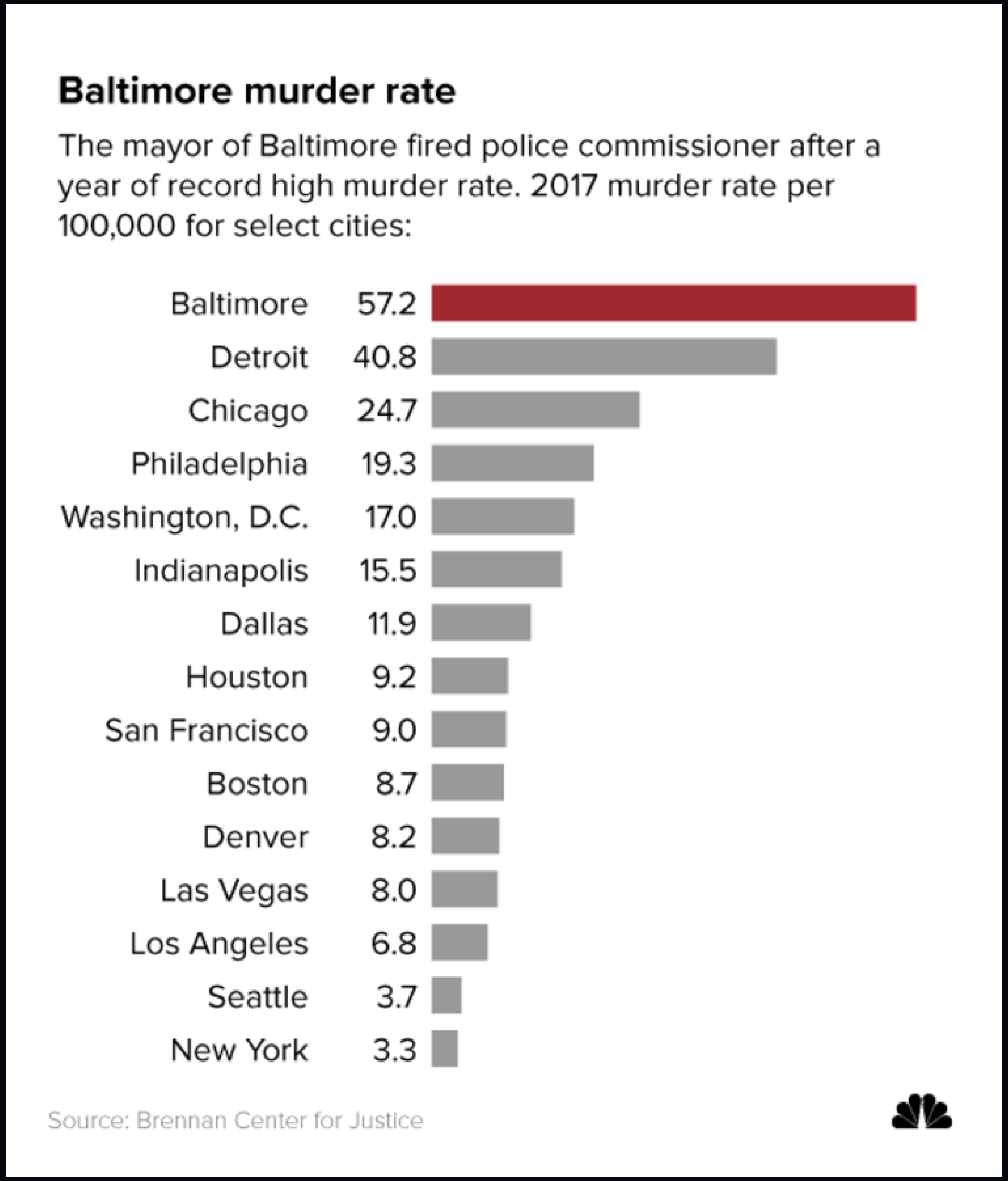
PREDICTING CRIME HOTSPOTS AND ANALYZING CRIME TRENDS IN BALTIMORE

A DATA-DRIVEN APPROACH TO IMPROVE RESOURCE ALLOCATION FOR CRIME PREVENTION.

BY
CHATLA VIVEK
LY42095

Understanding Baltimore's Crime Challenges:



Objective: → To predict high-risk crime areas and analyze crime trends in Baltimore.

Significance: → Provides insights for Baltimore city authorities to allocate resources effectively.Supports crime prevention and quicker response to incidents.

Impact: → Enhanced public safety and optimized resource distribution.

Vivek Chatla

Data Overview

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 638033 entries, 0 to 638032
Data columns (total 23 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   RowID            638033 non-null  int64
 1   CCNumber         638033 non-null  object
 2   CrimeDateTime    638033 non-null  object
 3   CrimeCode        638033 non-null  object
 4   Description      638033 non-null  object
 5   Inside_Outside   37723 non-null   object
 6   Weapon           166723 non-null  object
 7   Post             629894 non-null  object
 8   Gender           536109 non-null  object
 9   Age              515351 non-null  float64
 10  Race             607177 non-null  object
 11  Ethnicity        110769 non-null  object
 12  Location         634312 non-null  object
 13  Old_District     566413 non-null  object
 14  New_District     63563 non-null   object
 15  Neighborhood     629097 non-null  object
 16  Latitude         636700 non-null  float64
 17  Longitude        636700 non-null  float64
 18  GeoLocation      638033 non-null  object
 19  PremiseType      586168 non-null  object
 20  Total_Incidents  638033 non-null  int64
 21  x                636700 non-null  float64
 22  y                636700 non-null  float64
dtypes: float64(5), int64(2), object(16)
memory usage: 112.0+ MB
```

Source:

Baltimore Open Data Portal.

Dataset Description:

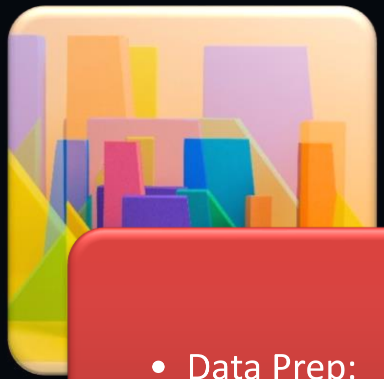
•638,033 records and 23 features.

•Key columns include Crime Type, Location (Latitude & Longitude), Date/Time, Neighborhood, and Description.

Preprocessing:

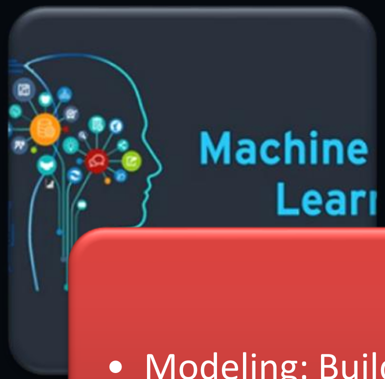
Addressed missing values in critical columns.Converted CrimeDateTime for temporal analysis.Prepared geospatial data for hotspot mapping.

Methodology



- Data Prep: Collect, clean, and preprocess data.

- EDA: Analyze patterns using visualizations.



- Modeling: Build and evaluate ML models.

- Insights: Visualize findings in dashboards.

- Recommendations: Propose strategies for prevention.

Data Cleaning

->Dropped unnecessary columns.

```
drop_columns = ['x','y','RowID','CCNumber','CrimeCode','Post','Location','Neighborhood','New_District','GeoLocation','Total_Incidents','PremiseType']
df1 = df.drop(drop_columns,axis=1)
df1.head()
```

	CrimeDateTime	Description	Inside_Outside	Weapon	Gender	Age	Race	Ethnicity	Old_District	Latitude	Longitude
0	4/14/2015 3:20:00 PM	COMMON ASSAULT	NaN	NaN	M	15.0	BLACK_OR_AFRICAN_AMERICAN	NaN	NORTHEAST	39.331378	-76.580758
1	4/14/2015 3:00:00 PM	LARCENY	NaN	NaN	F	37.0	BLACK_OR_AFRICAN_AMERICAN	NaN	WESTERN	39.318693	-76.654400
2	4/14/2015 1:30:00 PM	AUTO THEFT	NaN	NaN	F	54.0	BLACK_OR_AFRICAN_AMERICAN	NaN	NORTHERN	39.355620	-76.609304
3	4/14/2015 3:07:00 AM	LARCENY FROM AUTO	NaN	NaN	F	33.0	BLACK_OR_AFRICAN_AMERICAN	NaN	NORTHEAST	39.324435	-76.560800
4	4/14/2015 10:10:00 PM	BURGLARY	NaN	NaN	F	26.0	BLACK_OR_AFRICAN_AMERICAN	NaN	EASTERN	39.311000	-76.611729

->Handled missing values

```
[15]: df1.isnull().sum().map(lambda r:(r/len(df1))*100)
```

```
[15]: CrimeDateTime     0.000000
      Description       0.000000
      Gender           15.974722
      Age              19.228159
      Race              4.836113
      Old_District     11.225125
      Latitude          0.208923
      Longitude         0.208923
      dtype: float64
```

->Dropped rows with latitude and longitude with values ‘0’.

[18]:
```
df1[(df1.Longitude == 0) & (df1.Latitude == 0) ]
```

[18]:

	CrimeDateTime	Description	Gender	Age	Race	Old_District	Latitude	Longitude
41	9/7/2013 5:30:00 PM	AUTO THEFT	F	49.0	BLACK_OR_AFRICAN_AMERICAN	NaN	0.0	0.0
220	7/16/2013 10:30:00 PM	AGG. ASSAULT	F	24.0	UNKNOWN	NaN	0.0	0.0
281	4/8/2015 7:30:00 PM	AGG. ASSAULT	M	47.0	BLACK_OR_AFRICAN_AMERICAN	NaN	0.0	0.0
387	8/30/2013 5:00:00 PM	LARCENY FROM AUTO	F	44.0	BLACK_OR_AFRICAN_AMERICAN	NaN	0.0	0.0
467	4/2/2015 9:58:00 PM	AGG. ASSAULT	F	37.0	BLACK_OR_AFRICAN_AMERICAN	NaN	0.0	0.0
...	...	...	...	...	...	...	...	...
637102	10/13/2013 2:00:00 PM	COMMON ASSAULT	F	17.0	UNKNOWN	NaN	0.0	0.0
637287	11/15/2015 7:30:00 PM	AUTO THEFT	F	27.0	BLACK_OR_AFRICAN_AMERICAN	NaN	0.0	0.0
637332	9/13/2013 9:54:00 PM	AGG. ASSAULT	F	29.0	BLACK_OR_AFRICAN_AMERICAN	NaN	0.0	0.0
637452	10/30/2013 2:00:00 PM	LARCENY	NaN	NaN	UNKNOWN	NaN	0.0	0.0
637700	10/14/2013 11:10:00 AM	LARCENY	M	NaN	BLACK_OR_AFRICAN_AMERICAN	NaN	0.0	0.0

6622 rows × 8 columns

-> Checked categorical values for any anamolies

[44]:
```
df1.Gender.value_counts()
```

[44]:
```
Gender
F    268336
M    237225
U      1317
Name: count, dtype: int64
```

There are lot of classes in the variable 'Gender'. Converting all of them into 3 categories 'M','F','U'.

[46]:
```
def gend_conv(value):
    if value in ['Male','M']:
        return 'M'
    elif value in ['Female','F']:
        return 'F'
    else:
        return 'U'
```

[47]:
```
df1.Gender = df1.Gender.apply(gend_conv)
df1.Gender.value_counts()
```

[47]:
```
Gender
F    268336
M    237225
U      1317
Name: count, dtype: int64
```

Exploratory Data Analysis (EDA)

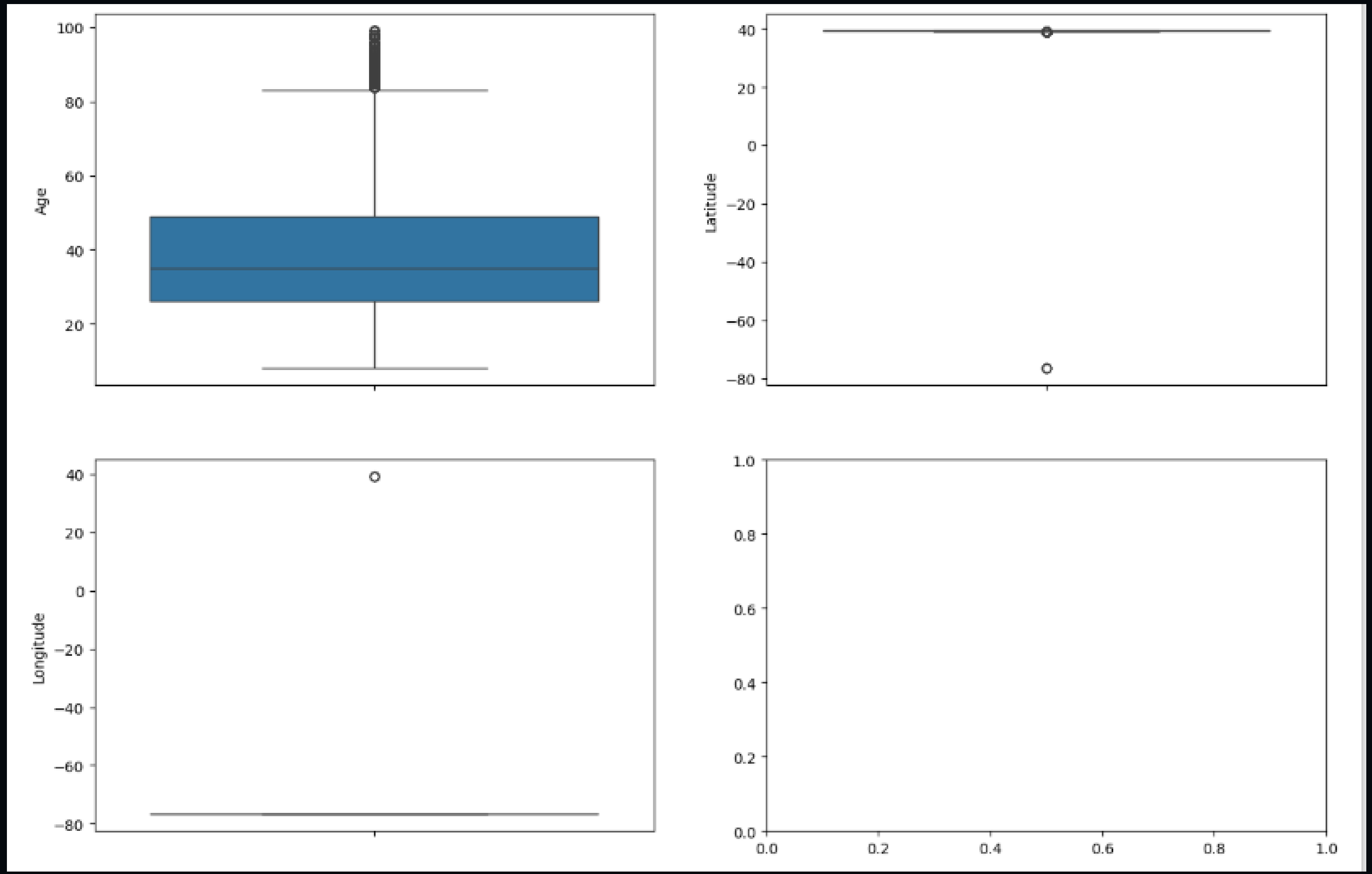
Objective of EDA:

•To uncover trends in Baltimore crime data, focusing on spatial, temporal, and demographic factors.

•To identify patterns that could inform predictive modeling and resource allocation.

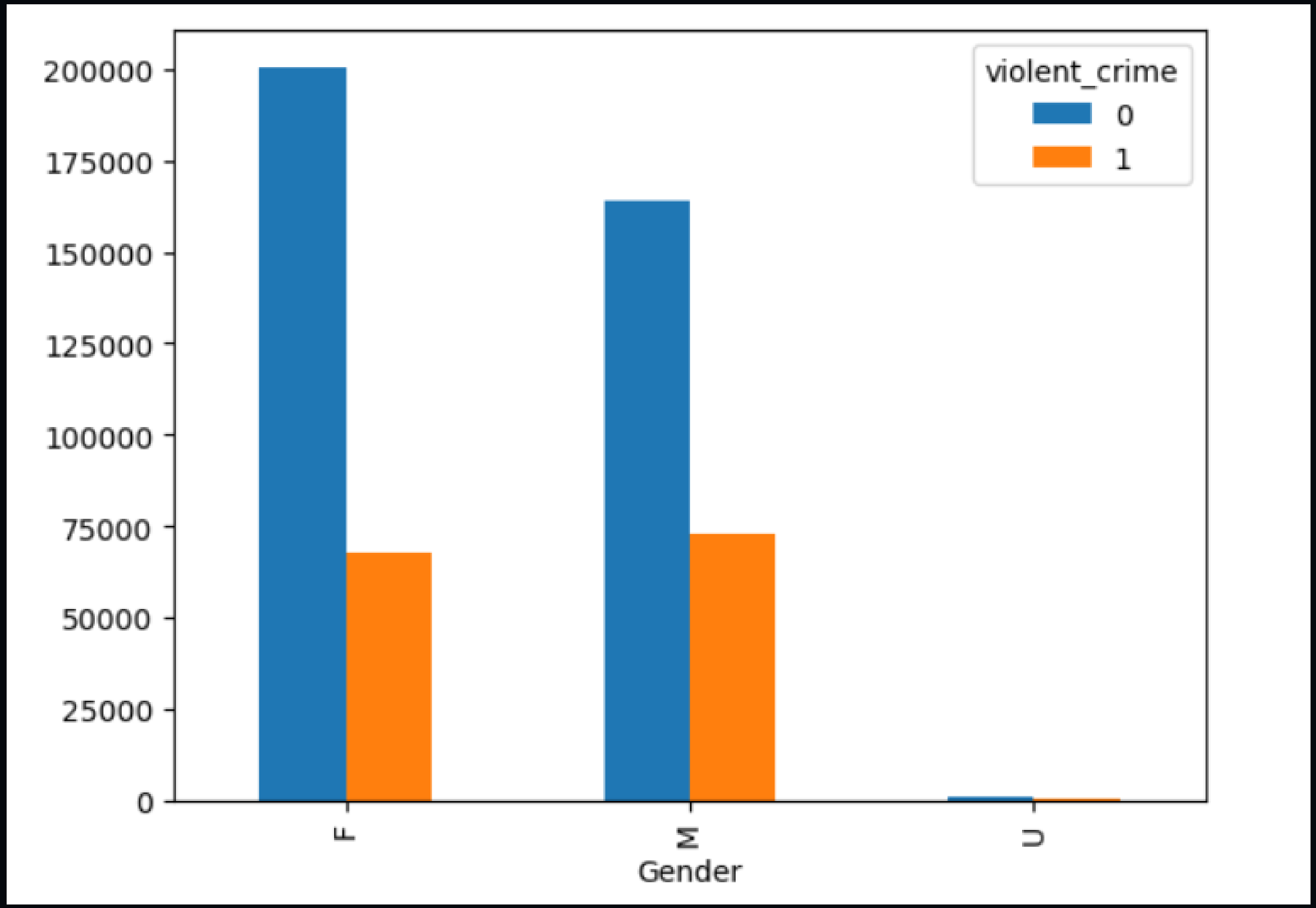
Key EDA Steps:

- Box Plots for Numerical Features:
- Visualized distributions of numerical variables to detect outliers and data variability.
- Example: Distribution of age-related patterns.



Cross-Tabulation Analysis:

Explored relationships between categorical variables, such as gender and crime type. Created bar charts to compare violent crimes by gender.



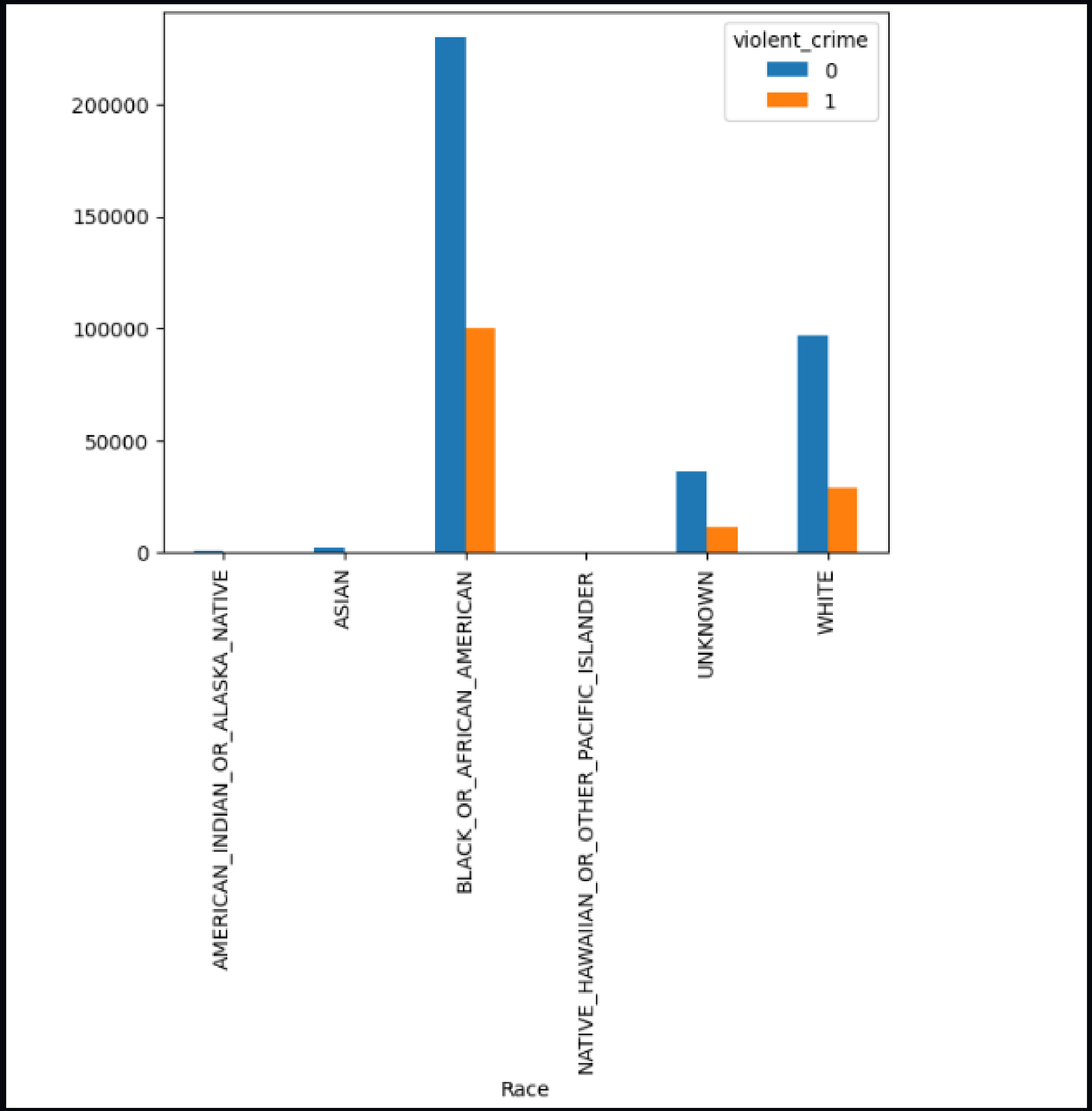
Bar Plots for Demographic Insights:

Analyzed relationships between race and violent crimes. Highlighted demographic trends in crime data.

Black or African American:
This group has the highest frequency of both violent and non-violent crimes compared to other racial categories.

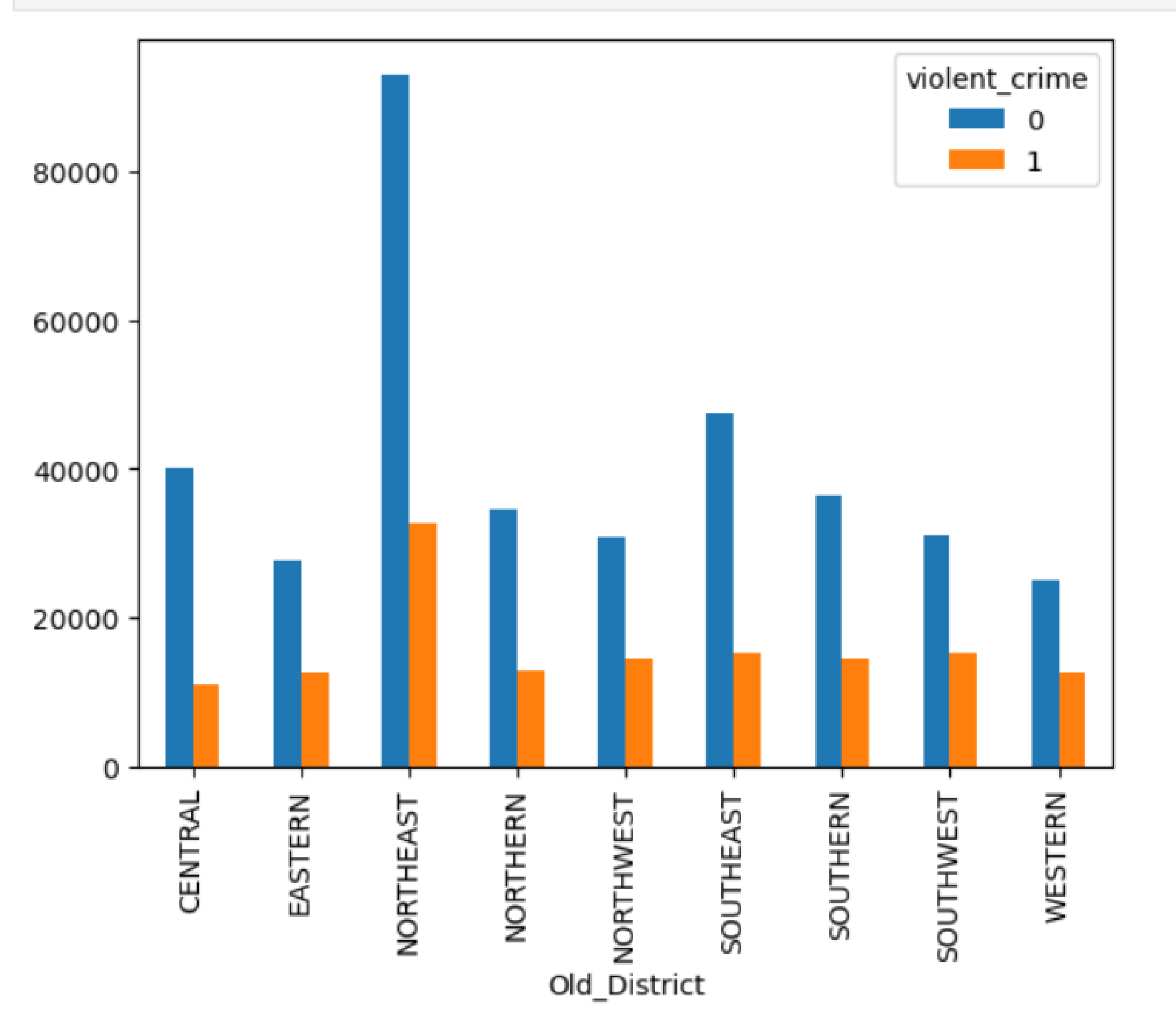
White:

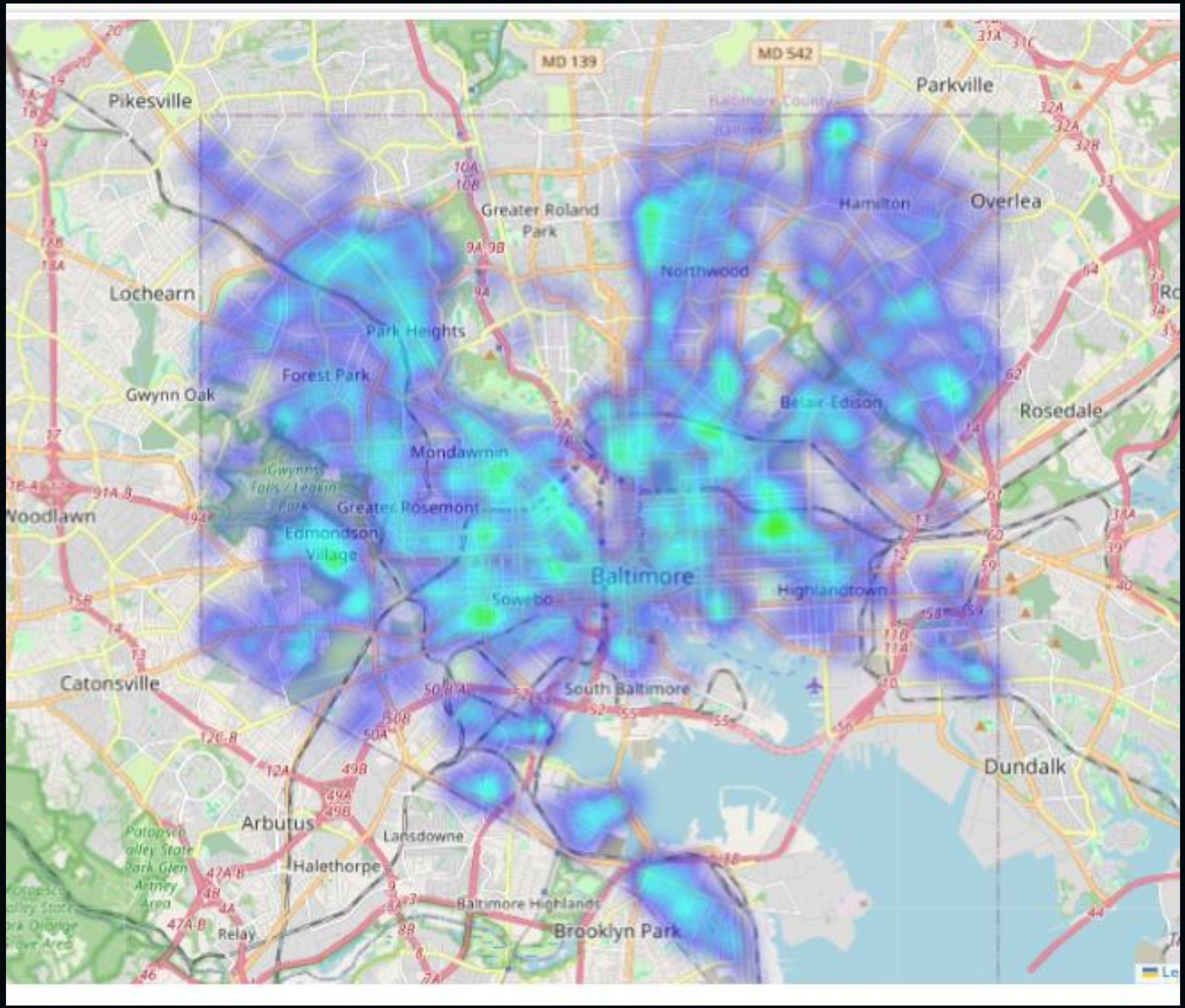
A significant number of crimes are recorded for this group, but the proportion of violent to non-violent crimes is lower compared to the "Black or African American" category.



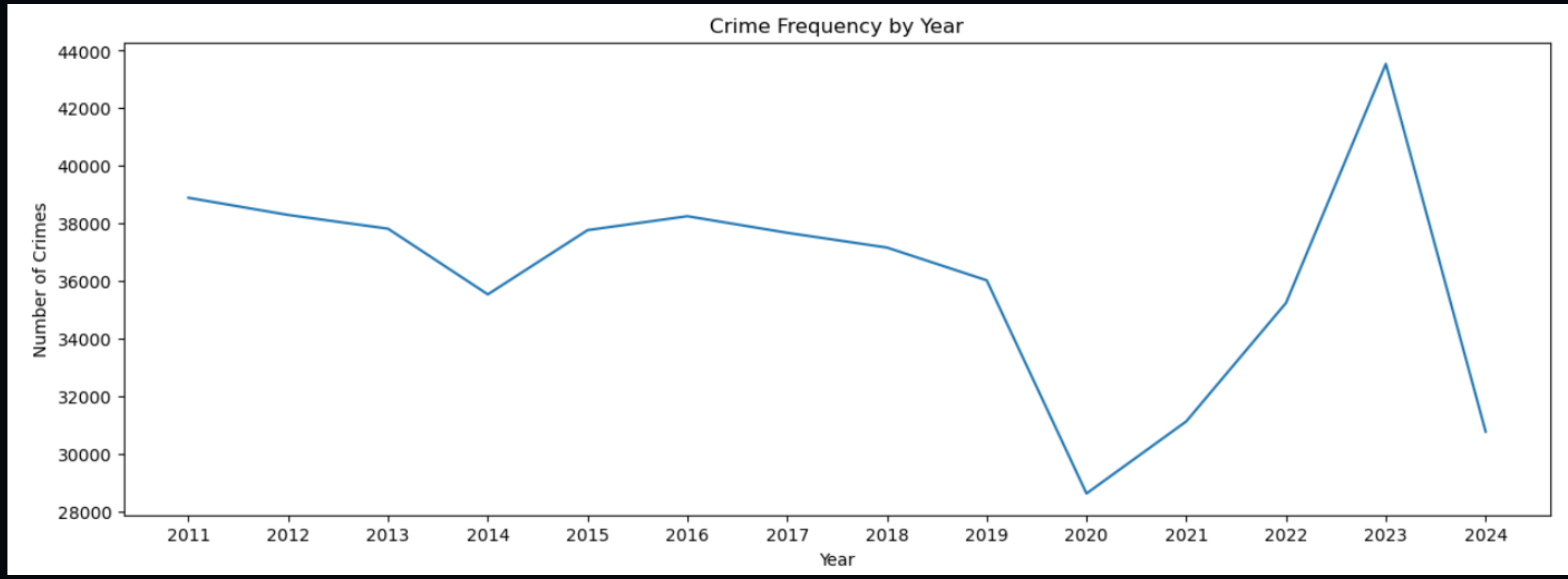
Geospatial Visualization:

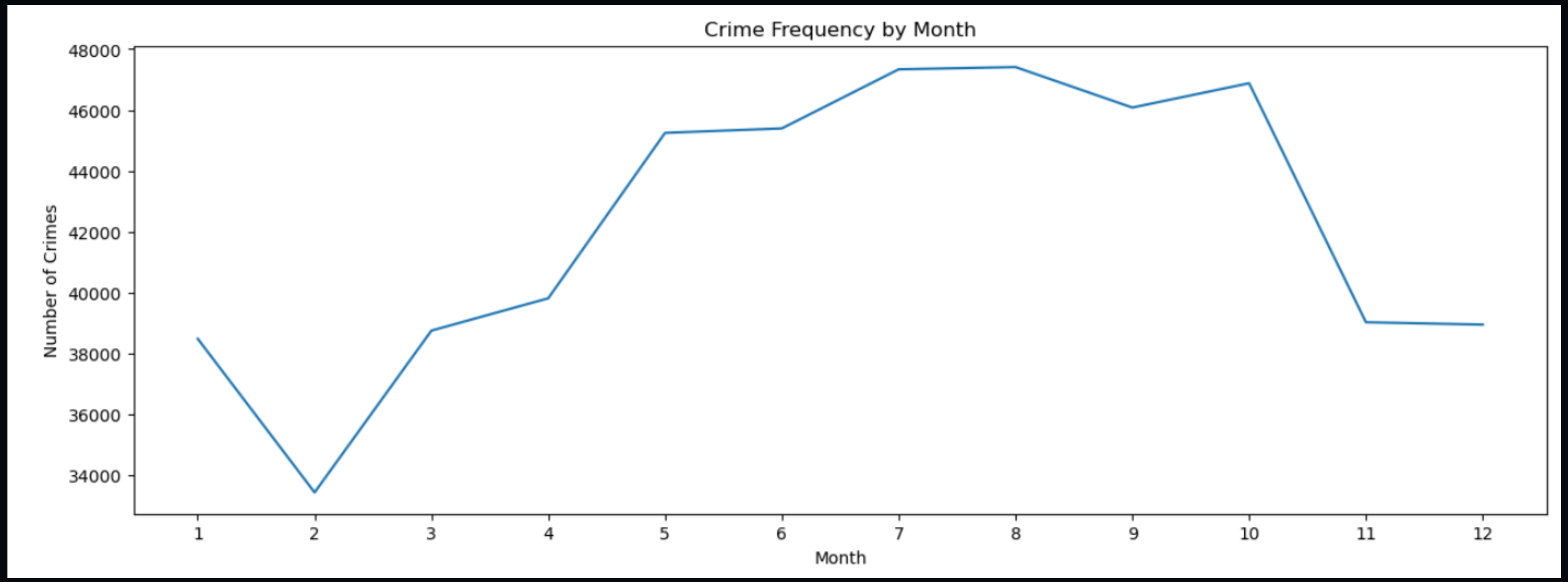
Highlighted crime occurrences using latitude and longitude.





Crime Trend Analysis



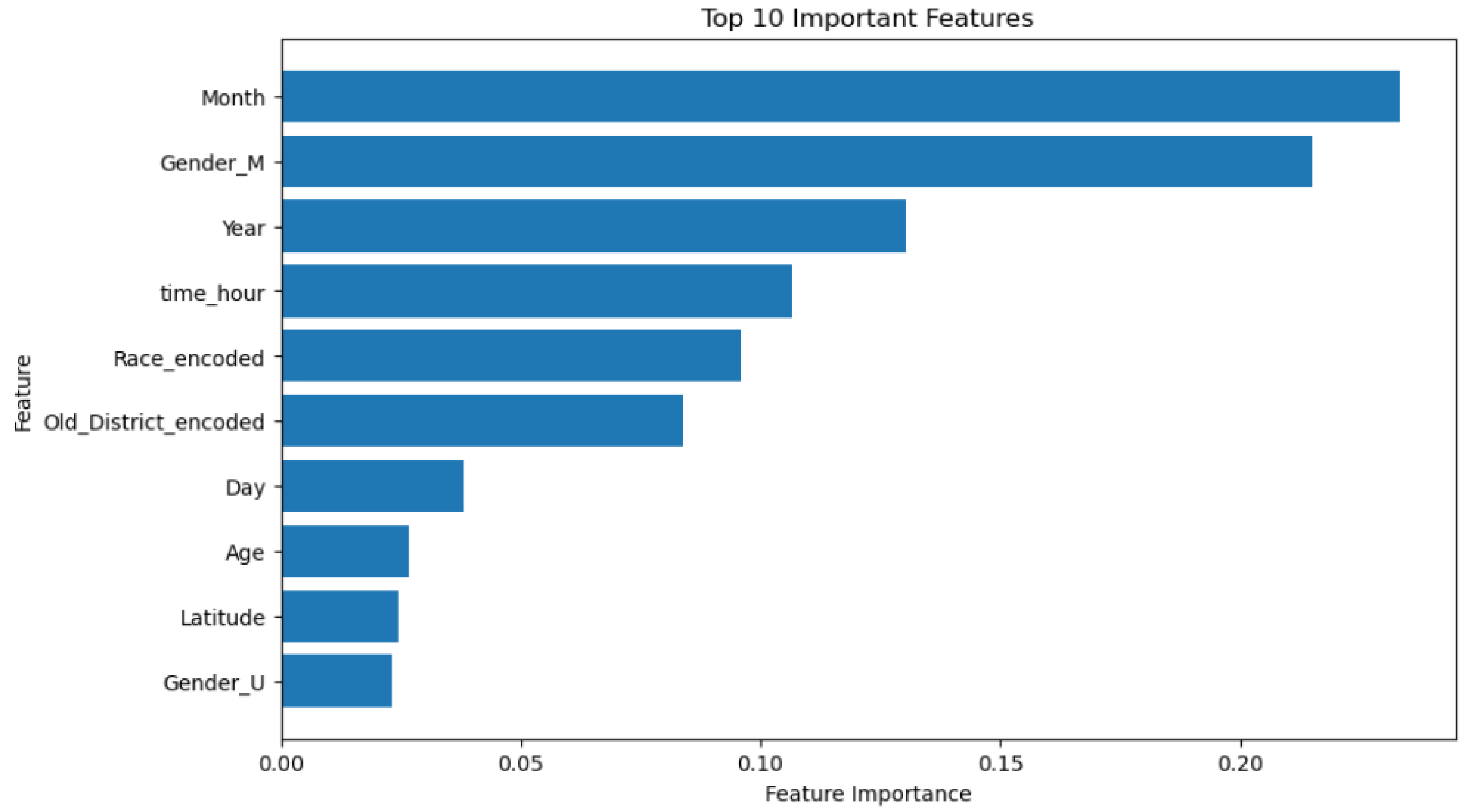


Crime Frequency by Year:
•**Trend Analysis**: Crime counts generally fluctuate over the years with notable dips and peaks.
•**Noteworthy Observations**:

- There was a sharp decline around 2020-2021, which might correlate with external factors such as the COVID-19 pandemic and its societal impact.
- A significant increase occurred after 2021, reaching a peak in 2023 before dropping sharply in 2024.

Crime Frequency by Month:
Seasonal Trends: Crimes tend to be lower in the early months of the year (January and February) and higher during the mid-year months (June to August).There's a noticeable decrease in crimes towards the end of the year (November and December).

Feature Importance



	Feature	Importance
4	Month	0.233146
7	Gender_M	0.214966
3	Year	0.130176
6	time_hour	0.106504
10	Race_encoded	0.095932
9	Old_District_encoded	0.083907
5	Day	0.038012
0	Age	0.026828
1	Latitude	0.024454
8	Gender_U	0.023206
2	Longitude	0.022869

->Used Label encoder for categorical columns

Using label encoder for the columns 'Old_District' and 'Race' as there are more categories

[88]:
```
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
```

[89]:
```
df1['Old_District_encoded'] = encoder.fit_transform(df1['Old_District'])
df1['Race_encoded'] = encoder.fit_transform(df1['Race'])
```

[90]:
```
df1.head()
```

Machine Learning Models

- **Models Implemented:**

- Logistic Regression
- Random Forest
- XGBoost
- Ada Boost
- KNN
- Gradient Boosting

Family	Models	Key Characteristics
Generalized Linear Models	Logistic Regression	Linear relationships, interpretable.
Decision Trees	Random Forest, Gradient Boosting, XGBoost, AdaBoost	Handle non-linear patterns, ensemble-based improvements.
Lazy Learning Models	K-Nearest Neighbors	Instance-based, no explicit training phase.

Comparison Summary

Metric	Focus	When to Prioritize
Accuracy	Overall correctness	Balanced datasets.
Precision	Correctness of positive predictions	Cost of false positives is high.
Recall	Identifying actual positives correctly	Cost of false negatives is high.
F1 Score	Balance between precision and recall	Imbalanced datasets.
ROC AUC	Distinguishing between classes	Evaluate ranking or threshold flexibility.

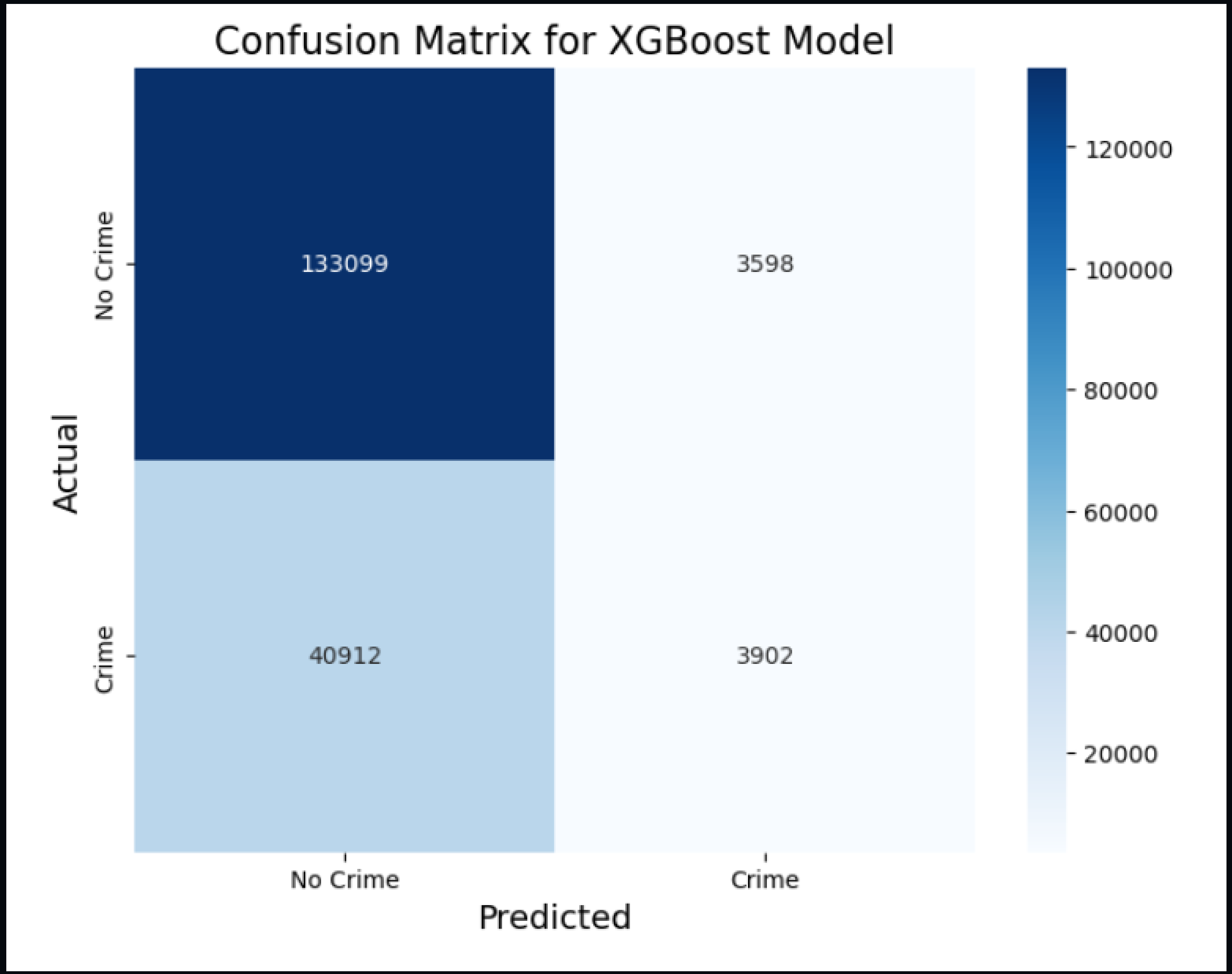
It can be seen that Xgboost model metrics are quite good when compared to other models.

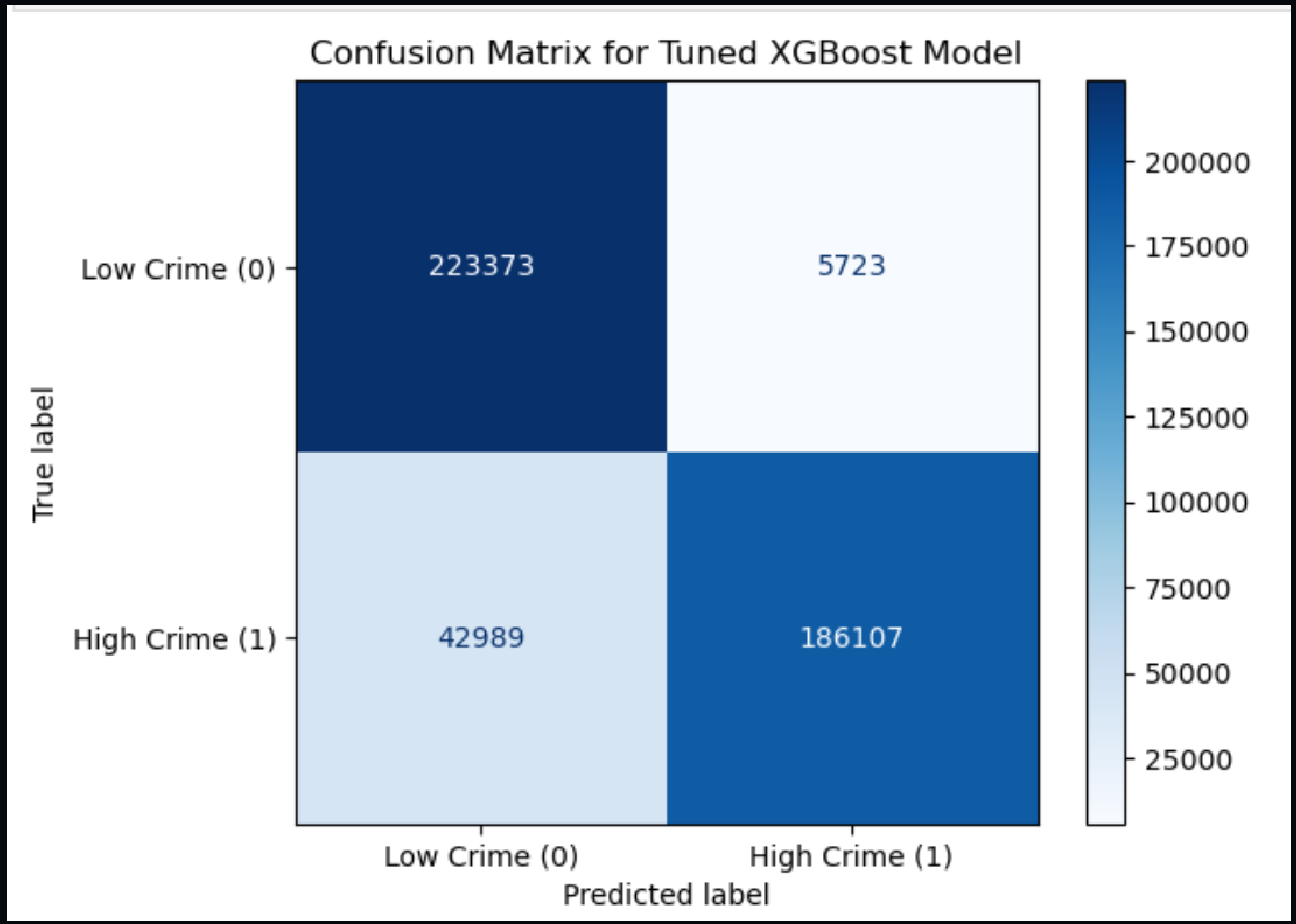
	Model_name	Accuracy_Score	Precision_Score	Recall_Score	F1 Score	ROC AUC Score
6	Xgboost	0.754781	0.520267	0.087071	0.149176	0.530375
5	Gradient Boosting	0.753139	0.800000	0.000179	0.000357	0.500082
0	Logistic Regression	0.753111	1.000000	0.000022	0.000045	0.500011
4	Ada boost	0.753106	0.000000	0.000000	0.000000	0.500000
3	Random Forest Classifier	0.749101	0.460842	0.095461	0.158160	0.529424
1	KNN	0.674042	0.282646	0.208216	0.239788	0.517486
2	Decision Tree	0.630821	0.289564	0.340764	0.313084	0.533338

After Smote analysis, fine-tuned XGBoost model to improve its performance

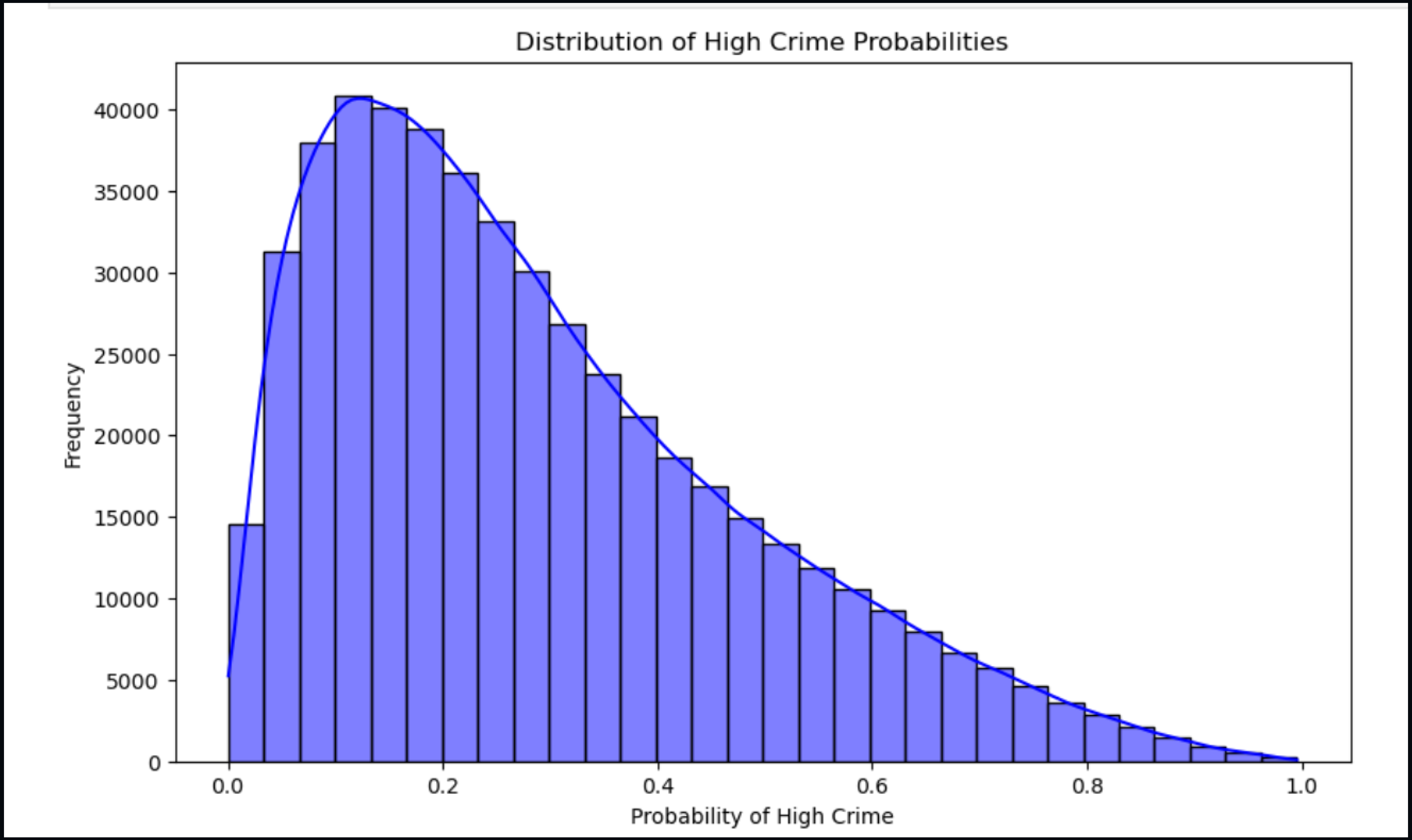
```
Tuned Accuracy: 0.8937
Tuned Precision: 0.9702
Tuned Recall: 0.8124
Tuned F1 Score: 0.8843
Tuned ROC AUC: 0.8937
Adjusted Accuracy: 0.8937
Adjusted Precision: 0.9702
Adjusted Recall: 0.8124
Adjusted F1 Score: 0.8843
Adjusted ROC AUC: 0.8937
```

Confusion Matrix





Predicted Outcomes



->Predicted Crime Counts per Neighbourhood