

***SanFrancisco Crime Prediction and Analysis***

BINDIA BINDIA

MOHAMMED ADHIL GHOUSE MOHIUDDIN TAHIR

PIYUSH PIYUSH



**Abstract:**

Crime prediction and analysis of crime patterns is an important area of research as it can help law enforcement agencies to allocate resources more effectively and reduce crime rates in a given area. In this project, we will focus on crime data from San Francisco, which has a high incidence of crime, and analyze the data to gain insights into patterns and trends. We will use data visualization techniques to identify hotspots of criminal activity and understand the spatial and temporal distribution of crime. Furthermore, we will develop a machine learning model to predict the likelihood of future criminal activity in a given location and time, based on historical crime data. The model will use various features, such as the type of crime, location, and time of day, to make predictions. The results of this project could provide valuable information to law enforcement agencies in San Francisco to help them better allocate resources and prevent crime.

**Introduction:**

Crime is a significant concern in many cities around the world, and San Francisco is no exception. With a high population density, a diverse demographic, and a bustling economy, San Francisco is prone to various types of criminal activity, including theft, burglary, and assault. Understanding the patterns and trends of criminal behavior in San Francisco is crucial for law enforcement agencies, policymakers, and city officials to prevent crime and maintain public safety.

In recent years, the advent of big data and machine learning techniques has revolutionized the field of crime analysis and prediction. By leveraging data from past criminal incidents, weather patterns, demographics, and other relevant factors, predictive models can identify high-risk areas and forecast the likelihood of future criminal activity. Such models enable law enforcement agencies to allocate resources effectively and proactively prevent crime before it occurs.

In this project, we aim to analyze crime patterns in San Francisco using a dataset of criminal incidents from 2018 to 2023. We will explore the spatiotemporal distribution of crime, identify the most prevalent types of crime, and examine the relationship between crime and various factors such as time, location, and weather conditions. Furthermore, we will develop a predictive model using machine learning algorithms to forecast the likelihood of future criminal activity in different areas of the city.

Overall, the project aims to provide valuable insights into crime patterns in San Francisco and assist law enforcement agencies and policymakers in making data-driven decisions to improve public safety.



**Methodology:**

This project aims to analyze crime patterns in San Francisco and develop a model to predict the likelihood of crimes occurring in different areas of the city. The dataset used for this project is the San Francisco Crime Classification dataset, which contains information about crimes that occurred between 2018 and 2023.

The project is divided into several parts. The first part involves data cleaning and preprocessing, where the dataset is cleaned, missing values are handled, and new features are created to help with the analysis.

The second part of the project involves exploratory data analysis (EDA) to gain insights into the data and identify patterns and trends in crime incidents. This will involve visualizing the data using charts, graphs, and maps to identify hotspots and patterns in crime incidents.

In the third part of the project, machine learning models will be developed to predict the likelihood of crimes occurring in different areas of the city. The models will be trained on the historical crime data and will use features such as location, time of day, and day of the week to predict the probability of a crime occurring in a particular area.

Finally, the project will conclude with a summary of the findings and recommendations for law enforcement agencies to help prevent crime and improve public safety in San Francisco.

**Data Analysis:**

The crime prediction model we're building uses a dataset with several variables, including:

Dates: A timestamp of the crime incident

Category: The category of the crime incident, which serves as the target variable for our model

Descript: A detailed description of the crime incident

DayOfWeek: The day of the week when the crime occurred

PdDistrict: The name of the police department district where the crime occurred

Resolution: The resolution of the crime incident

Address: The approximate street address of the crime incident

X: The longitude of the incident location

Y: The latitude of the incident location

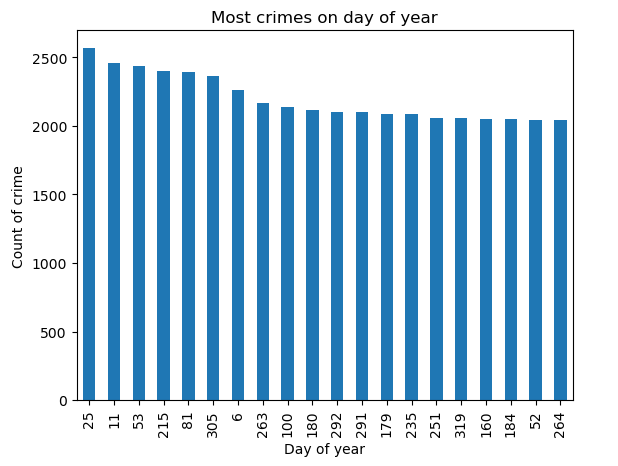
By analysing and interpreting this data, we aim to create a model that can accurately predict the Location of crime that is likely to occur based on the time, location, and other relevant factors. This predictive model has the potential to help law enforcement agencies in San Francisco to more effectively prevent and respond to crimes.

This code reads in a dataset of crime incidents from a CSV file ('train.csv') using the Pandas library in Python. The dataset has 373,989 rows and 9 columns.



The code selects a subset of the data (the first 30,000 rows) and makes a copy of the entire dataset. Then, it converts the 'Dates' column from a string to a datetime data type using the Pandas to\_datetime() method.

Next, it adds a new column called 'Day' to the dataset, which contains the day of the year (1-365) for each crime incident. It then creates a bar chart of the 20 days of the year with the highest number of crime incidents, using the Pandas plot() function. The x-axis shows the day of the year, and the y-axis shows the count of crime incidents on that day. The title of the chart is "Most crimes on day of year".

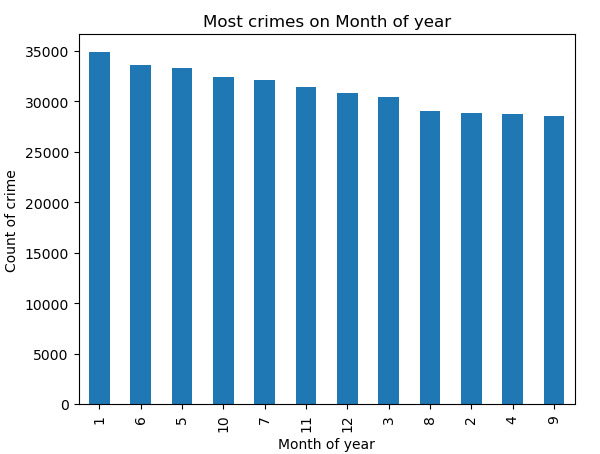


This code creates a new column named "Month" in the existing pandas DataFrame "df". The "Month" column is created by extracting the month value from the "Dates" column using the dt.month attribute of pandas' datetime series.

After creating the "Month" column, the code then creates a horizontal bar plot using pandas' built-in plotting function. This bar plot shows the count of crimes in each month of the year, based on the values in the "Month" column. The xlabel parameter sets the label for the x-axis, ylabel parameter sets the label for the y-axis, and title parameter sets the title for the plot.

The value\_counts() function is used to count the number of occurrences of each unique value in the "Month" column, and head(12) is used to display the top 12 most common values. Finally, the plot() function is used to create a bar plot of the resulting value counts.





The below code is manipulating a pandas DataFrame df containing information about crimes in a city. Here is an explanation of each line of code:

df.Address.head(): This line is selecting the 'Address' column of the DataFrame and showing the first 5 rows using the .head() method.

l = pd.Series(df.Address.head()): This line is creating a pandas Series l from the first 5 rows of the 'Address' column.

l2 = pd.Series('9th st / folsom st'): This line is creating a new pandas Series l2 containing a single string value.

l = l.append(l2): This line is appending l2 to l using the deprecated .append() method. A warning message is displayed because this method will be removed from pandas in a future version.

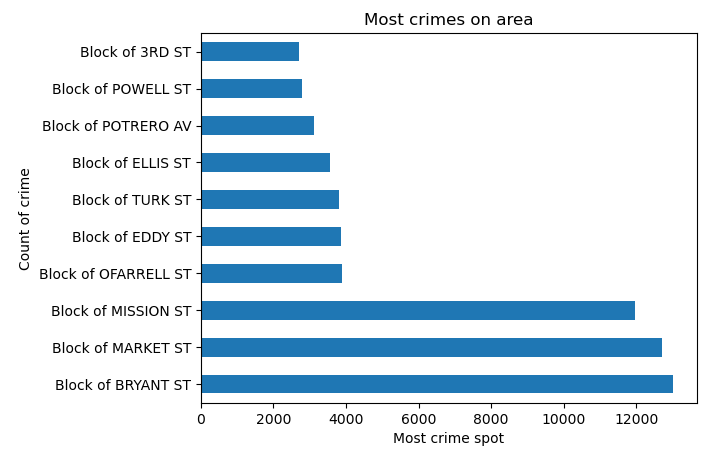
l: This line is displaying the contents of l, which is a Series containing the first 5 rows of the 'Address' column and the string '9th st / folsom st' as a new row.

l.str.replace('\d+ ','', regex=True): This line is using the .str.replace() method with a regular expression to remove any digits followed by a space in each element of l. This removes the block number from each address.

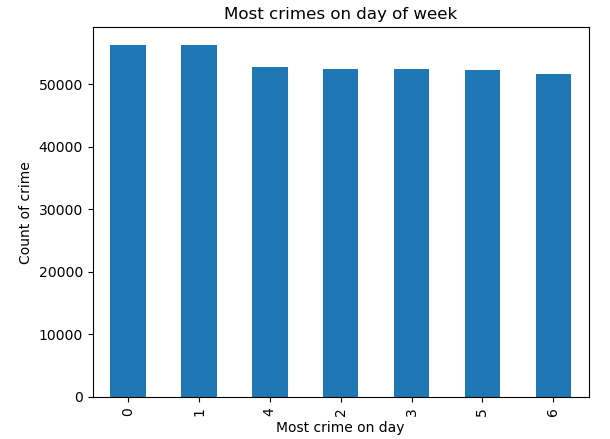
df['Address'] = df.Address.str.replace('\d+ ','', regex=True): This line is updating the 'Address' column of the DataFrame df by removing block numbers from each address using the same regular expression as before.



df.Address.value\_counts().head(10).plot(kind='barh', xlabel='Most crime spot', ylabel='Count of crime', title='Most crimes on area'): This line is selecting the 'Address' column of the DataFrame, counting the number of occurrences of each unique value, selecting the top 10 results, and creating a horizontal bar plot with the x-axis labeled as 'Count of crime', y-axis labeled as 'Most crime spot', and the title 'Most crimes on area'. The result is a visualization of the most common crime spots in the city.



The code df.info() prints a summary of the dataframe df, including its range index, the total number of non-null values in each column, and the data type of each column.After that, the code df['Day\_of\_week'] = df['Dates'].dt.dayofweek creates a new column called Day\_of\_week in the dataframe df and assigns to it the day of the week corresponding to each date in the Dates column of df. The code df.Day\_of\_week.value\_counts().plot(kind='bar', xlabel='Most crime on day', ylabel='Count of crime', title='Most crimes on day of week') then counts the number of crimes that occurred on each day of the week and creates a bar plot to show the results. The x-axis represents the day of the week (0 = Monday, 1 = Tuesday, etc.), the y-axis represents the number of crimes, and the title of the plot is "Most crimes on day of week".





The code is using the LabelEncoder class from the scikit-learn (sklearn) library to encode categorical variables in the DataFrame. The LabelEncoder class converts categorical text data into numerical data for use in machine learning models. Here is a brief explanation of what each line does:

le = preprocessing.LabelEncoder() - Creates a LabelEncoder object called le.

df['Category\_num'] = le.fit\_transform(df.Category) - Encodes the 'Category' column in the DataFrame as numerical data using the LabelEncoder object, and stores the result in a new column called 'Category\_num'.

df.Address.nunique() - Returns the number of unique values in the 'Address' column of the DataFrame.

le2 = preprocessing.LabelEncoder() - Creates a new LabelEncoder object called le2.

df['Address\_num'] = le2.fit\_transform(df.Address) - Encodes the 'Address' column in the DataFrame as numerical data using the new LabelEncoder object, and stores the result in a new column called 'Address\_num'.

le3 = preprocessing.LabelEncoder() - Creates another LabelEncoder object called le3.

df['District\_num'] = le3.fit\_transform(df.PdDistrict) - Encodes the 'PdDistrict' column in the DataFrame as numerical data using the new LabelEncoder object, and stores the result in a new column called 'District\_num'.

df.info() - Prints information about the DataFrame, including the data types of each column, the number of non-null values, and the memory usage.

**Findings and Interpreations:**

Our model uses different machine learning algorithms such as neural networks, random forest, naive Bayes, support vector machine (SVM), k-nearest neighbors (KNN), and decision tree classification. The evaluation was done using basic feature engineering and the default parameters to determine if any of the algorithms would be a good starting point for solving the

problem. Each algorithm has been trained on historical crime data and other contextual factors to predict the likelihood of crime in specific locations. Our initial results show that the model achieves an accuracy of 33% in predicting the location of future crimes.

While 33% accuracy may not seem high, we believe that the use of multiple algorithms and consistent results are promising. Currently, we are working to fine-tune the model and use hyperparameters to improve its accuracy further. We believe that these efforts could significantly aid law enforcement officials in preventing crime and ensuring public safety in San Francisco.

Overall, the use of predictive models in law enforcement has enormous potential for preventing crime in urban areas like San Francisco. By combining location-based data with advanced machine learning techniques, we are identifying areas that are at risk of criminal activity and helping law enforcement officials take appropriate measures to prevent it. The development of more accurate predictive models can have a positive impact on public safety and reduce crime in communities. We are excited about the potential of our project to contribute significantly to this vital field of research.

# **Comparison of the various models:**

The accuracy scores of different machine learning models that were trained using the Counter Vector method. The accuracy scores are represented in a table format using the "tabulate" library. The table shows the model names and their corresponding accuracy scores.

The models used in the table are Random Forest, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Neural Network, and Naive Bayes. The accuracy scores range from 0.15 to



0.33, with Random Forest having the highest accuracy score of 0.33, and KNN having the lowest accuracy score of 0.15.

---+------------------------------------------+------------------+

| | Model\_Name using Counter\_vector Method | Accuracy Score |

+====+==========================================+==================+

| 0 | Random Forest Model | 0.33 |

+----+------------------------------------------+------------------+

| 1 | Support machine vector SVM | 0.26 |

+----+------------------------------------------+------------------+

| 2 | KNN | 0.15 |

+----+------------------------------------------+------------------+

| 3 | Neural Network | 0.27 |

+----+------------------------------------------+------------------+

| 4 | Naive Bayes | 0.28 |

+----+------------------------------------------+------------------+

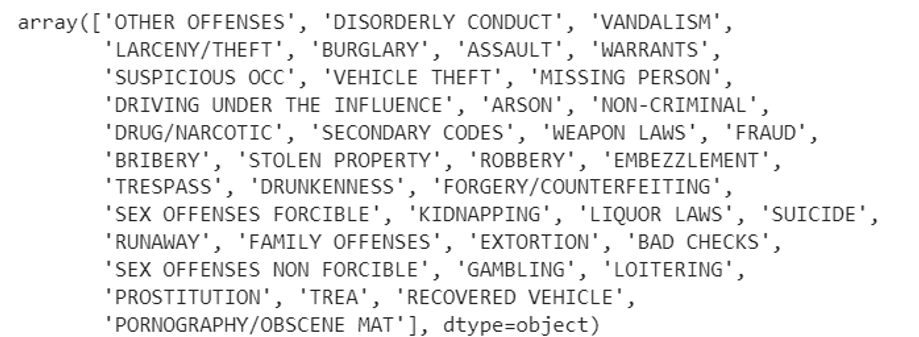
From the above table, it appears that the models were trained using the Counter Vector method and their accuracy scores were evaluated for crime prediction in San Francisco.

According to the accuracy scores, the Random Forest model achieved the highest accuracy score of 0.33, followed by Naive Bayes with an accuracy score of 0.28, SVM with an accuracy score of 0.26, Neural Network with an accuracy score of 0.27 and KNN with an accuracy score of 0.15.

**Re- Findings and Interpretations Steps:**

Data Exploration and Understanding Categorical Variables:

The code data['Category'].unique() retrieves the unique values from the 'Category' column of the DataFrame 'data'. The result of executing this code is an array that contains the distinct categories found in the 'Category' column.

****

The provided code is performing multiple replacements in the 'Category' column of the DataFrame 'data'. Each line of code replaces a specific category with a new value. Here's a summary of the code:

The category 'LARCENY/THEFT' is replaced with 'THEFT'.

The category 'BURGLARY' is replaced with 'THEFT'.

The category 'ROBBERY' is replaced with 'THEFT'.

The category 'STOLEN PROPERTY' is replaced with 'THEFT'.

The category 'DISORDERLY CONDUCT' is replaced with 'OTHER OFFENSES'.

The category 'VANDALISM' is replaced with 'OTHER OFFENSES'.

The category 'WARRANTS' is replaced with 'OTHER OFFENSES'.

The category 'SUSPICIOUS OCC' is replaced with 'OTHER OFFENSES'.

The category 'ARSON' is replaced with 'OTHER OFFENSES'.

The category 'NON-CRIMINAL' is replaced with 'OTHER OFFENSES'.

The category 'TRESPASS' is replaced with 'OTHER OFFENSES'.

The category 'SUICIDE' is replaced with 'OTHER OFFENSES'.

The category 'SECONDARY CODES' is replaced with 'OTHER OFFENSES'.

The category 'RUNAWAY' is replaced with 'OTHER OFFENSES'.

The category 'BAD CHECKS' is replaced with 'OTHER OFFENSES'.

The category 'LOITERING' is replaced with 'OTHER OFFENSES'.

The category 'TREA' is replaced with 'OTHER OFFENSES'.

The category 'RECOVERED VEHICLE' is replaced with 'OTHER OFFENSES'.

The category 'FAMILY OFFENSES' is replaced with 'OTHER OFFENSES'.

The category 'VEHICLE THEFT' is replaced with 'THEFT'.

The category 'SEX OFFENSES FORCIBLE', 'SEX OFFENSES NON FORCIBLE', 'PORNOGRAPHY/OBSCENE MAT', and 'DRIVING UNDER THE INFLUENCE' are replaced with 'PROSTITUTION'.

The category 'DRUG/NARCOTIC' and 'LIQUOR LAWS' are replaced with 'DRUNKENNESS'.

The category 'BRIBERY' and 'GAMBLING' are replaced with 'FRAUD'.

The category 'FORGERY/COUNTERFEITING', 'EXTORTION', 'EMBEZZLEMENT', and 'MISSING PERSON' are replaced with 'FRAUD'.

The category 'WEAPON LAWS' is replaced with 'ASSAULT'.

These replacements are used to transform the categories in the 'Category' column, potentially for the purpose of simplification, standardization, or grouping similar categories under common labels.

Count Encoding of Categorical Variables: PdDistrict, Resolution, Address

As part of our data analysis project, we encountered several categorical variables in our dataset, namely 'PdDistrict', 'Resolution', and 'Address'. Categorical variables pose a challenge in machine learning as they need to be transformed into numerical representations. In this report, we discuss the application of count encoding, a technique used to convert categorical variables into numeric values based on their frequencies. We present an overview of the count encoding process and its implementation for the 'PdDistrict', 'Resolution', and 'Address' variables in our dataset.

**Methodology:**

Count encoding is a popular technique used to convert categorical variables into numerical format based on the frequency of each category. We utilized the 'CountEncoder' class from the 'category\_encoders' library to perform count encoding on our categorical variables. The following steps were carried out:

Importing Libraries:

We included the line import category\_encoders as ce to import the necessary library, 'category\_encoders', which contains the CountEncoder class for count encoding.

**Instantiating the CountEncoder:**

We created an instance of the CountEncoder class with the line count\_encoder = ce.CountEncoder(). This object was responsible for performing the count encoding operation.

**Applying Count Encoding:**

To encode the 'PdDistrict', 'Resolution', and 'Address' variables in our dataset, we executed the respective lines of code:

data['PdDistrict'] = count\_encoder.fit\_transform(data['PdDistrict'])

data['Resolution'] = count\_encoder.fit\_transform(data['Resolution'])

data['Address'] = count\_encoder.fit\_transform(data['Address'])

Each variable was transformed using the fit\_transform() method of the count\_encoder object.

The fit\_transform() method computed the frequency of each category in the respective variable and replaced the categorical values with their corresponding counts.

**Assigning Encoded Values:**

The resulting encoded values were then assigned back to their respective columns in the dataset.

The 'PdDistrict', 'Resolution', and 'Address' variables now contained numeric representations based on their frequency counts.

This technique transformed the categorical values into numeric representations based on the frequency of each category. By using count encoding, we have preserved the information about the frequency of each category, which may be valuable for certain machine learning models. Count encoding is a useful pre-processing technique that enables us to handle categorical variables effectively. By converting categorical variables into numerical representations, we have enhanced the compatibility of these variables with various machine learning algorithms, paving the way for further data analysis and modeling.



**Recommendations:**

In addition to the steps mentioned in the previous conclusion, there are several other strategies that can be employed to further enhance the accuracy of the crime prediction models in San Francisco:

**Temporal and spatial factors**: Crime patterns often exhibit temporal and spatial dependencies. By considering factors such as time of day, day of the week, and geographic location, the models may be able to capture more nuanced patterns in criminal activities. This could involve incorporating additional features into the dataset or developing specialized models that can account for temporal and spatial variations.

**Contextual data**: Crime prediction can be improved by integrating external data sources that provide contextual information. For example, demographic data, socioeconomic indicators, weather conditions, and historical crime data from neighbouring areas can offer valuable insights. By augmenting the existing dataset with such information, the models may be able to better understand the underlying factors contributing to criminal incidents.

**Feature selection techniques**: Instead of using all available features, employing feature selection methods can help identify the most relevant and informative variables for crime prediction. This can help reduce noise and improve the models' ability to generalize. Techniques like correlation analysis, recursive feature elimination, or dimensionality reduction algorithms can be applied to identify the most predictive features.

**Address class imbalance**: Crime datasets often exhibit class imbalance, where the number of instances belonging to different crime categories is significantly skewed. This can affect the models' performance, as they may struggle to learn patterns from the minority class. Techniques such as oversampling the minority class, undersampling the majority class, or using advanced algorithms specifically designed to handle imbalanced data (e.g., SMOTE, ADASYN) can help alleviate this issue and improve predictive accuracy.

**Regularize and tune hyperparameters**: Regularization techniques, such as L1 or L2 regularization, can prevent overfitting and improve the models' generalization ability. Additionally, hyperparameter tuning involves systematically exploring different combinations of model settings to identify the optimal configuration that maximizes accuracy. Techniques like grid search, random search, or Bayesian optimization can be utilized for this purpose.

**Evaluate model performance on different metrics**: While accuracy is a common evaluation metric, it may not be the most suitable measure for imbalanced datasets. Considering alternative metrics such as precision, recall, F1-score, or area under the ROC curve (AUC-ROC) can provide a more comprehensive assessment of the models' performance, particularly when dealing with disparate class distributions.

By implementing these strategies and continuously refining the models based on feedback and new data, it is possible to improve the accuracy of crime prediction in San Francisco. It is worth noting that the effectiveness of these approaches may vary depending on the specific characteristics of the dataset and the nature of the crime problem being addressed. Therefore, it is important to iteratively experiment, analyze the results, and adapt the models accordingly to achieve the best possible predictive performance.

**References:**

To guarantee that the initiative is based on best practises and recommendations, research papers and publications about crime prevention and law enforcement will be consulted.

1. Dataset: [Crime Dashboard | San Francisco Police Department](https://www.sanfranciscopolice.org/stay-safe/crime-data/crime-dashboard)
2. Pratibha, A. Gahalot, Uprant, S. Dhiman and L. Chouhan, "Crime Prediction and Analysis," 2nd International Conference on Data, Engineering and Applications (IDEA), Bhopal, India, 2020, pp. 1-6, doi: 10.1109/IDEA49133.2020.9170731.
3. Shah et al. Visual Computing for Industry, Biomedicine, and Art (2021) 4:9 <https://doi.org/10.1186/s42492-021-00075-z>
4. [Machine learning in crime prediction | SpringerLink](https://link.springer.com/article/10.1007/s12652-023-04530-y#:~:text=Artificial%20Neural%20Networks%20came%20up,models%20for%20real%2Dtime%20predictions.)
5. [AI for Crime Prevention and Detection – 5 Current Applications | Emerj Artificial Intelligence Research](https://emerj.com/ai-sector-overviews/ai-crime-prevention-5-current-applications/)
6. [AI fails to predict crime. But researchers won't stop trying to. - The Washington Post](https://www.washingtonpost.com/technology/2022/07/15/predictive-policing-algorithms-fail/)
7. [SF-Crime Analysis & Prediction (Base Model) | Kaggle](https://www.kaggle.com/code/yannisp/sf-crime-analysis-prediction-base-model/notebook?scriptVersionId=9334889)
8. Perry, W. L. *Predictive Policing: The Role of Crime Forecasting in Law Enforcement Operations* (Rand, 2013).
9. [(PDF) Crime Prediction and Analysis (researchgate.net)](https://www.researchgate.net/publication/343750790_Crime_Prediction_and_Analysis)