Forecasting the Trends and Patterns of Crime in San Francisco using Machine Learning Model

Saad Rasool, Azhar Ghafoor, Zeshan Fareed

Abstract—The Crime Prediction Method is a systemic approach to preventing the detection and review of past crime data patterns and trends and uses it for future crime forecasting. This system is designed to predict the type of crime in a given region that is highly likely to occur and can view areas susceptible to crime. Crime analyzers are helping law enforcement officers to solve crimes with the growing advent of computerized systems. Using machine learning methods, we can derive information from past crime data that is unknown and useful before and can predict future crimes. Here we are working to build a machine learning procedure between computer science and criminal justice, so that crimes can be resolved faster. Instead of focusing on the causes of crime, such as crime, political hostility, etc., we concentrate mostly on the crime factors of every day.

Index Terms—crimes, data, dataset, deep learning, machine learning, pattern, prediction, RNN, machine learning

1 Introduction

Crime is a colossal social issue worldwide, which damages our growth both economically and socially. To diminish the increase of outbreaks of violence and all kinds of crimes, the law enforcement agencies require a sort of futuristic automated tools that can help them to predict geographic information of future crime, types of crime, and the reasons behind the acts to protect the society and upgrade the crime analytics. It is better to prevent a crime than to investigate what or how it has happened. Just as a child is vaccinated for the prevention of illness, prevention systems to prevent crime in the world today have become necessary with such an increased crime rate and brutal crime [2]. To predict crime categories, we will make a prediction model, train it on historical crime data and test it on test data to get higher accuracy. Then prepare the model again on test data also for achieving even higher accuracy.

2 RELATED WORK

Prior to machine learning, historical trends were extrapolated. Also more complicated approaches, including multivariate regression, improved predictive capacity yet only dependent on historical criminal knowledge[1]. Other factors influence and do not take into account the rate of crime, such as weather and socio-economic factors. The use of other datasets greatly increases the number of features.

- *Saad Rasool is currently pursuing masters degree program in Hohai Universit Nanjing, China, PH-+86 188 51668513. E-mail: saadrasool6@gmail.com*
- *Azhar Ghafoor is currently pursuing masters degree program in COMSATS University Islamabad, Pakistan, PH-+92 305 5125974. E-mail: azharghafoor39@gmail.com*
- *Zeeshan Fareed is currently pursuing bachelors degree in Jiangsu Normal University China, PH-+92 3366420354 E-mail: zeeshan7250@gmail.com*

The cost of these machine learning techniques was significantly reduced by the high-performance computer growth. In many fields, they are very efficient. Deep learning has reached remarkable results with many forms of classification problems from speaking up to visual recognition, as a relatively recent advancement of artificial intelligence. Crime prediction is an area with little emphasis in deep learning. Deep knowledge is well suited for managing the time and space components of the problem and a robust feature set that can be developed using applicable public datasets as well as criminal datasets[2]. The intrinsic ambiguity of the situation makes it perfect for profound learning.

The use of neural networks eliminates the need for robust functionality in earlier work and facilitates training on large datasets. To discuss the temporal dimensions of crime prevention, use the Recurrent Neural Network (RNN)[3]. It is presumed that all inputs and outputs are independent in a typical neural network, but this is not the case for crime.

Today would be influenced by crimes that occurred yesterday. For instance, if robbery rises, the area will be targeted and the robbery rates for some time remain higher than average. For each sequence element, RNNs apply the same weights when they are saved in the line from previous states. This memory is used to encode the sequence (also called the hidden state). This implies that the output is calculated by the current input and hidden history, which allows more predictive ability[4].

3. PROPOSED SOLUTION

To predict crime categories, we will make a prediction model and train it on historical crime data and test it on test data to get higher accuracy. Then introduce the model again on test data also for achieving even higher accuracy.

3.1Data Collection

The data used in this research project are the data on crime in San Francisco made available on the website, part of the "open data" initiative, by the San Francisco police department. Incidents from the SFPD crime reporting system are included in this dataset[5]. Data from 1 January 2003 to 2 February 2018. This

dataset has the following attributes: the dataset has about 2.1 million rows, and the dataset is about 450 MB in size. Data from 2003 to May) 2018 are provided here.

3.2 Crime Prediction System Architecture

Various frameworks can implement this prediction system, but we use a simple architecture with different technologies to implement this prediction system.

3.3 Machine Learning Model

Fig (1) Machine Learning Model

For creating, we first did the preprocessing of the dataset and then featured engineering.

3.3.1 Preprocessing and Feature Engineering

As there is no missing value for any feature so, we did not eliminate any row and divided the data in train and test dataset using skLearn library. Then we implement the following steps:

1. Drop the Resolution Column and Location Column:

```
train_df = train_df.drop('Resolution', axis=1)
train_df = train_df.drop('Location', axis=1)
test_df = test_df.drop('Resolution', axis=1)
test_df = test_df.drop('Location', axis=1)
```

2. Parsing the 'Date' Column because the 'Date' column type is String. It will be easier to work with by parsing it to Date &time.

```
train_df['Date'] = pd.to_datetime(train_df.Date)
test_df['Date'] = pd.to_datetime(test_df.Date)
train_df.sample(1)
```

3. Engineer a feature 'Date' to indicate whether the crime was committed by day or by night.

```
train_df['IsDay'] = 0
train_df.loc[ (train_df.Date.dt.hour > 6) & (train_df.Date.dt.hour < 20), 'IsDay' ] = 1
test_df['IsDay'] = 0
test_df.loc[ (test_df.Date.dt.hour > 6) & (test_df.Date.dt.hour < 20), 'IsDay' ] = 1
```

4. Create 'Month', 'Year' and 'DayOfWeekInt' columns by Encoding 'DayOfWeek' to Integer.

```
days_to_int_dic = {
        'Monday': 1,
        'Tuesday': 2,
        'Wednesday': 3,
        'Thursday': 4,
        'Friday': 5,
        'Saturday': 6,
        'Sunday': 7,
}
train_df['DayOfWeek'] = train_df['DayOfWeek'].map(days_to_int_dic)
test_df ['DayOfWeek'] = test_df ['DayOfWeek'].map(days_to_int_dic)
```

5. We are creating Hour, Month and Year Columns from the 'Date' column.

```
train_df['Hour'] = train_df.Date.dt.hour
train_df['Month'] = train_df.Date.dt.month
train_df['Year'] = train_df.Date.dt.year
train_df['Year'] = train_df['Year'] - 2000

test_df['Hour'] = test_df.Date.dt.hour
test_df['Month'] = test_df.Date.dt.month
test_df['Year'] = test_df.Date.dt.year
test_df['Year'] = test_df['Year'] - 2000 #
```

6. Deal with the cyclic characteristic of Hours, Months and Days of Week.

```
train_df['HourCos'] = np.cos((train_df['Hour']*2*np.pi)/24 )
train_df['DayOfWeekCos'] = np.cos((train_df['DayOfWeek']*2*np.pi)/7 )
train_df['MonthCos'] = np.cos((train_df['Month']*2*np.pi)/12 )

test_df['HourCos'] = np.cos((test_df['Hour']*2*np.pi)/24 )
test_df['DayOfWeekCos'] = np.cos((test_df['DayOfWeek']*2*np.pi)/7 )
test_df['MonthCos'] = np.cos((test_df['Month']*2*np.pi)/12 )

train_df.sample(1)
```

7. Dummy Encoding of 'PdDistrict'

```
days_to_int_dic = {
        'Monday': 1,
        'Tuesday': 2,
        'Wednesday': 3,
        'Thursday': 4,
        'Friday': 5,
        'Saturday': 6,
        'Sunday': 7,
}
train_df['DayOfWeek'] = train_df['DayOfWeek'].map(days_to_int_dic)
test_df ['DayOfWeek'] = test_df ['DayOfWeek'].map(days_to_int_dic)
```

8. Label Encoding of 'Category'

```
from sklearn.preprocessing import LabelEncoder

cat_le = LabelEncoder()
train_df['CategoryInt'] = pd.Series(cat_le.fit_transform(train_df.Category))
test_df['CategoryInt'] = pd.Series(cat_le.fit_transform(test_df.Category))
train_df.sample(5)
#cat_le.classes_
```

3.3.2 Feature Selection

We have selected the feature that is useful for prediction. Dataset had the following part after preprocessing.

```
Index(['Date', 'DayOfWeek', 'Address', 'X', 'Y', 'Category', 'Descript',
       'IsDay', 'Hour', 'Month', 'Year', 'HourCos', 'DayOfWeekCos', 'MonthCos',
       'PdDistrict_BAYVIEW', 'PdDistrict_CENTRAL', 'PdDistrict_INGLESIDE',
       'PdDistrict_MISSION', 'PdDistrict_NORTHERN', 'PdDistrict_PARK',
       'PdDistrict_RICHMOND', 'PdDistrict_SOUTHERN', 'PdDistrict_TARAVAL',
       'PdDistrict_TENDERLOIN', 'CategoryInt', 'InIntersection'],
      dtype='object')
```

For model training, we had selected the following features:

```
feature_cols = ['X', 'Y', 'IsDay', 'DayOfWeek', 'Month', 'Hour', 'Year', 'InIntersection',
                'PdDistrict_BAYVIEW', 'PdDistrict_CENTRAL', 'PdDistrict_INGLESIDE',
                'PdDistrict_MISSION', 'PdDistrict_NORTHERN', 'PdDistrict_PARK',
                'PdDistrict_RICHMOND', 'PdDistrict_SOUTHERN', 'PdDistrict_TARAVAL', 'PdDistrict_TENDERLOIN']
target_col = 'CategoryInt'

train_x = train_df[feature_cols]
train_y = train_df[target_col]

test_x = test_df[feature_cols]
test_y = test_df[target_col]
```

3.3.3 Machine Learning Algorithm

Gradient boosting is a method of regression and classification, generating a prediction model in several poor models usually policymakers. It sets up the model as other stimulation approaches and generalizes this model by optimization of the arbitrary differentiating loss function. We used the XGBoost library to implement this algorithm. For increasing distributed gradients, XGBoost is a very efficient and versatile portable library. Under the gradient improvement paradigm, it implements machine learning algorithms. In order to solve many problems in information science easily and accurately, XGBoost provides a parallel tree extension (also known as GBDT).

3.3.4 Model Creation

First, we imported the XGBoost library and created DMatrices

```
import xgboost as xgb
train_xgb = xgb.DMatrix(train_x, label=train_y)
test_xgb  = xgb.DMatrix(test_x)
```

Then we set parameters with proper values and did cross-validation

```
params = {
    'max_depth': 4,  # the maximum depth of each tree
    'eta': 0.3,  # the training step for each iteration
    'silent': 1,  # logging mode - quiet
    'objective': 'multi:softprob',  # error evaluation for multiclass training
    'num_class': 39,
}
```

```
CROSS_VAL = False
if CROSS_VAL:
    print('Doing Cross-validation ...')
    cv = xgb.cv(params, train_xgb, nfold=3, early_stopping_rounds=10, metrics='mlogloss', verbose_eval=True)
```

The primary step was fitting (training) and model and making predictions

```
SUBMIT = not CROSS_VAL
if SUBMIT:
    print('Fitting Model ...')
    m = xgb.train(params, train_xgb, 10)
    res = m.predict(test_xgb)
    cols = ['Id'] + cat_le.classes_
    submission = pd.DataFrame(res, columns=cat_le.classes_)
    submission.to_csv('submission.csv', index=False)
    print('Done Outputing !')
    print(submission.head(3))
else:
    print('NOT SUBMITING')
```

After formatting the results in a suitable format, we stored the MySQL database's prediction using MySQL-connector-Python. We accessed that stored results through services to our application user interface.

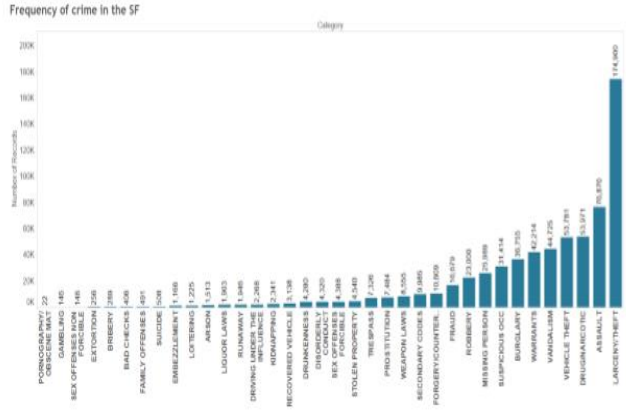
```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  passwd="password",
  database="mydatabase"
)
```

4 Analysis of Crime

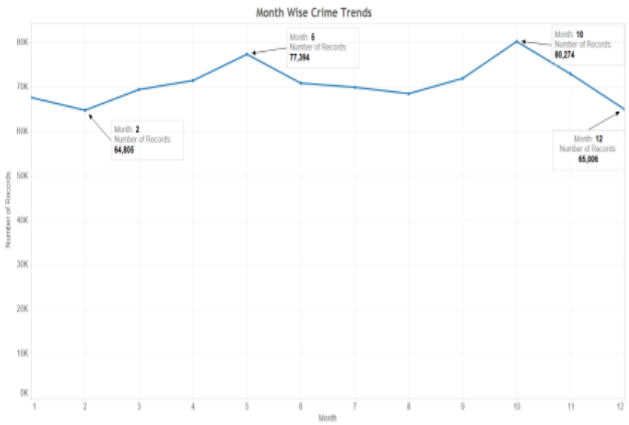
4.1 Trends

Thirty-nine incident reporting types are available in the dataset, including other offences" and "Non-criminal." Robbery and theft was San Francisco's most prevalent crime between 1 January 2003 and 13 May 2018. The following high crime categories: larceny and robbery, assault, drug trafficking, car theft, vandalism, drug usage, and burglary, are the main focus of the study.



4.2 Monthly distribution

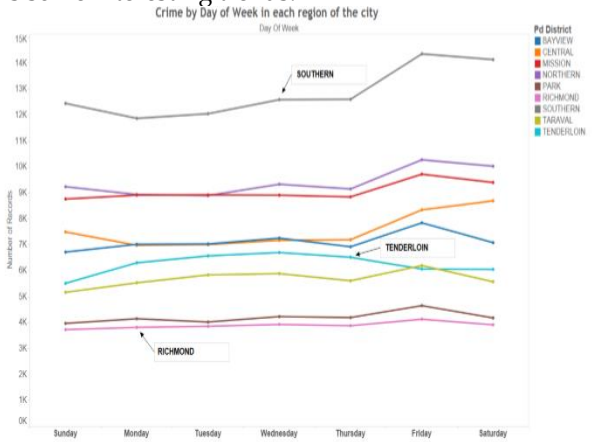
Since the 2018 crime incidents are only up to May, certain records have been omitted for this review. February's month is the best, and the least crime incidents are registered. October is the least safe of recorded crime incidents.



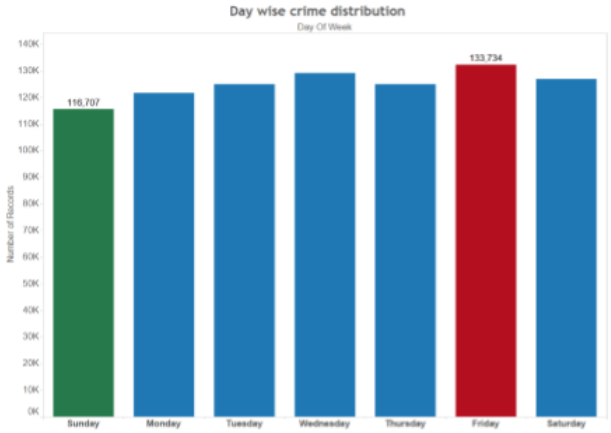
4.3 Distribution region-wise

A contour plot describing the region-wise crime density using the latitude/ longitude details of the recorded crime is as follows:

A giant hotspot can be found in the Southern and Tenderloin area with comparatively less dense plots in the neighborhoods. Most crime appears to be concentrated in these and nearby areas. Plotting a line graph reflecting crime in each area on a weekday shows some interesting trends:



Crime rate by day of the week:

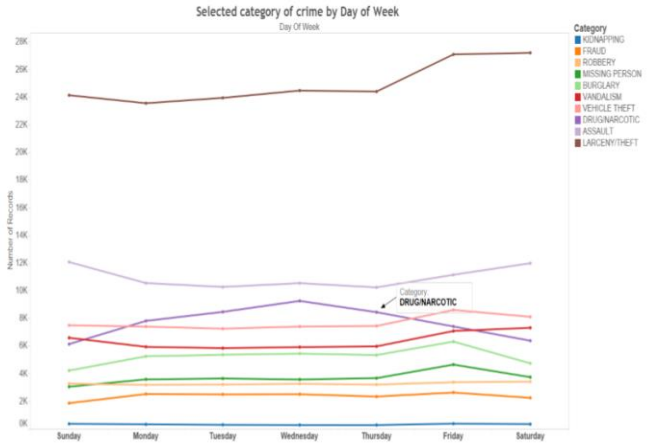


The Southern region is the most infamous, with reported maximum number of crimes, followed by the Northern area. Richmond's San Francisco's safest spot.

The crime rate will grow over the weekend. Apparently the highest number of recorded accidents on Fridays. Nearly all regions will see the same pattern. However, Tenderloin is an exception, as this region's crime rate falls over the weekend and is the highest on Wednesdays. Wednesday, Friday and Saturday are big crime days. Interestingly, fewer accidents were registered on Mondays, followed by Sundays. In the Southern, Northern, Central and Mission districts, the number of crimes on Friday and Saturday increased dramatically and decreased for the rest of the week.

4.3.1 Specific crime trends

A handful of crimes (kidnapping, fraud, robbery, missing person, arson, vandalism, auto theft, drug/narcotics, assault and larceny/theft) are investigated. Following patterns are noted:



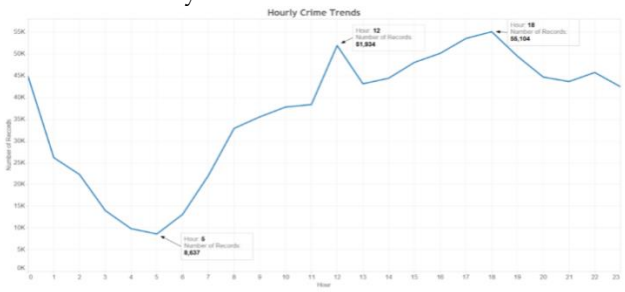
Interestingly, drug-related crimes decrease over the weekend (unlike other categories). Recorded accidents are the highest on Friday.

Categories of high crime include burglary and theft on Friday and Saturday. On the other hand, attack incidents gradually increased from Thursday to Sunday, while burglary usually occurred more frequently than weekends. Drug crimes were more frequently reported on Wednesday and less reported on

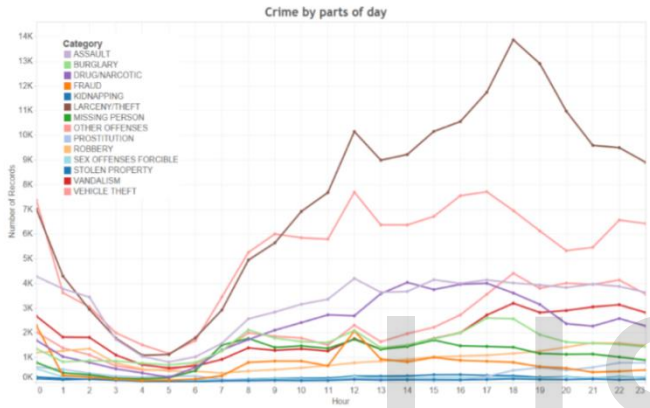
weekends, and daily robberies occurred at around the same level.

4.3.1.1 Hourly Crime Analysis

The overall hourly crime trends are as follows:

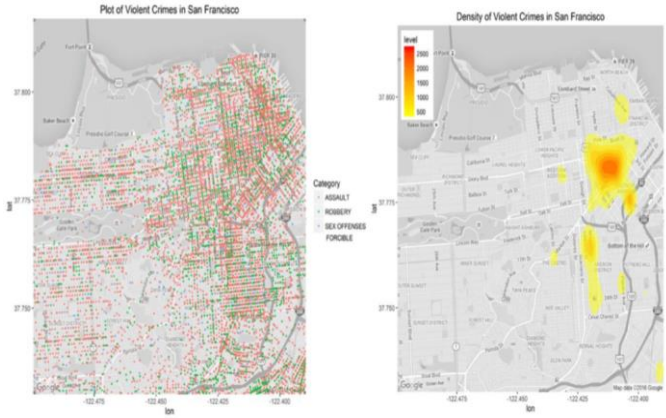


Hourly trend analysis for specific crimes:



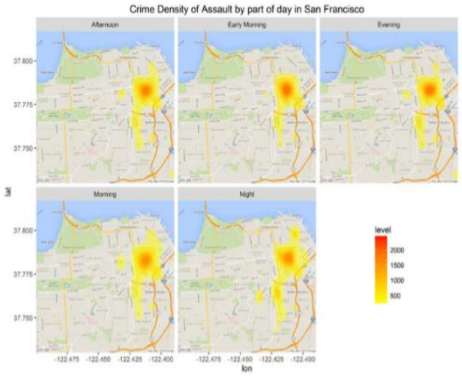
Hourly developments unravel some fascinating facts about crime 05 am the day's safest part and 6 pm the most dangerous hour. Surprisingly, 12 pm is the second most dangerous hour of the day which is the hour of highest accidents recorded in certain crime categories. Kidnapping and stolen property incidents take place uniformly throughout the day.

4.4 San Francisco Violent Crimes: Assault, Robbery and Sex (forcible) Distribution



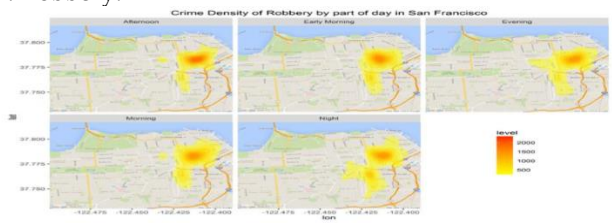
4.4.1 Density of Violent crime (each) by parts of day:

1. Assault:

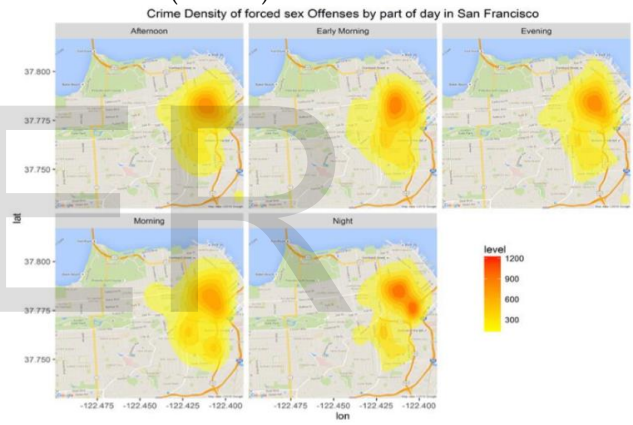


Part of Day	Time Frame
Early Morning	4am to 8am
Morning	8am to 12pm
Afternoon	12pm to 5 pm
Evening	5pm to 9pm
Night	9pm to 4am

2. Robbery:

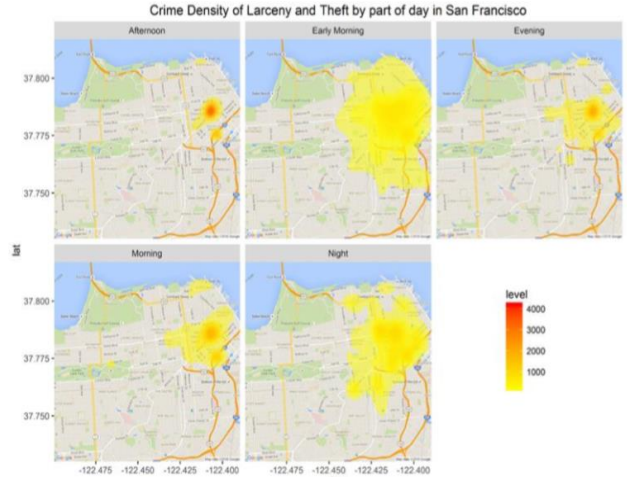


3. Sex Offences (forcible):



Different patterns with an increase in density at night can be seen at different times.

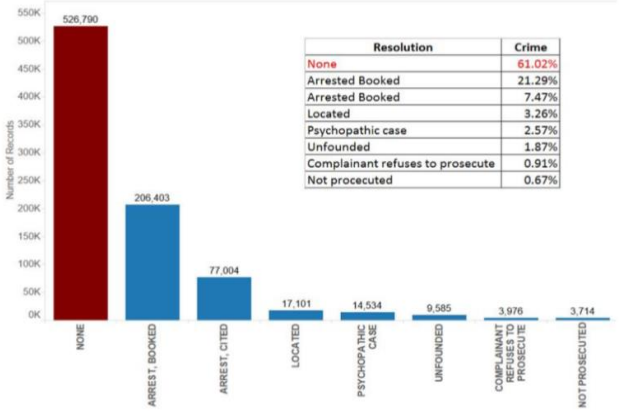
1. Larceny and Theft:



In the early morning and at night, crime is distributed in many areas. On the afternoon it is concentrated in one area.

4.4.2 Food for thought:

Following is a bar chart which shows the percentage of incidents investigated by law enforcement authorities compared with those which did not occur. Surprisingly, no definitive action has been taken on 61.02 percent of the reported incidents.



Resolution	Crime
None	61.02%
Arrested Booked	21.29%
Arrested Booked	7.47%
Located	3.26%
Psychopathic case	2.57%
Unfounded	1.87%
Complainant refuses to prosecute	0.91%
Not procecuted	0.67%

5 Conclusion

A detailed analysis has been conducted in this project on different crimes in San Francisco. Prediction models (Classifiers) have also been trained with a few machine learning algorithms and the neural network, but gradient boosters are good compared with others.

6 Future Work

It would be interesting as part of the future research to see if additional datasets and crime datasets could be used to make even more class splits. It helps us to see if additional factors such as population data, housing data, weather data, district alphabet rates, and transport data contribute to crime.

Other crime datasets, like the Chicago crime data, are also available for testing on the same model and are another famous dataset. How the crimes in other towns compare with the crimes of San Francisco would be interesting to see.

References

[1] S. R. Bandekar and C. Vijayalakshmi, "Design and analysis of machine learning algorithms for the reduction of crime rates in India," *Procedia Comput. Sci.*, vol. 172, pp. 122–127, 2020, doi: 10.1016/j.procs.2020.05.018.

[2] P. Stalidis, T. Semertzidis, and P. Daras, "Examining deep learning architectures for crime classification and prediction," *arXiv*, pp. 1–12, 2018.

[3] R. R. Shah, "Crime Prediction Using Machine Learning," pp. 4–6, 2017.

[4] A. Bharati and S. Rak, "Crime Prediction and Analysis Using Machine Learning," *Int. Res. J. Eng. Technol.*, vol. 5, no. 9, pp. 1037–1042, 2018, [Online]. Available: www.irjet.net.

[5] N. Shama, "A Machine Learning Approach to Predict Crime Using Time and Location Data," pp. 1–52, 2017, [Online]. Available: http://dspace.bracu.ac.bd/xmlui/handle/10361/8197.