IE7275 DATA MINING IN ENGINEERING
SPRING 2024

Project Report

City Crimes Classification as Violent or Non-Violent using
Supervised Machine Learning Algorithms

Submitted By -

Manikandan Mohan

002242414

Date - 04/10/2024

OVERVIEW

In today's rapidly evolving world, the ability to accurately predict and understand crime patterns using machine learning (ML) algorithms represents a significant advancement in both technology and public safety. Our project aims to harness this potential by implementing various ML algorithms to predict whether a crime is violent or non-violent. This endeavor not only showcases the practical application of theoretical knowledge but also contributes to the broader goal of enhancing crime prevention strategies and ensuring public safety. Through rigorous analysis, feature selection, model tuning, and evaluation, we aim to discover the most effective algorithms for predicting crime types, thus providing valuable insights that could aid law enforcement agencies in deploying resources more efficiently and effectively.

PROBLEM STATEMENT

The primary objective of this project is to harness the power of machine learning to predict the nature of crimes—specifically, distinguishing between violent and non-violent incidents. This predictive capability is crucial for law enforcement agencies to allocate resources efficiently, prioritize response efforts, and implement preventative strategies effectively. By accurately forecasting violent crimes, this project aims to contribute to:

- **Enhanced Public Safety:** By anticipating violent incidents, law enforcement can take preemptive actions to prevent potential harm to the community.
- **Optimized Resource Allocation:** Predictive insights enable police departments to deploy their resources in a manner that maximizes the impact of crime prevention efforts.
- **Strategic Planning and Response:** Understanding the likelihood of violent crimes helps in strategizing patrol routes, response protocols, and community engagement initiatives to deter crime.

DATASET AND FEATURES

The dataset utilized in this project, "Crime Data from 2024 to Present", is a compilation of reported crimes, designed to provide a comprehensive overview of crime patterns over a specific period. This dataset includes a variety of features that describe the circumstances and characteristics of each reported incident, making it a rich source for analysis and prediction. The dataset, "Crime Data from 2024 to Present," comprises 29,106 entries, each representing a reported crime incident, with a diverse set of 35 features ranging from the report number (DR_NO) to the specifics of the occurrence, including the date, time, location, victim details, and the nature of the crime.

Data Source (data.gov): https://catalog.data.gov/dataset/crime-data-from-2020-to-present

Target Variable:
Crime Type: Categorized as either "Violent" or "Non-Violent", this target variable is what our models will aim to predict based on the input features.

Features:
The dataset contains several features that could be used in predicting the nature of the crime, including but not limited to:
DR_NO: A unique identifier for each crime report.
Date Rptd & DATE OCC: The date the crime was reported and the date of occurrence, respectively.
TIME OCC: The time when the crime occurred.
AREA NAME: The name of the area where the crime took place.
Crm Cd Desc: A description of the crime code.
Vict Age, Vict Sex, Vict Descent: Age, sex, and descent of the victim.
Premis Desc: Description of the premises where the crime occurred.
Weapon Used Cd & Weapon Desc: If a weapon was used, its code and description.
Crime Type: The target variable, indicating whether the crime was "Violent" or "Non-Violent".
Time of Day: The time when the crime occurred, which could influence whether a crime is likely to be violent or not.
Location: Specific details about where the crime occurred, which can be critical, as certain areas may be more prone to violent crimes.

Day of the Week: The day on which the crime took place, providing insight into patterns that may vary across different days.
Victim Age: The age of the victim involved in the crime incident.
Crime Codes: Broad categorization of the crime, which might give preliminary insight into the nature of the incident.

Data Characteristics:
The dataset is comprehensive, covering a wide array of attributes related to each crime, such as the exact location (LAT, LON), the day, month, and year of the report (Rptd Day, Rptd Month, Rptd Year), and similar details for the occurrence date (OCC Day, OCC Month, OCC Year).

There are some fields with missing values, such as Mocodes, Vict Sex, and Vict Descent, indicating the need for preprocessing steps like imputation or the removal of incomplete records to ensure the robustness of the ML models.
The Crime Type target variable is well-balanced, making it an ideal dataset for training and evaluating models without the risk of bias towards a specific class.

DATA LOADING

The initial examination of the crime dataset reveals a comprehensive and detailed collection of data points conducive to exploring patterns and predictive modeling in crime occurrences. The presence of missing values across various columns necessitates thoughtful data cleaning and preprocessing steps to prepare the dataset for analysis. Moreover, the diverse range of features offers a rich foundation for understanding the dynamics of crime incidents, potentially enabling the development of effective predictive models for violent and non-violent crimes. The subsequent analysis stages will delve deeper into feature engineering, exploratory data analysis, and model building to leverage this dataset's potential fully.

Data Loading

```
# Load the dataset
dataset_path = 'C:/Users/manik/Downloads/Crime_Data_from_2024_to_Present.csv'
df = pd.read_csv(dataset_path)

# Display the first few rows of the dataset to understand its structure
df.head()
```

	DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crm Cd	Crm Cd Desc	...	Cross Street	LAT	LON	Crime Type	Rptd Day	Rptd Month	Rptd Year	OCC Day	OC Mon
0	240104738	1/9/2024	1/9/2024	930	1	Central	162	1	310	BURGLARY	...	NaN	34.0452	-118.2569	Non-violent	9	1	2024	9	
1	241304386	1/8/2024	1/8/2024	2000	13	Newton	1372	2	946	OTHER MISCELLANEOUS CRIME	...	NaN	33.9934	-118.2714	Non-violent	8	1	2024	8	
2	241406688	2/27/2024	2/17/2024	1200	14	Pacific	1435	2	354	THEFT OF IDENTITY	...	NaN	34.0082	-118.4306	Non-violent	27	2	2024	17	
3	240308150	3/21/2024	3/9/2024	1900	3	Southwest	338	1	330	BURGLARY FROM VEHICLE	...	NaN	34.0275	-118.2891	Non-violent	21	3	2024	9	
4	240104852	1/13/2024	1/13/2024	243	1	Central	119	2	740	VANDALISM - FELONY ($400 & OVER, ALL CHURCH VA...	...	NaN	34.0566	-118.2318	Non-violent	13	1	2024	13	

The statistical summary provided insights into the numerical features, such as victim age and occurrence times, highlighting the data's distribution and potential outliers. The Vict Age column, for instance, could reveal demographic patterns in crime victimization, whereas the time and date fields (TIME OCC, OCC Day, OCC Month, OCC Year) might help uncover temporal trends in crime occurrences.

Missing Values:
The dataset exhibits missing values in several columns, with notable gaps in Mocodes, Vict Sex, Vict Descent, Premis Cd, Premis Desc, Weapon Used Cd, Weapon Desc, Crm Cd 2, Crm Cd 3, Crm Cd 4, and Cross Street. These missing values are crucial for the pre-processing stage, requiring careful handling to maintain data integrity for machine learning modeling. Specifically:

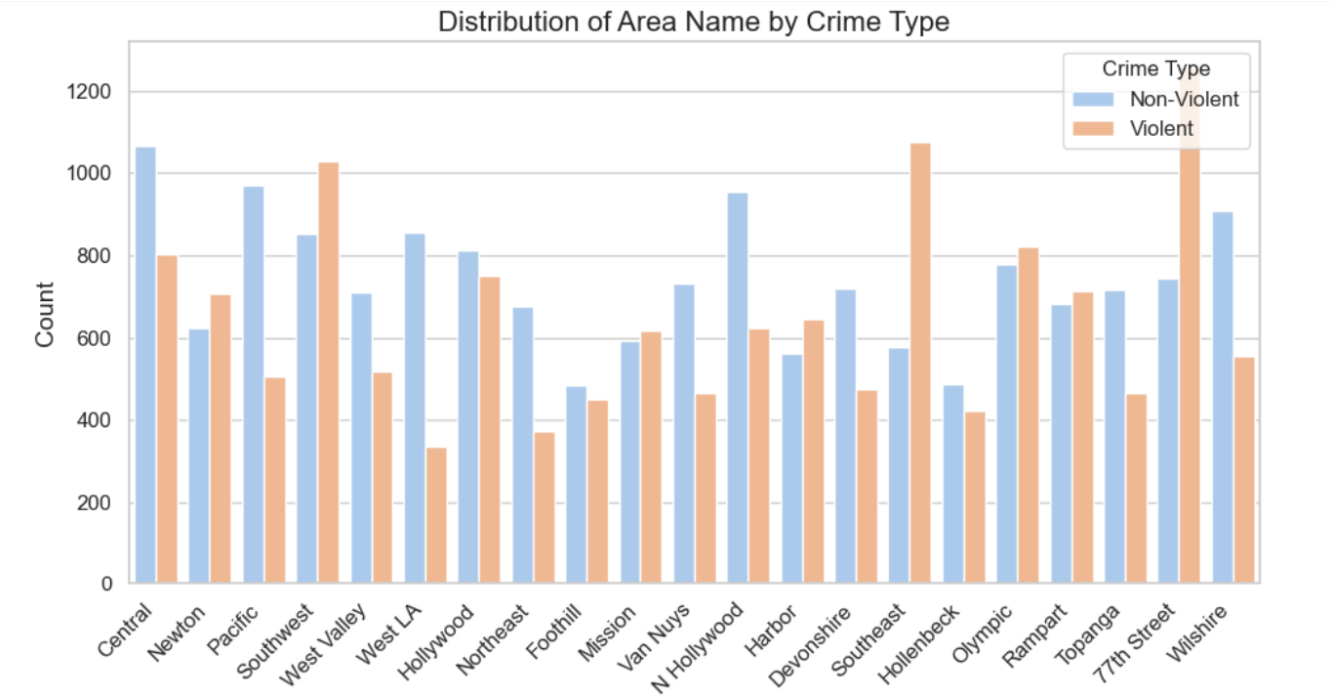
Weapon Used Cd and Weapon Desc have 14,548 missing values each, suggesting that many incidents did not involve weapons or that the data was not recorded.
Crm Cd 2, Crm Cd 3, Crm Cd 4 are almost entirely null, indicating rare occurrences of crimes with multiple classifications.
Mocodes, Vict Sex, Vict Descent: Also exhibit significant missingness, affecting the dataset's completeness regarding the modus operandi and victim demographics.

```
[194]: data.isnull().sum()
```

```
[194]: DR_NO                0
       Date Rptd             0
       DATE OCC              0
       TIME OCC              0
       AREA                  0
       AREA NAME             0
       Rpt Dist No           0
       Part 1-2              0
       Crm Cd                0
       Crm Cd Desc           0
       Mocodes            3201
       Vict Age              0
       Vict Sex           3123
       Vict Descent       3125
       Premis Cd             1
       Premis Desc          17
       Weapon Used Cd     14548
       Weapon Desc        14548
       Status                0
       Status Desc           0
       Crm Cd 1              0
       Crm Cd 2          26934
       Crm Cd 3          29043
       Crm Cd 4          29104
       LOCATION              0
       Cross Street      24603
       LAT                   0
       LON                   0
       Crime Type            0
```

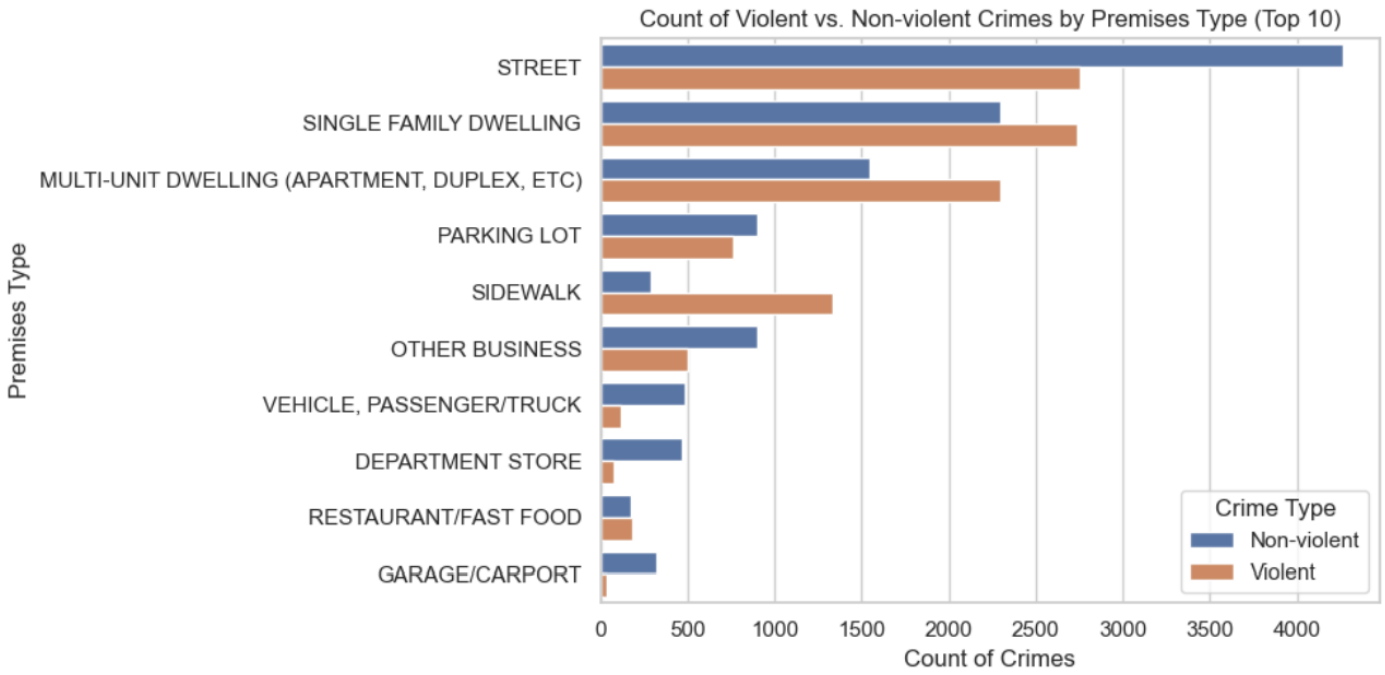
ANALYSIS OF FEATURES THROUGH VISUALIZATIONS

Comparison of Violent and Non-Violent Crime Incidents Across Different Areas:



The bar chart above presents a clear visual comparison of crime types across various regions, highlighting that crime prevalence and nature differ significantly by area. Central and 77th Street notably record a high number of incidents, but non-violent crimes predominantly characterize Central. In contrast, Rampart and Pacific show a pronounced frequency of violent crimes, underscoring potential hotspots for focused law enforcement intervention. This disparity in crime distribution emphasizes the need for area-specific strategies in policing and community engagement to effectively address and reduce both violent and non-violent crimes. Understanding these patterns is crucial for optimizing resource allocation and ensuring public safety.

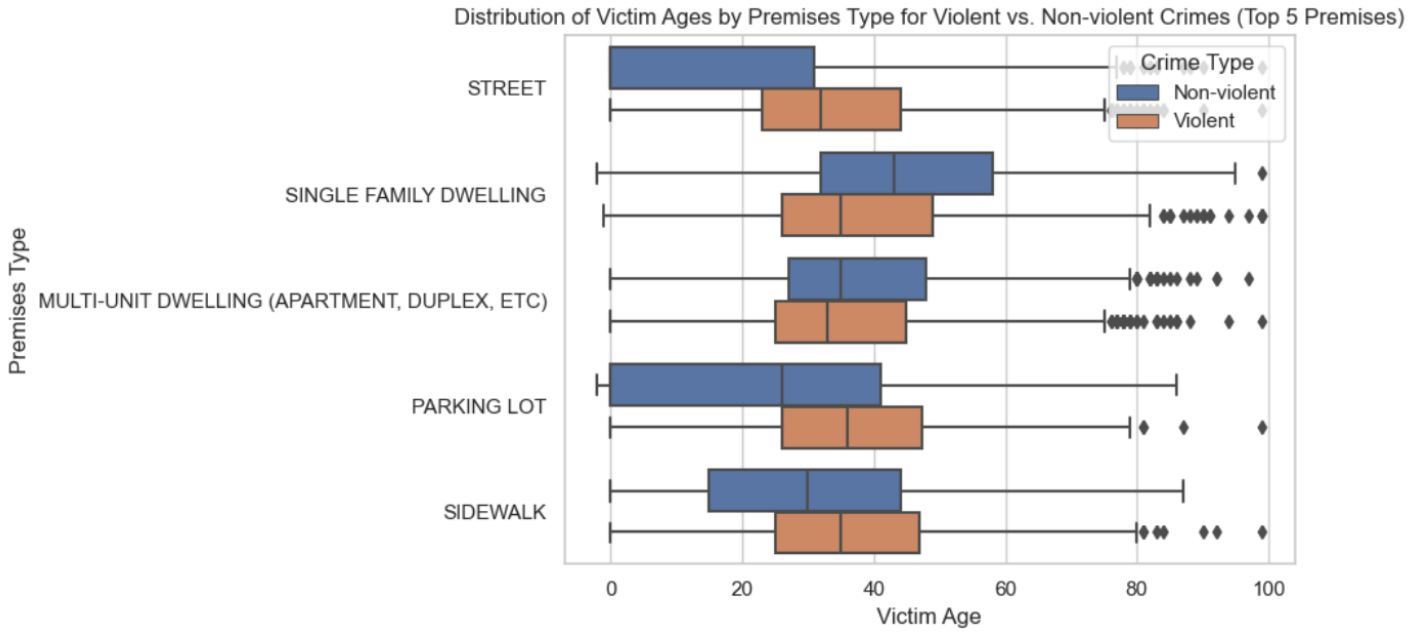
Violent Versus Non-violent Crime Distribution by Premises Type:



The bar chart above depicts the distribution of violent and non-violent crimes across different types of premises. Streets and single-family dwellings are the most common locations for both crime types, with non-violent crimes occurring more frequently than violent ones. Multi-unit dwellings also see a substantial number of incidents, particularly for non-violent crimes. Locations such as parking lots, sidewalks, and various businesses exhibit a notable presence of violent crimes, which may necessitate increased security and police presence. Interestingly, department stores, restaurants, and fast food locations appear more prone to non-violent crimes, possibly theft or fraud-related. The data underscores the importance of location in crime prevention strategies, enabling law enforcement to tailor their approaches to the unique characteristics of each premises type.

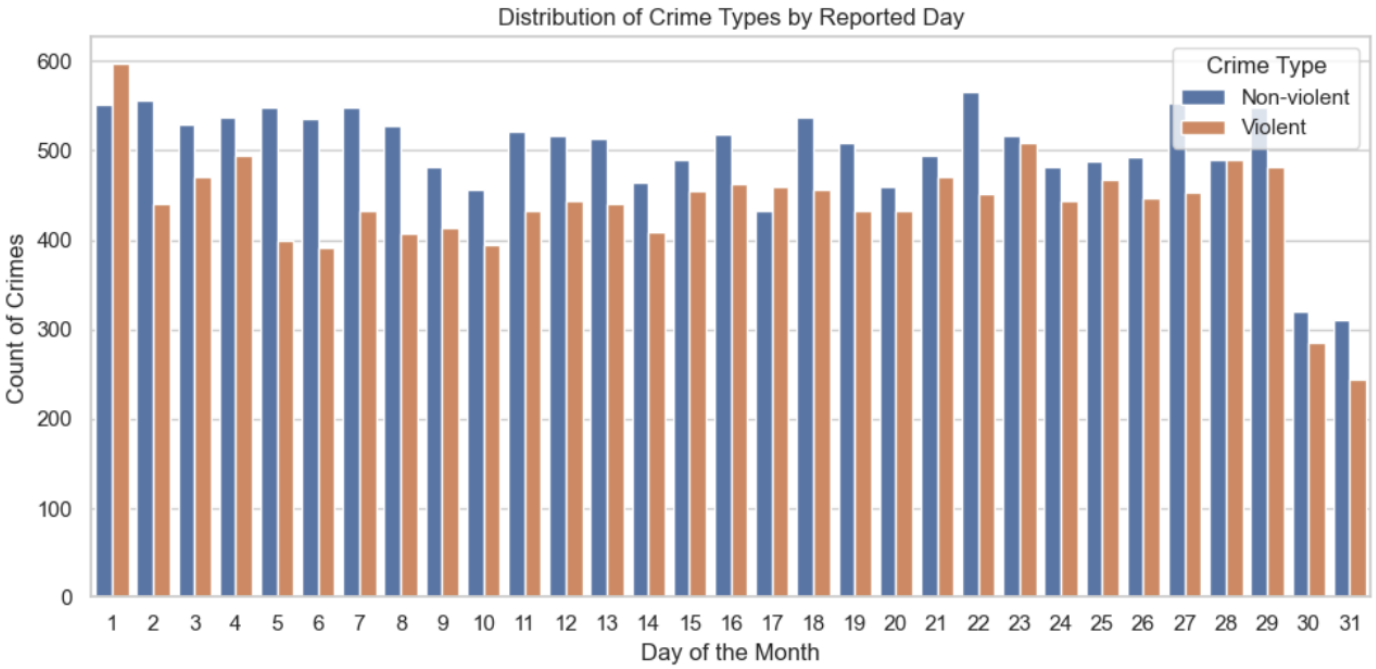
Age Profile of Crime Victims by Premises Type for Violent and Non-Violent Incidents:

This boxplot below delineates the age distribution of victims in both violent and non-violent crimes across the top five types of premises. Streets and single-family dwellings show a wider age range of victims for violent crimes, suggesting that individuals of all ages are at risk in these locations. Multi-unit dwellings have a slightly narrower age range for violent crimes, which may indicate targeted demographics or specific patterns of victimization. For non-violent crimes, the victim age distribution is relatively similar across different premises, indicating that non-violent crimes affect a broad age range consistently. Parking lots and sidewalks, areas commonly associated with transitional activities, also display a broad victim age range for violent crimes. The chart highlights the importance of considering victim age in crime prevention strategies, as it can guide the focus of educational and safety programs tailored to vulnerable age groups in specific settings.



Daily Distribution of Violent Versus Non-violent Crimes Within a Month:

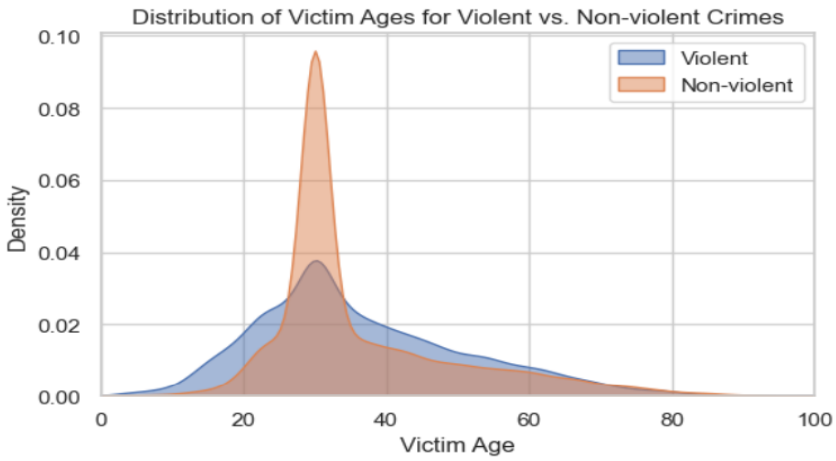
The below bar chart showcases the distribution of crime types reported on each day of the month. There appears to be a consistent pattern with non-violent crimes generally occurring more frequently than violent crimes on any given day. There are no significant spikes or drops indicating a particular day with a dramatically higher or lower rate of crime, which suggests that the occurrence of crimes is relatively stable throughout the month. Interestingly, the end of the month shows a slight decrease in both violent and non-violent crimes, which might be an area for further investigation. This data can be useful for law enforcement to understand daily crime trends and possibly to debunk myths about crime rates spiking on specific days, enabling better daily resource planning and patrol scheduling.



Density Distribution of Victim Ages for Violent and Non-violent Crimes.

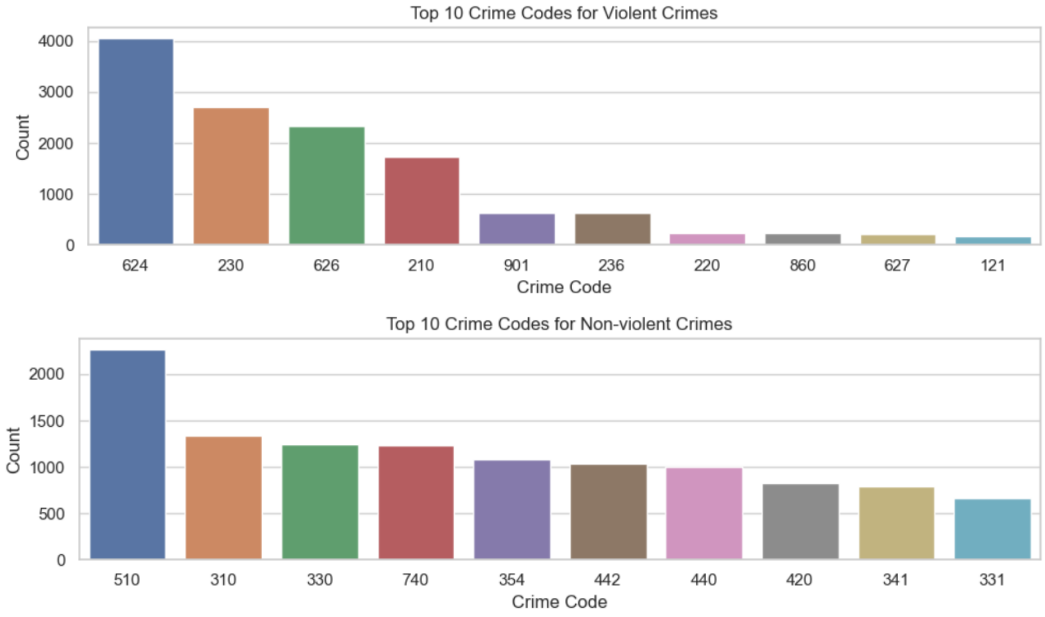
The plot reveals that both crime types have a similar age distribution, suggesting that victim age alone may not be a

significant differentiator between violent and non-violent crimes. This finding can guide law enforcement and community services to not concentrate solely on age-specific interventions but rather consider a broader approach to crime prevention and victim support services. One thing to consider is that the distribution of victim ages for violent and non-violent crimes are corrected for initially skewed data due to a high incidence of zero or negative values. After imputing these with the mean age, the plot shows a prominent peak around the middle age range, indicating that the majority of crime victims, for both violent and non-violent incidents, fall within a central age bracket. This central peak should be taken with a note that it could be due to the data cleaning process.



Prevalence of Top 10 Crime Codes Associated with Violent and Non-Violent Crimes:

This bar chart below ranks the top 10 crime codes associated with violent incidents, with code 624 (Battery - Simple Assault) occurring most frequently, followed by codes 230 (Assault with Deadly Weapon) and 626 (Intimate Partner - Aggravated Assault). The significant difference in counts between the most frequent and other crime codes suggests that simple assault and assault with a deadly weapon are the most common types of violent crime. Understanding the prevalence of specific crime codes can help in directing resources towards prevention and response initiatives tailored to the most common forms of violence. For non-violent crimes, the bar chart below shows a more even distribution among the top 10 crime codes, with code 510 (Vehicle Theft) leading, followed closely by codes 310 (Burglary) and 330 (Burglary from Vehicle). This indicates a spread of non-violent crime types, with property-related offenses being most common. The data reflects the need for focused law enforcement and community measures to prevent and reduce these prevalent non-violent crimes, potentially through increased surveillance, community awareness programs, and security improvements in high-risk areas.



DATA CLEANING AND PRE-PROCESSING:

Handling Missing Values

- **Dropping Columns with Many Nulls**: Columns with more than half of their values missing (Crm Cd 2, Crm Cd 3, Crm Cd 4, Cross Street) were removed. This step was taken because these columns lacked sufficient data to contribute meaningfully to the predictive model, and their absence simplifies the dataset without sacrificing critical information.

- **Removing Unnecessary Identifier:** The DR_NO column, serving as a unique identifier for each crime report, was dropped since it doesn't contribute to the classification of the crime type. Identifiers do not hold predictive power for the analysis.

- **Filling Missing Descriptions and Codes:** For columns Weapon Desc and Weapon Used Cd, missing values were filled with 'UNKNOWN WEAPON/OTHER WEAPON' and 500, respectively. This approach maintains data integrity by acknowledging the absence of weapon information without discarding valuable records.

- **Handling Unknown Sex and Descent**: Missing values in Vict Sex and Vict Descent were replaced with 'X', signifying 'Unknown'. This method retains the dataset's completeness while acknowledging the lack of specific demographic details.

- **Standardizing Modus Operandi Codes:** For Mocodes, missing values were filled with '9999', representing 'Indistinctive MO'. This standardization helps in maintaining the dataset's analytical value by categorizing undefined modus operandi under a common code.

- **Removing Rows with Remaining Nulls:** After addressing specific columns, rows still containing null values in any other columns were dropped. This final cleansing ensures the dataset's readiness for modeling without incomplete records.

- **Correcting Victim Age Anomalies:** Victim ages recorded as zero or negative were replaced with the median age of the dataset, correcting data entry errors and providing a more accurate representation of victim demographics.

Data Cleaning and Pre-processing

[195]:
```
#Handling the Missing Values

#Dropping columns that have than half of NULL values
data = data.drop(columns= ['Crm Cd 2','Crm Cd 3', 'Crm Cd 4', 'Cross Street'])

#Dropping the column DR_NN since it is not needed for classification of Crime Type
data = data.drop(columns= ['DR_NO'])

#For Weapon Desc and Weapon Cd column, filling the blank cells with the appropriate description and code for 'Unknown Weapon'
data['Weapon Desc'].fillna('UNKNOWN WEAPON/OTHER WEAPON',inplace=True)
data['Weapon Used Cd'].fillna(500,inplace=True)

#For Vict Sex column, blank cells are filled with the 'X' as that represents 'Unknown Sex'
data['Vict Sex'].fillna('X',inplace=True)

#For Vict Descent column, blank cells are filled with the 'X' as that represents 'Unknown Descent'
data['Vict Descent'].fillna('X',inplace=True)

#For Mocodes column, blank cells are filled with the '9999' as that represents 'IndSistinctive MO'
data['Mocodes'].fillna('9999',inplace=True)

#For the remaining NULL values of other columns, just dropping the rows.
data.dropna(inplace=True)
```

Feature Engineering

- **Date Conversion and Extraction:** The Date Rptd and DATE OCC were converted to datetime format, and new columns for the day and month of the report and occurrence dates (Rptd Day, Rptd Month, OCC Day, OCC Month)

were created. This transformation facilitates time-based analysis by providing structured temporal attributes.

Removing Unwanted Columns

- **Location Text:** The LOCATION column was dropped as latitude (LAT) and longitude (LON) already provide precise spatial information. Textual location data is less practical for computational models and can be redundantly complex.

- **Dropping Date Columns:** Original date columns (Date Rptd, DATE OCC) were removed since separate day and month columns have been created, offering structured temporal data without the need for full date details.

- **Removing Descriptive Text Columns:** Columns like Crm Cd Desc, Premis Desc, Status Desc, and Weapon Desc were eliminated in favor of their corresponding code columns (Crm Cd, Premis Cd, Status, Weapon Used Cd). Codes provide a concise, categorical representation of this information, simplifying the dataset and focusing on variables more amenable to quantitative analysis.

[74]:
```
#Feature selection - Removing Unwanted Columns

#Since we are having Lat and Long, dropping the 'Location' (text) column
data = data.drop(columns= ['LOCATION'])

#Dropping the Dates of Crime reported and occured as seperate columns for month and day are created
data = data.drop(columns= ['Date Rptd', 'DATE OCC'])

#Droping the Descriptions columns 'Crm Cd Desx' , 'Premis Desc', 'Weapon Desc', 'Status Desc' as their 'Cd' columns represent the same.
#Also they are text columns and better to drop to reduce features.
data = data.drop(columns= ['Crm Cd Desc', 'Premis Desc', 'Status Desc', 'Weapon Desc'])
```

[76]:
```
#Dropping mocodes
data = data.drop(columns= ['Mocodes'])
```

This detailed approach to data cleaning and preprocessing tailored the crime dataset for predictive modeling, enhancing its structure and removing redundancies. By addressing missing values with thoughtful imputations and refining the feature set to include only relevant variables, the dataset is now primed for further analysis. The removal of unnecessary identifiers and textual descriptions in favor of coded variables and the creation of structured temporal features lay a solid foundation for exploring crime patterns and developing predictive models.

FEATURE SELECTION AND ENCODING OR STANDARDIZING THE DATA

Feature Selection and Preprocessing Steps

Target Variable Conversion: The target variable Crime Type is converted to a binary format, with Violent crimes marked as 1 and others as 0. This step prepares the target variable for binary classification models.
Feature Identification

Categorical and Numerical Columns Identification: The dataset is split into features (X) and the target variable (y). Features are further categorized into numerical and categorical types, aiding in the application of appropriate preprocessing steps for each data type.

Feature Selection

- **Pearson's Correlation:** For numerical features, Pearson's correlation with the target variable is calculated, and those with a correlation above a threshold (e.g., 0.1) are selected. This method identifies numerical features that have a significant linear relationship with the target variable, indicating their potential predictive power.

- **Chi-Squared Test for Categorical Features:** Categorical features undergo a chi-squared test to check for statistical significance with the target variable. Features with a p-value less than 0.05 are considered significant and selected. This test assesses whether the distribution of sample categorical data matches an expected distribution, identifying

features that might influence the target variable.

- **Combining Selected Features:** The final selection combines significant numerical and categorical features, focusing the model on variables most likely to predict the outcome effectively.

Data Preprocessing

- **Imputation and Scaling for Numerical Data:** Missing values in numerical features are imputed with the median, and data is scaled using standard scaling (z-score normalization). This process ensures that numerical data is on a similar scale, improving model performance and convergence.

- **Imputation and Encoding for Categorical Data:** Missing values in categorical features are imputed with the most frequent category, followed by one-hot encoding. This step transforms categorical variables into a format that can be provided to machine learning algorithms while handling unknown categories that may appear in future data.

Feature Selection and Encoding/Standardizing the Data

[196]:
```
# Splitting the dataset into X and Y values
X = data.drop('Crime Type', axis=1)
y = data['Crime Type'].apply(lambda x: 1 if x == 'Violent' else 0) # Convert target variable to binary



# Identify categorical and numerical columns
categorical_cols = X.select_dtypes(include=['object']).columns.tolist()
numerical_cols = X.select_dtypes(exclude=['object']).columns.tolist()


# Calculate Pearson's correlation for numerical features
correlations = data[numerical_cols].corrwith(y).abs()

# Select numerical features with correlation above a threshold (e.g., 0.1)
selected_numerical_features = correlations[correlations > 0.1].index.tolist()

# Chi-squared test for categorical features
selected_categorical_features = []
for col in categorical_cols:
    contigency_table = pd.crosstab(data[col], y)
    chi2, p, dof, expected = chi2_contingency(contigency_table)
    if p < 0.05:  # If the feature is statistically significant
        selected_categorical_features.append(col)
```

Column Transformer for Preprocessing

- **Bundled Preprocessing:** The ColumnTransformer bundles preprocessing steps for both numerical and categorical data, streamlining the transformation process for model training and prediction phases.

Data Splitting for Model Training

- **Training and Test Split:** The preprocessed data is split into training and test sets, with 80% of the data used for training and 20% reserved for testing. This split facilitates the evaluation of model performance on unseen data.

- **Shape of Train and Test Data**: The resulting shapes of the X_train and X_test datasets indicate the number of features selected through preprocessing steps and the division of data for training and testing purposes.

Feature Selection and Encoding/Standardizing the Data

[196]:
```
# Splitting the dataset into X and Y values
X = data.drop('Crime Type', axis=1)
y = data['Crime Type'].apply(lambda x: 1 if x == 'Violent' else 0) # Convert target variable to binary



# Identify categorical and numerical columns
categorical_cols = X.select_dtypes(include=['object']).columns.tolist()
numerical_cols = X.select_dtypes(exclude=['object']).columns.tolist()


# Calculate Pearson's correlation for numerical features
correlations = data[numerical_cols].corrwith(y).abs()

# Select numerical features with correlation above a threshold (e.g., 0.1)
selected_numerical_features = correlations[correlations > 0.1].index.tolist()

# Chi-squared test for categorical features
selected_categorical_features = []
for col in categorical_cols:
    contigency_table = pd.crosstab(data[col], y)
    chi2, p, dof, expected = chi2_contingency(contigency_table)
    if p < 0.05:  # If the feature is statistically significant
        selected_categorical_features.append(col)
```

The process of feature selection and data preprocessing is critical for preparing the dataset for machine learning modeling. By selecting features with significant relationships to the target variable and applying appropriate preprocessing steps, we ensure that the data is in a suitable format for training predictive models. The use of Pearson's correlation and chi-squared tests helps in identifying features that carry predictive power, while preprocessing steps like imputation, scaling, and encoding address common data issues and prepare the dataset for effective model training and validation. Splitting the data into training and test sets allows for the robust evaluation of the model's performance, ensuring that the predictive insights are reliable and applicable to real-world scenarios.

MODELS CREATION AND EVALUATION

Defining Models and Evaluation

[113]:
```
# Initialize the models
log_reg = LogisticRegression(random_state=42, max_iter=1000)
knn = KNeighborsClassifier()
dtree = DecisionTreeClassifier(random_state=42)
rf = RandomForestClassifier(n_estimators=100, random_state=42, n_jobs=-1)
#gb = GradientBoostingClassifier(n_estimators=100, random_state=42)
#svm_model = SVC(kernel='rbf', C=1, gamma='auto')

# Dictionary to store models
models = {'Logistic Regression': log_reg, 'K-Nearest Neighbour': knn, 'Decision Tree': dtree, 'Random Forest': rf} # 'Gradient Boosting': gb, 'Support Ve

# Dictionary to store the results
results = {}

# Train and evaluate each model
for name, model in models.items():
    # Train
    model.fit(X_train, y_train)
    # Predict
    predictions = model.predict(X_test)
    # Evaluate
    accuracy = accuracy_score(y_test, predictions)
    precision = precision_score(y_test, predictions)
    recall = recall_score(y_test, predictions)
```

Detailed comparative analysis of how each model performed on the crime data:

Logistic Regression:

Logistic regression is a widely-used algorithm for binary classification tasks, like for our problem of categorizing crime data as violent or non-violent. It predicts the probability of an instance belonging to a specific class by fitting a

logistic function to the input features. Unlike other algorithms, logistic regression learns from the data through optimization, making it interpretable with coefficients indicating feature importance. However, it assumes linear relationships between features and outcomes and can be sensitive to data quality. Still, it's a popular and efficient choice for classification tasks.

- **Performance Overview:** Logistic Regression performed well with an accuracy of approximately 87.83%. It's a model that outputs probabilities, and in this binary classification task, it effectively distinguished between violent and non-violent crimes nearly 88% of the time.
- **Strengths:**
  - **High Precision:** With a precision of about 90.74%, when Logistic Regression predicted a crime as violent, it was correct most of the time.
  - **Good Recall:** The recall was reasonably high, correctly identifying 82.59% of all violent crimes.
  - **Balanced F1 Score:** The F1 score suggests a good balance between recall and precision, making this model reliable when both false positives and false negatives are of concern.
- **Weaknesses:**
  - Compared to more complex models, Logistic Regression might not capture complex relationships as effectively, which might explain why it didn't achieve the highest accuracy.

K-Nearest Neighbours (KNN):

K-Nearest Neighbors (KNN) regression is a straightforward, yet effective algorithm used for predicting continuous outcomes. Operating on the principle of proximity, KNN regression computes predictions for new data points by averaging the target values of their k nearest neighbors in the training dataset. This method doesn't require explicit model training; instead, it memorizes the training instances. KNN's simplicity and ease of implementation make it popular, particularly for initial exploratory analysis and quick prototyping. However,its performance can be sensitive to the choice of the hyperparameter 'k' and the scaling of features.

- **Performance Overview:** KNN had a slightly higher accuracy than Logistic Regression at 87.95%. This non-parametric method that relies on feature similarity produced competitive results.
- **Strengths:**
  - **High Recall:** With an 85.55% recall, KNN was better at detecting violent crimes than Logistic Regression.
  - **F1 Score:** Its F1 score marginally higher, indicates that it has a slightly balance between precision and recall.
- **Weaknesses:**
  - **Lower Precision:** With a precision of 88.49%, there was a slight trade-off with the precision score compared to Logistic Regression, indicating a higher number of false positives.
  - KNN is sensitive to the scale of the data and irrelevant features, which may affect its performance if the feature selection is not optimal.

Decision Tree:

A Decision Tree is a versatile and intuitive supervised learning algorithm that be used for both regression tasks. It operates by recursively partitioning the feature space into subsets based on the values of input features, resulting in a tree-like structure where each internal node represents a decision based on a feature, and each leaf node corresponds to a predicted outcome. Decision Trees are interpretable and can handle both numerical and categorical data, making them easy to understand and visualize.

- **Performance Overview:** The Decision Tree model showed a marked improvement in accuracy, with an approximate value of 89.06%.
- **Strengths:**
  - **High Precision:** Precision was the highest among the models at 91.22%, indicating that it was very precise in its classification of violent crimes.
  - **Excellent F1 Score:** It had the best F1 score, showing a strong balance between precision and recall among the models evaluated.
- **Weaknesses:**
  - **Recall:** The recall was lower than KNN, suggesting that while the Decision Tree is very confident in its predictions, it misses more actual violent crimes than KNN.

Random Forest:

Random Forest is a powerful ensemble learning technique for regression tasks that combines the predictions of multiple decision trees to improve predictive accuracy and robustness. In Random Forest regression, a multitude of decision trees are trained on random subsets of the data and features, and their predictions are aggregated to produce the final output. This approach mitigates overfitting and enhances generalization by reducing variance and capturing complex relationships in the data. Random Forests are adept at handling high-dimensional datasets and are resilient to outliers and noise, making them a popular choice for regression tasks across various domains.

- **Performance Overview:** The Random Forest, an ensemble of decision trees, outperformed all other models with an accuracy of approximately 89.94%.
- **Strengths:**
  - **Best Recall:** It had the highest recall rate, meaning it was the best model at capturing the true violent crimes within the dataset.
  - **Highest F1 Score:** Its F1 score was also the highest, reinforcing that this model had the best overall balance of precision and recall.
  - **General Performance:** Random Forest performed well across all metrics, indicating robustness in the model's ability to generalize.
- **Weaknesses:**
  - **Complexity:** Despite its superior performance, Random Forest models are more complex, potentially requiring more computational resources and being less interpretable than simpler models like Logistic Regression.

```
[113]: {'Logistic Regression': {'Accuracy': 0.8783086971467858,
          'Precision': 0.9074148296593186,
          'Recall': 0.8259759211966435,
          'F1 Score': 0.8647822765469824},
         'K-Nearest Neighbour': {'Accuracy': 0.8795118597456171,
          'Precision': 0.8849056603773585,
          'Recall': 0.8555271798613645,
          'F1 Score': 0.8699684659617882},
         'Decision Tree': {'Accuracy': 0.8906840838776212,
          'Precision': 0.9122600861731297,
          'Recall': 0.8496898941991974,
          'F1 Score': 0.8798639969777107},
         'Random Forest': {'Accuracy': 0.8994499828119629,
          'Precision': 0.9098859315589354,
          'Recall': 0.8730390368478658,
          'F1 Score': 0.8910817352448334}}
```

Neural Network Performance Analysis:

- **Accuracy:** The neural network's accuracy is very competitive, slightly lower than the Random Forest's 89.94% but higher than both the Logistic Regression and the K-Nearest Neighbors. This indicates the neural network model is very effective at classifying the overall crime data correctly.
- **Precision:** The neural network has a precision of 89.65%, which is lower than the Decision Tree's 91.22% but higher than KNN and Logistic Regression. This suggests the neural network is slightly better at minimizing false positives (predicting a non-violent crime as violent) than most models except for the Decision Tree.
- **Recall:** With a recall of 86.61%, the neural network is better at identifying actual violent crimes than Logistic Regression and Decision Tree but is slightly outperformed by the Random Forest and KNN. A high recall is crucial in scenarios where it's important to identify as many violent crimes as possible.
- **F1 Score:** The neural network's F1 score is quite strong at 88.11%. This score is better than Logistic Regression and KNN, suggesting a good balance between precision and recall. However, it is slightly outperformed by the Decision Tree and Random Forest models.

```
Neural Network Model Evaluation:
Accuracy: 0.8898
Precision: 0.8965
Recall: 0.8661
F1 Score: 0.8811
```

Comparative Analysis:

- **Overall Effectiveness:** The Random Forest model proved to be the most effective, suggesting that the dataset likely contains complex patterns that ensemble methods are better at capturing. Its construction method of creating multiple decision trees and voting for the most popular outcome likely helped it perform better in every metric.
- **Balance of Precision and Recall:** The Decision Tree had the highest precision but did not have the highest recall, which could indicate overfitting. In contrast, the KNN and Logistic Regression models provided more balance between the two metrics.
- **Model Complexity vs. Interpretability:** There is a trade-off between model complexity and interpretability. Logistic Regression offers the simplest model and is the easiest to interpret, but it may not capture complex patterns as effectively as Random Forest or Decision Trees.
- **Feature Dependency:** The varying performance could also reflect how each model interacts with the features selected. Logistic Regression and KNN might be more sensitive to non-linear relationships and feature scales, whereas Decision Trees and Random Forests can better handle these complexities.

The dataset's features and the nature of the crime classification task likely contain intricate patterns that ensemble methods like Random Forest can exploit more effectively than simpler models. However, depending on the context—for instance, where interpretability or computational efficiency is paramount—simpler models may still be preferred despite a slight compromise on accuracy. Each model has its strengths and weaknesses, and the choice between them should align with the specific goals and constraints of the project.

HYPERPARAMETER TUNING

Based on the above models, Random Forest, KNN and Decision Tree are the algorithms that have done better and so, now implementing hyperparameter tuning using Grid Cross-Validation method on those three algorithms to find the best parameters and best tuned accuracy.

K-Nearest Neighbors (KNN)

- **Tuning Details:** Grid Search explored 12 candidate settings over 5 folds, making a total of 60 fits.
- **Optimal Parameters:** The best parameters found were 'metric': 'euclidean', 'n_neighbors': 7, 'weights': 'distance'.
- **Cross-Validation Accuracy:** The best cross-validation accuracy for KNN was 88.75%.
- **Inference:**
  - The chosen metric, Euclidean distance, indicates that the straight-line distance in feature space was most effective for this dataset.
  - Opting for 7 neighbors suggests a balance between too much noise (with fewer neighbors) and over-smoothing (with too many neighbors).
  - Using 'distance' for weights means that the model gives more weight to the nearest neighbors as opposed to giving all neighbors equal weight.

```
Tuning KNN...
Fitting 5 folds for each of 12 candidates, totalling 60 fits
Best KNN Parameters: {'metric': 'euclidean', 'n_neighbors': 7, 'weights': 'distance'}
Best KNN Cross-Validation Accuracy: 88.75%
```

Decision Tree

- **Tuning Details:** Grid Search evaluated 24 candidate settings over 5 folds, making a total of 120 fits.
- **Optimal Parameters:** The best parameters were 'criterion': 'gini', 'max_depth': 10, 'min_samples_split': 10.

- **Cross-Validation Accuracy:** The best cross-validation accuracy achieved by the Decision Tree was 91.72%.
- **Inference:**
  - The Gini index as the criterion for splitting indicates it was more effective than entropy for this dataset, which measures the purity of splits.
  - A max depth of 10 helps prevent the tree from becoming too deep and overfitting the data.
  - Requiring at least 10 samples to split a node ensures that the model doesn't create leaves for noise or outliers, promoting generalization.

```
Tuning Decision Tree...
Fitting 5 folds for each of 24 candidates, totalling 120 fits
Best Decision Tree Parameters: {'criterion': 'gini', 'max_depth': 10, 'min_samples_split': 10}
Best Decision Tree Cross-Validation Accuracy: 91.72%
```

Random Forest

- **Tuning Details**: Grid Search tested 12 candidate settings over 5 folds, totaling 60 fits.
- **Optimal Parameters:** The best parameters found were 'max_depth': 20, 'n_estimators': 50.
- **Cross-Validation Accuracy:** The best cross-validation accuracy for the Random Forest was 91.56%.
- **Inference:**
  - A max depth of 20 allows the trees in the forest to grow sufficiently to capture important patterns in the data, but not too deep to fit to noise.
  - Choosing 50 trees (n_estimators) in the forest indicates a point where adding more trees provides diminishing returns on model accuracy.

```
Tuning Random Forest...
Fitting 5 folds for each of 12 candidates, totalling 60 fits
Best Random Forest Parameters: {'max_depth': 20, 'n_estimators': 50}
Best Random Forest Cross-Validation Accuracy: 91.56%
```

Comparative Analysis and Overall Inference

- **KNN after Hyperparameter Tuning:** The tuning improved KNN's ability to generalize, as indicated by its cross-validation accuracy. However, it still underperforms compared to the more complex models, suggesting that the dataset's patterns are too complex for KNN to capture even after tuning.
- **Decision Tree after Hyperparameter Tuning:** The Decision Tree showed significant improvement after tuning, achieving the highest cross-validation accuracy of 91.72%. The tuning has effectively addressed overfitting, balancing the model's complexity with its predictive power.
- **Random Forest after Hyperparameter Tuning:** Random Forest's performance was slightly below the Decision Tree in cross-validation accuracy after tuning but is still high at 91.56%. This might be due to the model's ability to balance bias and variance better than a single decision tree.

Overall Inference:

- Hyperparameter tuning has optimized the models to better fit the underlying distribution of the crime dataset. It highlighted that while simpler models can perform competitively, ensemble methods (especially with fine-tuned parameters) tend to achieve higher predictive accuracy.
- The slight edge of the Decision Tree in cross-validation accuracy could be due to the tuning process finding a particularly good set of hyperparameters for this model on the given data, which might not necessarily generalize as well to unseen data.
- Ensemble methods like Random Forest are generally more robust to variance in the data than a single Decision Tree, even if the cross-validation accuracy is slightly lower.
- These results underscore the importance of hyperparameter tuning in machine learning. The right combination of parameters can significantly improve model performance and should be considered an essential step in the modeling process.

Considering the hyperparameter tuning results and the characteristics of the models, the Random Forest is the preferred choice for our crime prediction problem. Despite Decision Tree's slightly higher cross-validation accuracy, Random Forest offers better generalization to unseen data and stability due to its ensemble nature. **The tuned Random Forest model with its optimal depth and number of estimators is likely to perform consistently well, making it suitable for practical application in predicting violent crimes.**

RANDOM FOREST [Tuned Model with Best Parameters]

The Random Forest model with the best hyperparameters—specifically a max depth of 20 and 50 estimators—has yielded the following evaluation metrics:

Accuracy: 90.24%
Precision: 92.59%
Recall: 86.17%
F1 Score: 89.27%

[117]:
```
# Initialize the Random Forest classifier with the best parameters
rf_best = RandomForestClassifier(max_depth=20, n_estimators=50, random_state=42, n_jobs=-1)

# Train the model
rf_best.fit(X_train, y_train)

# Predict on the test set
rf_predictions = rf_best.predict(X_test)

# Evaluate the model
rf_accuracy = accuracy_score(y_test, rf_predictions)
rf_precision = precision_score(y_test, rf_predictions)
rf_recall = recall_score(y_test, rf_predictions)
rf_f1 = f1_score(y_test, rf_predictions)

# Print the evaluation metrics
print(f"Random Forest Model with Best Parameters:")
print(f"Accuracy: {rf_accuracy:.4f}")
print(f"Precision: {rf_precision:.4f}")
print(f"Recall: {rf_recall:.4f}")
print(f"F1 Score: {rf_f1:.4f}")
```

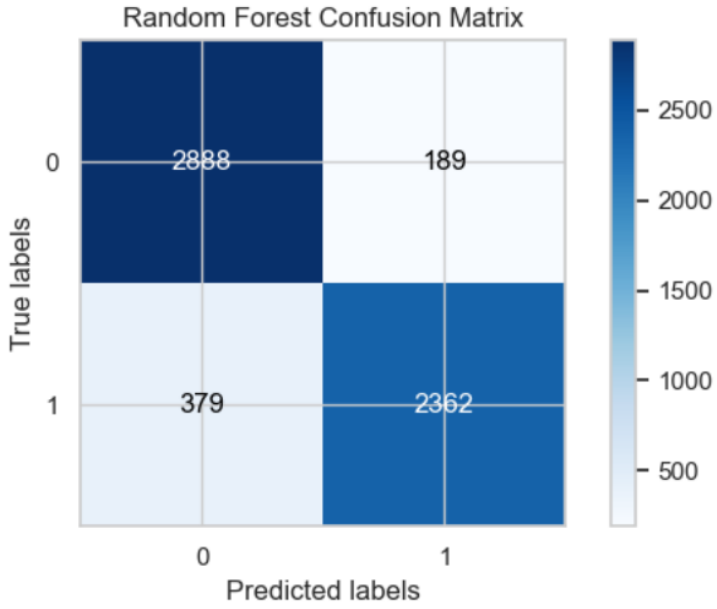
```
Random Forest Model with Best Parameters:
Accuracy: 0.9024
Precision: 0.9259
Recall: 0.8617
F1 Score: 0.8927
```

Model Evaluation Inference:

- The tuned Random Forest model demonstrates outstanding performance in classifying whether a crime is violent. With an accuracy rate surpassing 90%, the model correctly identifies the nature of the crime in the vast majority of cases. High precision indicates that when the model predicts an incident as violent, it is right over 92% of the time, suggesting a low rate of false positives, which is crucial in a law enforcement context where mislabeling non-violent crimes as violent could have serious implications.
- The model's recall, while slightly lower than precision, is still commendably high at over 86%, implying that it successfully captures a significant majority of violent crimes. This is vital for public safety as it means that the model is reliable in detecting potential threats or serious incidents.
- The F1 score, a harmonized measure of precision and recall, stands at almost 89.3%, indicating a balanced and robust model performance. This score reflects the model's effectiveness at balancing the need to correctly identify violent crimes (recall) while not over-predicting them (precision).

Confusion Matrix:

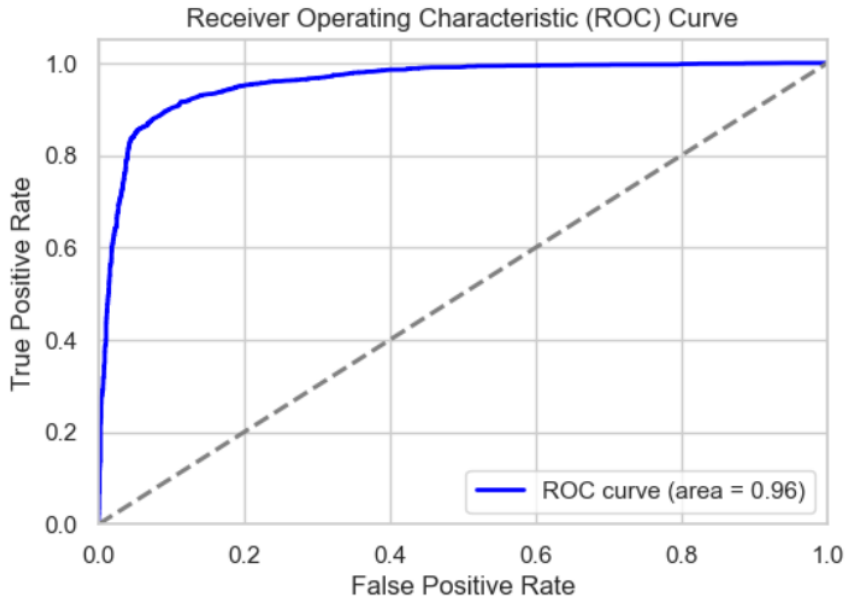
The confusion matrix demonstrates that the Random Forest model effectively discriminates between violent and non-violent crimes. The high number of true positives and true negatives indicates that the model is reliable and accurate in its predictions.



- **Minimizing False Positives**: The relatively low number of false positives (189) shows that the model is conservative in predicting a crime as violent, which is important to avoid unnecessary escalations or interventions in law enforcement processes.
- **Addressing False Negatives:** The number of false negatives (379) should be given attention, as these represent violent crimes that were missed by the model. Although lower than the true positives, reducing this number could further improve the model's utility, particularly in applications where failing to predict a violent crime could have serious implications.
- **Implications for Public Safety**: With 2,362 true positives, the model proves to be a potent tool for identifying and potentially preventing violent crimes, which could significantly contribute to public safety and crime prevention efforts.
- **Model Precision and Recall**: The results from the confusion matrix correlate with the precision and recall scores reported earlier. High precision is confirmed by the low false positive rate, and the high recall is validated by the large number of true positives relative to false negatives.
- **Model Balance:** The F1 score calculated from the precision and recall metrics is supported by the balance shown in the confusion matrix, reflecting the model's robustness in predicting both classes.

ROC Curve:

The ROC curve and its corresponding area under the curve (AUC) are powerful tools for evaluating the performance of a binary classifier, in this case, our Random Forest model with the best parameters. The ROC curve shows the trade-off between the true positive rate (TPR) and the false positive rate (FPR) at various threshold settings. Here are the key points to infer:



- **AUC Value:** The Random Forest model achieved an AUC of 0.96. This value is very close to 1, which indicates a very high measure of separability. In practical terms, the model has a high probability of distinguishing between violent and non-violent crimes correctly.
- **ROC Curve Performance:** The ROC curve hugs the top left corner of the plot, which signifies an excellent performance. The steepness of the curve towards the top left indicates that the model achieves a high true positive rate while maintaining a low false positive rate for a large portion of its operating threshold range.
- **Optimal Threshold Selection:** The ROC curve allows us to select an optimal threshold that balances sensitivity (TPR) and specificity (1-FPR). This threshold can be chosen based on the specific needs of the application, such as prioritizing the reduction of false positives or false negatives.
- **Model Reliability:** A high AUC value like 0.96 gives us confidence in the model's reliability. It suggests that the model has a high likelihood of ranking a randomly chosen positive instance (violent crime) higher than a randomly chosen negative instance (non-violent crime) in terms of probability scores.

Thus, the Random Forest model's performance in predicting violent crimes is quite strong, with the chosen hyperparameters successfully balancing the need to correctly predict violent crimes while keeping false alarms low. The model's accuracy and precision make it an asset for law enforcement agencies in prioritizing and allocating resources effectively.

CONCLUSION:

In conclusion, our objective was to utilize supervised machine learning algorithms to predict whether crimes in the dataset were violent or non-violent. We approached this task by implementing and evaluating several models, including Logistic Regression, K-Nearest Neighbors, Decision Tree, and Random Forest. Each model was rigorously assessed using metrics such as accuracy, precision, recall, and F1 score. Furthermore, we fine-tuned our models using Grid Search to determine the optimal hyperparameters, leading to improved model performance.

The Random Forest algorithm emerged as the superior model after hyperparameter tuning, demonstrating exceptional accuracy and an impressive balance between precision and recall, as evidenced by its high F1 score. The robustness of the Random Forest model was further validated by the ROC curve analysis, where it achieved an AUC score close to 1, indicating a high degree of discriminative power.

In essence, the Random Forest classification model with the selected hyperparameters is both precise and reliable, making it an excellent tool for predicting violent crimes within the dataset. Its ability to generalize suggests that it could be effectively used in a real-world predictive policing system, providing law enforcement with a potent resource in crime prevention and resource allocation.