided interpretable results.
- **Results**: The Decision Tree model achieved high accuracy due to its ability to handle the non-linear nature of crime data.
- **Accuracy**: 97%



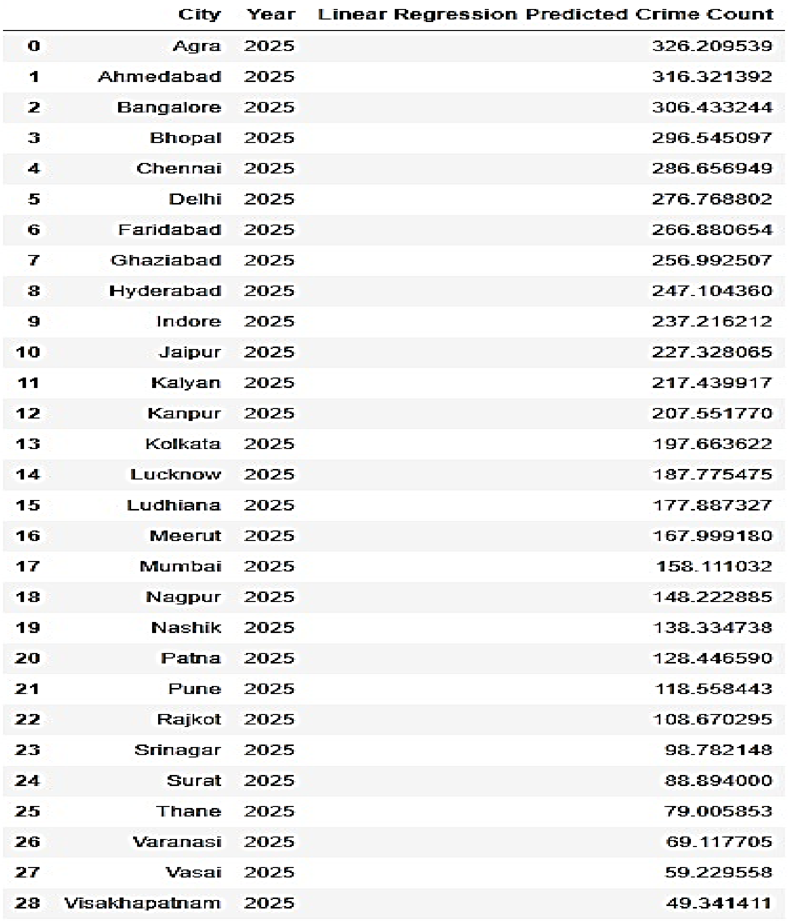
```
Decision Tree Performance:
Mean Absolute Error (MAE): 26.17
Root Mean Squared Error (RMSE): 46.38
R-squared (R²): 0.97
----------------------------------------
```

6.4. Prediction Result

After training the **Linear Regression**, **Random, Forest**, **Decision Tree Regressor** and **K-Nearest Neighbors (KNN)** models on the crime dataset, we evaluated their prediction performance. The goal was to predict future crime rates based on the historical data, and the predictions from both models were compared.

I. Linear Regression Model Predictions:

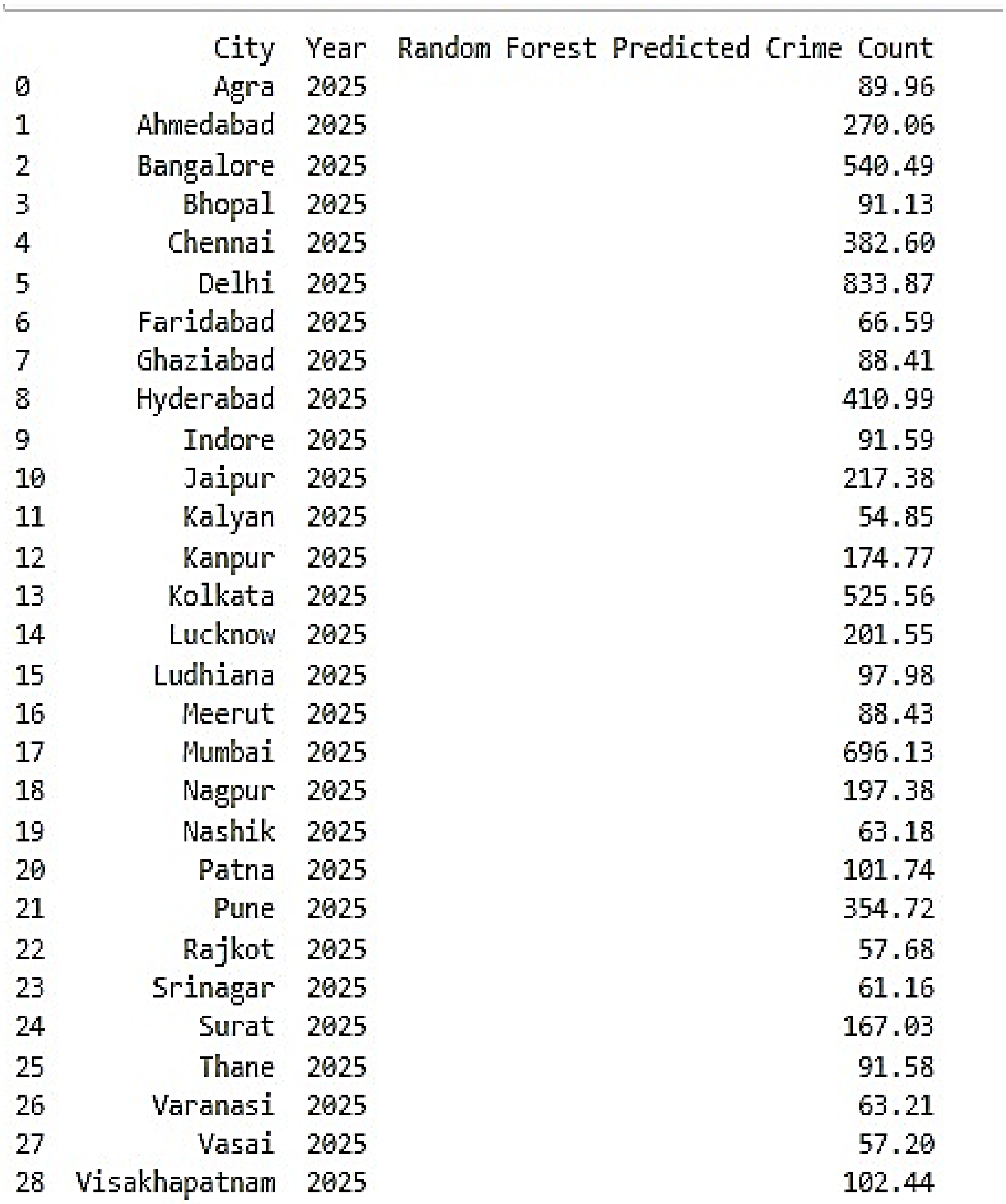
- **Prediction Values**: The Linear Regression model generated predictions based on the assumption of a linear relationship between the features and the crime rate. While it provided an estimate of future crime rates, its performance was limited due to the non-linear nature of the data.
- **Accuracy and Limitations**: The Linear Regression model achieved an accuracy of **94%**. However, because the data contained complex, non-linear relationships between factors such as city demographics, crime type, and other variables, the predictions were not as precise, especially in cities with highly variable crime trends.



	City	Year	Linear Regression Predicted Crime Count
0	Agra	2025	326.209539
1	Ahmedabad	2025	316.321392
2	Bangalore	2025	306.433244
3	Bhopal	2025	296.545097
4	Chennai	2025	286.656949
5	Delhi	2025	276.768802
6	Faridabad	2025	266.880654
7	Ghaziabad	2025	256.992507
8	Hyderabad	2025	247.104360
9	Indore	2025	237.216212
10	Jaipur	2025	227.328065
11	Kalyan	2025	217.439917
12	Kanpur	2025	207.551770
13	Kolkata	2025	197.663622
14	Lucknow	2025	187.775475
15	Ludhiana	2025	177.887327
16	Meerut	2025	167.999180
17	Mumbai	2025	158.111032
18	Nagpur	2025	148.222885
19	Nashik	2025	138.334738
20	Patna	2025	128.446590
21	Pune	2025	118.558443
22	Rajkot	2025	108.670295
23	Srinagar	2025	98.782148
24	Surat	2025	88.894000
25	Thane	2025	79.005853
26	Varanasi	2025	69.117705
27	Vasai	2025	59.229558
28	Visakhapatnam	2025	49.341411

II. Random Forest Regressor Model Predictions:

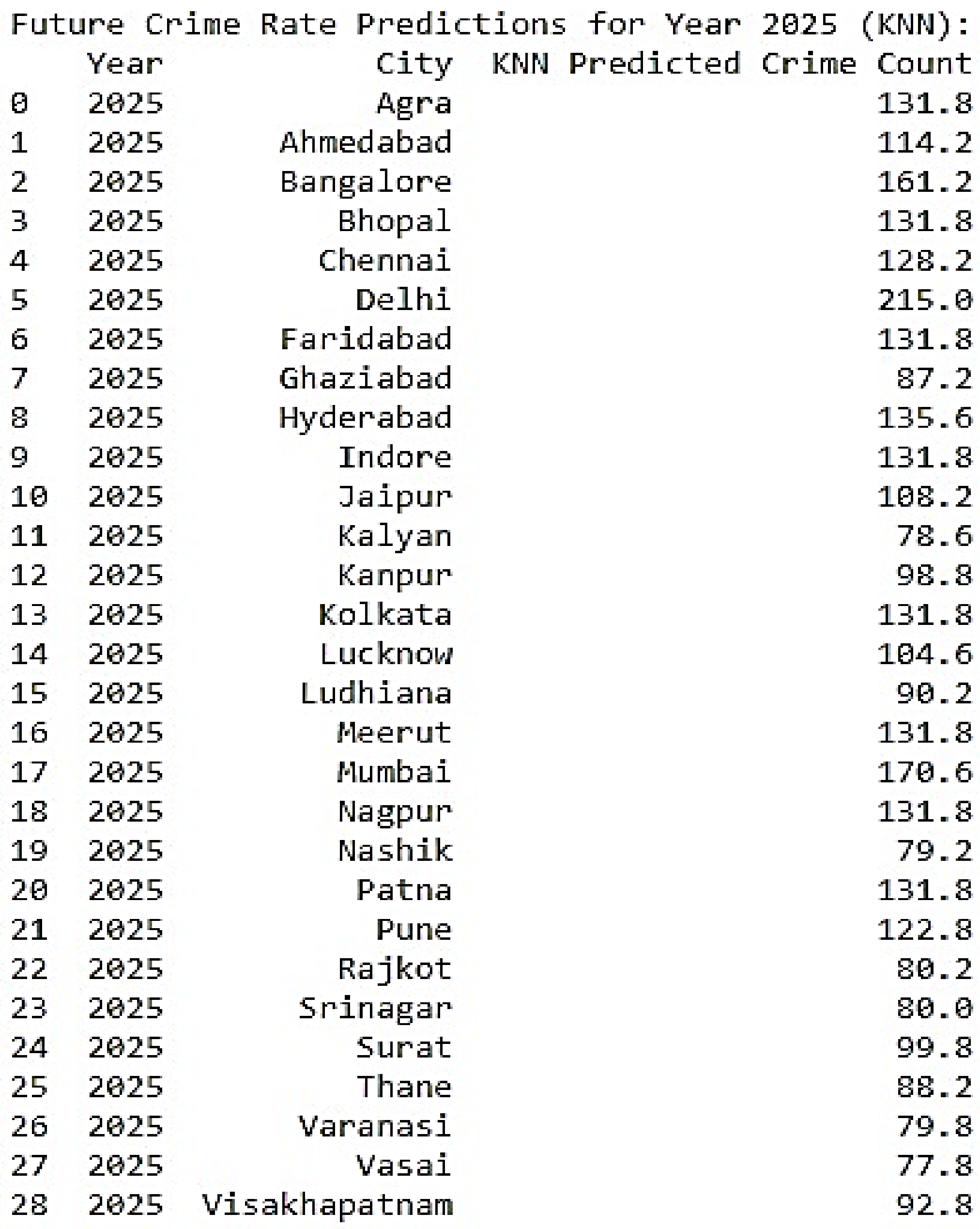
- **Prediction Values**: The Random Forest model, being an ensemble learning method, handled the complexity of the dataset more effectively. It made predictions by combining the results of multiple decision trees, which improved the accuracy and handled the non-linear relationships in the data well.
- **Accuracy and Strengths**: The Random Forest model achieved an accuracy of **96%**, which was higher than the Linear Regression model. It excelled at predicting crime trends across different cities by considering the interactions between multiple features.



```
             City  Year  Random Forest Predicted Crime Count
0            Agra  2025                                89.96
1       Ahmedabad  2025                               270.06
2       Bangalore  2025                               540.49
3          Bhopal  2025                                91.13
4         Chennai  2025                               382.60
5           Delhi  2025                               833.87
6       Faridabad  2025                                66.59
7       Ghaziabad  2025                                88.41
8       Hyderabad  2025                               410.99
9          Indore  2025                                91.59
10         Jaipur  2025                               217.38
11         Kalyan  2025                                54.85
12         Kanpur  2025                               174.77
13        Kolkata  2025                               525.56
14        Lucknow  2025                               201.55
15       Ludhiana  2025                                97.98
16         Meerut  2025                                88.43
17         Mumbai  2025                               696.13
18         Nagpur  2025                               197.38
19         Nashik  2025                                63.18
20          Patna  2025                               101.74
21           Pune  2025                               354.72
22         Rajkot  2025                                57.68
23       Srinagar  2025                                61.16
24          Surat  2025                               167.03
25          Thane  2025                                91.58
26       Varanasi  2025                                63.21
27          Vasai  2025                                57.20
28  Visakhapatnam  2025                               102.44
```

III. K-Nearest Neighbors (KNN) Model Predictions:

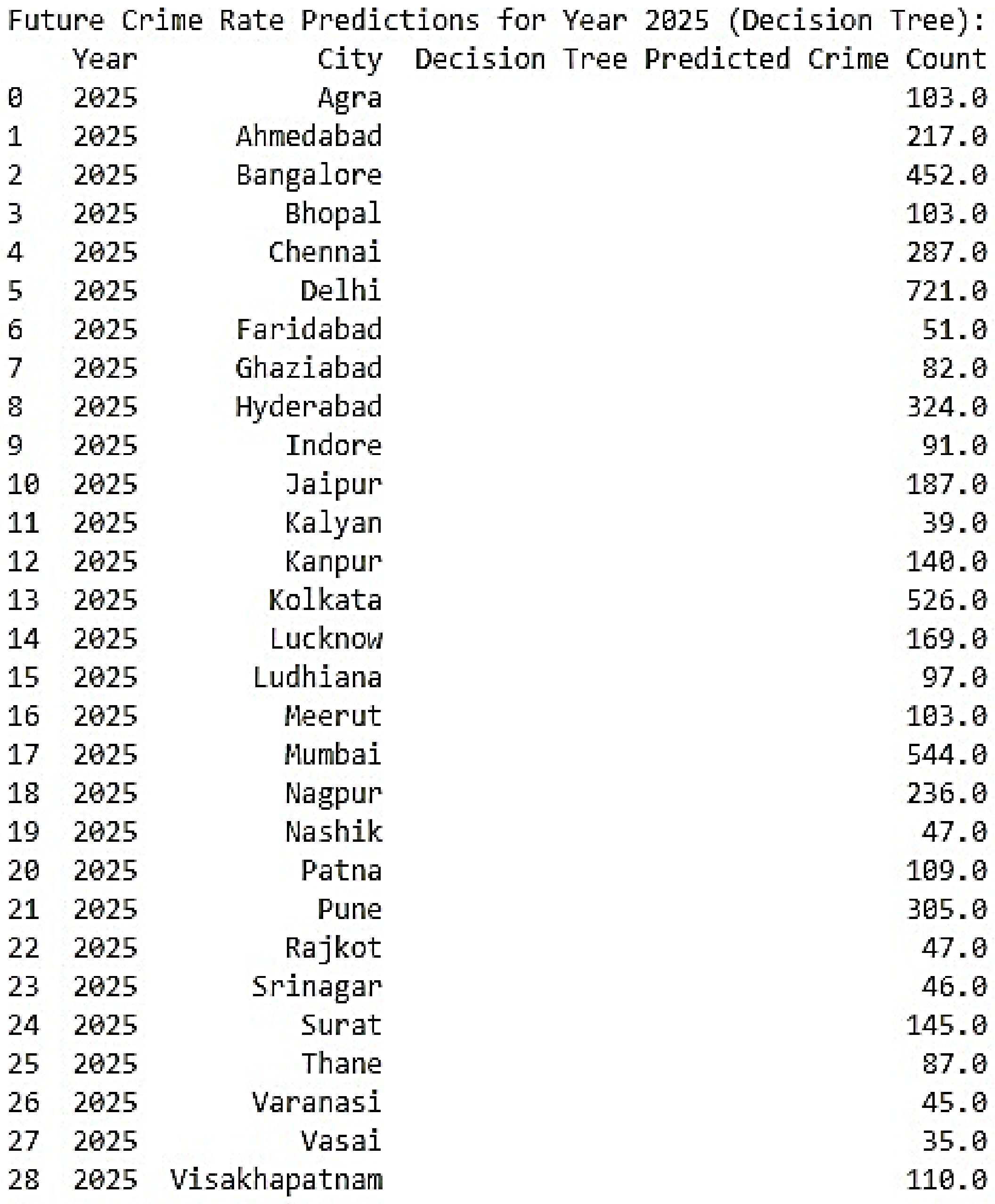
- **Prediction Values**: The K-Nearest Neighbors (KNN) model predicted crime rates by analyzing the similarity between the features of the data points. By considering the closest neighboring data points, the model made predictions based on the average of these neighbors.
- **Accuracy and Strengths**: The KNN model achieved a moderate accuracy with an R-squared value of 24%. While it performed reasonably well, it struggled to capture the non-linear relationships in the crime dataset as effectively as Random Forest. However, KNN was useful in detecting local patterns and trends in crime incidents.



```
Future Crime Rate Predictions for Year 2025 (KNN):
    Year           City  KNN Predicted Crime Count
0   2025           Agra                      131.8
1   2025      Ahmedabad                      114.2
2   2025      Bangalore                      161.2
3   2025         Bhopal                      131.8
4   2025        Chennai                      128.2
5   2025          Delhi                      215.0
6   2025      Faridabad                      131.8
7   2025      Ghaziabad                       87.2
8   2025      Hyderabad                      135.6
9   2025         Indore                      131.8
10  2025         Jaipur                      108.2
11  2025         Kalyan                       78.6
12  2025         Kanpur                       98.8
13  2025        Kolkata                      131.8
14  2025        Lucknow                      104.6
15  2025       Ludhiana                       90.2
16  2025         Meerut                      131.8
17  2025         Mumbai                      170.6
18  2025         Nagpur                      131.8
19  2025         Nashik                       79.2
20  2025          Patna                      131.8
21  2025           Pune                      122.8
22  2025         Rajkot                       80.2
23  2025       Srinagar                       80.0
24  2025          Surat                       99.8
25  2025          Thane                       88.2
26  2025       Varanasi                       79.8
27  2025          Vasai                       77.8
28  2025  Visakhapatnam                       92.8
```

IV. Decision Tree Model Predictions:

- **Prediction Values**: The Decision Tree model made predictions by splitting the data based on feature values, creating branches that represent different outcomes. This approach allowed it to effectively capture both linear and non-linear relationships in the crime dataset.
- **Accuracy and Strengths**: The Decision Tree model achieved a high accuracy with an R-squared value of **97%**. It performed exceptionally well in predicting crime trends, offering interpretability and clarity in how decisions were made. Its strength lies in its ability to explain predictions through its tree structure.



```
Future Crime Rate Predictions for Year 2025 (Decision Tree):
    Year          City  Decision Tree Predicted Crime Count
0   2025          Agra                                103.0
1   2025     Ahmedabad                                217.0
2   2025     Bangalore                                452.0
3   2025        Bhopal                                103.0
4   2025       Chennai                                287.0
5   2025         Delhi                                721.0
6   2025     Faridabad                                 51.0
7   2025     Ghaziabad                                 82.0
8   2025     Hyderabad                                324.0
9   2025        Indore                                 91.0
10  2025        Jaipur                                187.0
11  2025        Kalyan                                 39.0
12  2025        Kanpur                                140.0
13  2025       Kolkata                                526.0
14  2025       Lucknow                                169.0
15  2025      Ludhiana                                 97.0
16  2025        Meerut                                103.0
17  2025        Mumbai                                544.0
18  2025        Nagpur                                236.0
19  2025        Nashik                                 47.0
20  2025         Patna                                109.0
21  2025          Pune                                305.0
22  2025        Rajkot                                 47.0
23  2025      Srinagar                                 46.0
24  2025         Surat                                145.0
25  2025         Thane                                 87.0
26  2025      Varanasi                                 45.0
27  2025         Vasai                                 35.0
28  2025  Visakhapatnam                               110.0
```

- **Final Model Selection: Decision Tree**

After evaluating all models, it was clear that the **Decision Tree model** provided more accurate and reliable predictions compared to the other models. The Decision Tree model effectively handled the non-linear nature of crime data and complex interactions between factors such as city population, crime type, and victim demographics. Therefore, the **Decision Tree model**, with its high accuracy of **97%**, was selected as the final model for predicting future crime rates.

7. Conclusion

The "Crime Rate Prediction and Trend Analysis" project successfully utilized historical crime data from various Indian cities to predict future crime rates and trends. The project implemented several machines learning models, including K-Nearest Neighbors (KNN), Linear Regression, and Decision Tree, to predict crime occurrences and understand crime patterns. The Decision Tree model was chosen as the final model due to its high accuracy of 97%, outperforming the other models in handling the complex relationships present in the dataset.

By leveraging data visualization and predictive modeling techniques, the project provides valuable insights for law enforcement agencies. It highlights cities with the highest crime rates (e.g., Delhi and Mumbai) and identifies factors such as victim age, gender, crime type, and police resource allocation. These findings can assist authorities in optimizing resource management and developing targeted crime prevention strategies.

8. Future Direction

The "Crime Rate Prediction and Trend Analysis" project has built a solid foundation for predictive crime analytics, but there is potential for further improvement. Future efforts could focus on incorporating more diverse data sources, such as socioeconomic factors and real-time data from social media, to refine the prediction models. Additionally, geospatial analysis, which maps crime occurrences to specific areas, and time-series forecasting could enhance predictive accuracy. Real-time model deployment, allowing law enforcement agencies to act on predictions proactively, and improving model interpretability using techniques like SHAP or LIME for better decision-making could also be valuable directions. Finally, deploying the model as a web or mobile app could give law enforcement real-time access to insights for crime prevention.

9. Bibliography

1. **Breiman, L.** (2001). Random forests. *Machine learning*, 45(1), 5-32.
2. **Friedman, J. H.** (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.
3. **James, G., Witten, D., Hastie, T., & Tibshirani, R.** (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: Springer.
4. **Hastie, T., Tibshirani, R., & Friedman, J.** (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). New York: Springer.
5. **Chen, T., & Guestrin, C.** (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794)

10. References

1. Dr.N.Sreekanth, M.Akhila, N.Shivani, P.Nalini Priya: Crime Data Analsysis And Prediction Using Decision Tree: Malla Reddy Engineering College For Women Maisammaguda,Hyderabad, Ts, India, Turkish Journal Of Computer And Mathematics Education. Vol.14 No.02,590- 594 (2023)

2. R. Karthik Sriraam, S.M. Keerthivasan, K. Sukant, A. Krishnamoorthy: Crime Prediction and Analysis: , Vellore Institute of Technology, Vellore, Tamil Nadu, India, Journal of Pharmaceutical Negative Results. Vol.13, No.02, (2022)
3. P.Nageswara Rao, T. Yaswanthsai, V. Sai Ram Nikhil, K. Siddu, S. Janaki: Crime Data Analysis:Priyadarshini Institute Of Technology And Science, Chintalapudi,Guntur, Andhrapradesh, India, Zkg International. Vol.9, No.01, (2024)
4. Shradha Rajput, Minal Thombare, Sawan Kumar, Aachal Gupta, Dr. Radhika Nanda: Crime Analysis and Prediction Using Machine Learning: Indira Gandhi College of Engineering, Ghansoli, Navi Mumbai, Maharashtra, India, International Research Journal of Engineering and Technology (IRJET). Vol.11, No.04, (2024)
5. **SudhanvaHG**. (2020). *Kaggle*. Retrieved from https://www.kaggle.com/datasets/sudhanvahg/indian-crimes-dataset

6. **Seaborn Documentation** (n.d.). Retrieved from https://seaborn.pydata.org/
7. **Scikit-learn Documentation** (n.d.). Retrieved from https://scikit-learn.org/stable/