PROJECT REPORT

VIGILANTES | CMPE 255-01: DATA MINING

TEAM - VIGILANTES

AASHISH SUBRAMANIAN [011818755]
CHINTAN VACHHANI [011447696]
SUDHEER SAMMETA [011545170]

INTRODUCTION

Crime prevention and resolution are major issues faced by governments and law enforcement agencies around the world. Observing an increase in crime frequency around the university, we realized that majority of the crimes could be prevented if the victims had better information to help them stay vigilant. To help provide such information, we believe that current Data Mining techniques can be used on publicly available datasets to extract crime patterns. This knowledge can be used to provide insight into a threat level depicting the possibility of crime occurrence in the vicinity of the user and in turn, alert them to stay safe. The crime prediction problem is an age old one, recent papers use various configurations of machines learning algorithms to find accurate solutions. In this report, we provide a different approach by integrating public infrastructure datasets to classify the threat level as low, moderate or high.

SYSTEM DESIGN AND IMPLEMENTATION

ALGORITHM

The following algorithms were used to develop three different models:

a. KNN Classifier:

The KNN Classifier algorithm was selected as it helps in achieving the desired goal of the model, which is to consider the user location, time, date ,and district information and find the the similarity to existing known crime set features to predict the most likely label for threat levels.

b. RandomForest:

The RandomForest Classifier was selected as it is an ensemble method that generates multiple decision trees. The given dataset seemed to be perfectly suitable for a tree based algorithm and hence the idea was to use this ensemble approach to get the best results.

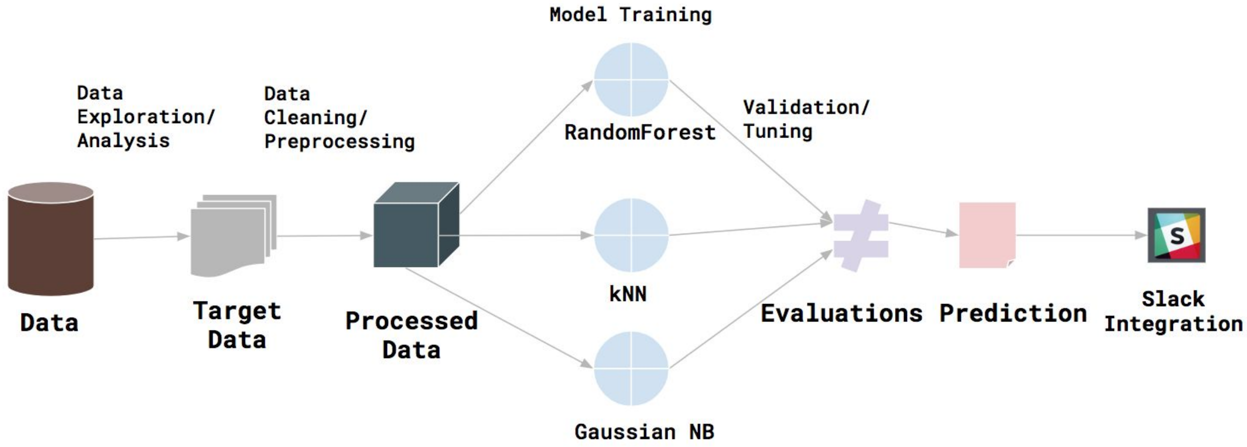
c. Gaussian Naive Bayes:

This algorithm was selected to find the fit of a probability based model on the dataset. It was found that the data did adhere to the Gaussian distribution and hence provided fairly reasonable results.

TECHNOLOGIES AND TOOLS USED

- Jupyter notebook was used for data exploration, analysis, cleaning, preprocessing, model training ,and evaluation
- Sublime editor was used to develop the slack integration module
- Primarily, pandas and numpy libraries were used throughout the project
- Sklearn library was used for the model training/testing, and evaluation
- Seaborn and matplotlib were used in the visualization module
- Pickle library was used to save the intermediate datasets and the final model

ARCHITECTURE



The components depicted in the diagram have been described below:

a. Data Gathering:

This component deals with manual exploration of finding datasets available for use. Different sources were searched, and finally, a subset of datasets that would work well with the crime data was selected. In this case, static infrastructure information has been considered and added as features to the San Francisco crime dataset to obtain threat levels in a given location based on crime patterns and the static structures.

b. Data Exploration:

This component deals with exploring the selected datasets to understand the features offered in each of them. The required features are identified which leads to a target dataset which is used in the analysis phase. Elimination of features were based on a feature’s irrelevance to the core dataset which is that of crime.

c. Data Analysis:

The target dataset is analyzed using visualization libraries such matplotlib and seaborn to understand the relationship between the datasets, and justify their selection for the model.

d. Data Cleaning:

The datasets are then cleaned by removing records with missing values. The columns types are changed as per the requirement. In certain datasets, the location information and address information was extracted from single columns to create standard dataset formats that consisted of Name, Address, Latitude and Longitude. These datasets were then saved.

e. Data Preprocessing:

- In this stage, the cleaned datasets were loaded, transformed , and combined to generate a single dataset. The following steps were carried out in this phase:
- The core dataset of crimes in San Francisco was cleaned to remove values with nan
- Column values were transformed to lowercase and labelled as required
- The data values were split into year, month and day to obtain granular features that would help improve classification
- Performed discretization on the time column such that each value falls in one of six time periods that span 4 hours in a day
- Categorized the resolution column to specify if the crime was resolved
- Introduce a label column to reflect the threat level for each of the crime records
- Integrated different datasets by considering the latitude and longitude values

to obtain binary responses that depict the presence and absence of structures considered using the other cleaned datasets. [1 mile radius considered]

- Save the final dataset

f. Model Training:

- The following steps have been performed as part of model training:
- Load the pre-processed dataset
- Perform the label encoding task on columns such as day, district, month, time_interval, label
- Split the dataset into training and test using StratifiedKFold strategy to ensure labels are well represented in both training and test sets
- Fit the data using KNNClassifier, Gaussian Naive Bayes and RandomForest algorithms

g. Validation:

Validation is performed using the 10-fold cross validation approach

h. Tuning/Evaluation:

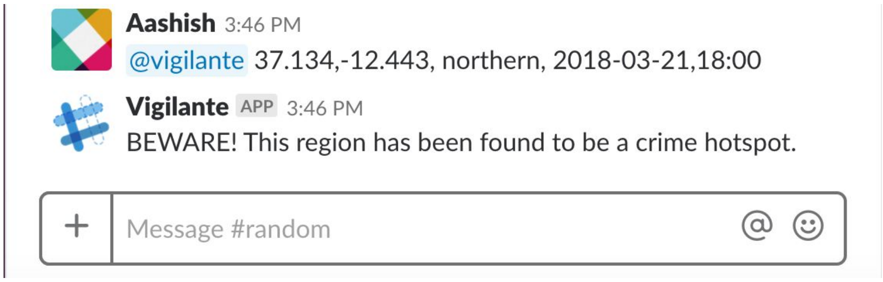
- Tuning is performed using sklearn GridSearch which considers different combination of hyperparameters to provide the one with the best results
- The F1-Score metric has been used to evaluate each model highlighted above
- The best model is then saved using pickle

i. Slack Interface:

- The saved model is loaded and used by a bot service that runs locally
- The input from the user is used to generate a feature vector that can be used with the loaded model
- The model is then used to provide the prediction based on the feature vector
- The prediction is conveyed to the user by the bot, through the Slack application

USE CASE/SCREENSHOTS

The model developed has been integrated with the Slack platform. This provides a bot interface that takes in the user input of location, time, date ,and district which is then used to create a data point. This is fed to the model which provides a prediction based on the inputs. The idea is to have this application available to people in the city of San Francisco, so that they can stay safe as they move around the city. The screenshot below depicts the Slack interface:



Aashish 3:46 PM
@vigilante 37.134,-12.443, northern, 2018-03-21,18:00

Vigilante APP 3:46 PM
BEWARE! This region has been found to be a crime hotspot.

EXPERIMENTS

DATASET AND PRE-PROCESSING

The dataset used was obtained from sfgov.org which provides different public datasets pertaining to the city of San Francisco. The following table describes the datasets considered for this model:

Name	**Select Feature/ Type**	**Size of data**	**Instances**	**Preprocessing**
crime	IncidntNum (int64) Category (object) Descript (object) DayOfWeek (object) Date (object) Time (object) PdDistrict (object) Resolution (object) Address (object) X (float64) Y (float64) Location (object) PdId (int64)	450.8MB	2188068	Removed NaN values Decomposed features Modified column data types Dropped irrelevant columns Aggregated features from other datasets

colleges	latitude (float64) longitude (float64) name (object) address (object)	5kB	46	Removed NaN values Decomposed features Modified column data types Dropped irrelevant columns
landmarks	latitude (float64) longitude (float64) name (object) address (object)	157kB	304	Removed NaN values Decomposed features Modified column data types Dropped irrelevant columns
public_open_spaces	latitude (float64) longitude (float64) name (object) address (object)	41kB		Removed NaN values Decomposed features Modified column data types Dropped irrelevant columns
schools	latitude (float64) longitude (float64) name (object) address (object)	88kB	445	Removed NaN values Decomposed features Modified column data types Dropped irrelevant columns
commuter_stops	latitude (float64) longitude (float64) name (object) address (object)	45kB	159	Removed NaN values Decomposed features Modified column data types Dropped irrelevant columns
facilities	latitude (float64) longitude (float64) name (object) address (object)	336kB	1805	Removed NaN values Decomposed features Modified column data types Dropped irrelevant columns
private_spaces	latitude (float64) longitude (float64) name (object) address (object)	31kB	78	Removed NaN values Decomposed features Modified column data types Dropped irrelevant columns
public_park	latitude (float64) longitude (float64) name (object) address (object)	41kB	230	Removed NaN values Decomposed features Modified column data types Dropped irrelevant columns

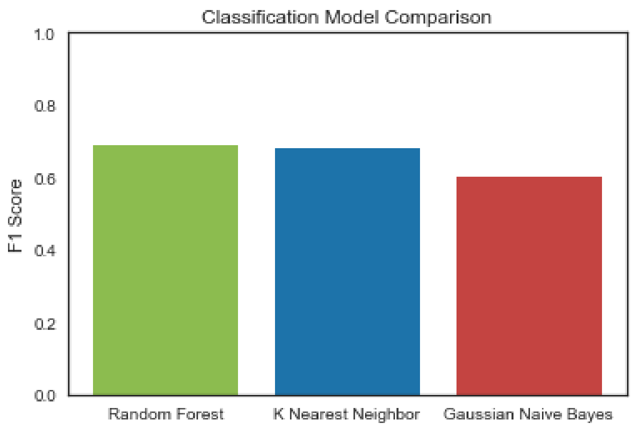
EVALUATION METHODOLOGY

- The dataset was split using a 80:20 ratio from training and test respectively
- The Stratified Split method was used to ensure equal probabilistic representation since the data was imbalanced
- The training set consisted of 1750455 samples and the test set consisted of 437613 samples
- 10-Fold-Cross validation was used on each of the 3 models

- F1 Score metric was used to evaluate the models

ALGORITHM RESULT COMPARISON

The following diagram depicts the comparison between the 3 models selected:



ANALYSIS

- Based on the experiments conducted, similar results were obtained using kNN and RandomForest algorithms for the model
- Gaussian Naive Bayes provided lowest F1-Score with 0.60
- RandomForest provided the best accuracy with 0.69

DISCUSSIONS AND CONCLUSION

DECISIONS

Critical decisions were made for different stages of design and implementation of the model:

- A large number of datasets were initially selected, and some were eliminated based on data exploration results and their relevance to the problem being solved
- Each selected dataset was explored and decisions were taken to eliminate and transform some of the features
- Decisions were made to try different classification models such as kNN, RandomForest and Gaussian Naive Bayes.
- Multinomial Naive Bayesian which was initially considered, was eliminated

due to a requirement to provide non negative values which did not adhere with the given dataset

- Evaluation metrics were decided post data analysis which identified imbalances in the desired labels

DIFFICULTIES

- Difficulties were faced in cleaning the dataset. Each dataset had different data formats for crucial features and the types varied as well
- Merging data from different datasets was time consuming due to the size of the datasets

THINGS THAT WORKED

- The approach for dataset analysis worked well as we gained important information to hand pick the relevant datasets
- The approach to creating models and the evaluation of them allowed us to arrive at the optimal solution
- The planning and implementation was done well as a team. We each were able to accomplish the required tasks as per the decided deadlines and support each other when required

THINGS THAT DIDN’T WORK WELL

- Local development on large datasets was a challenging. It would take a huge amount of time to perform modelling tasks and data integration tasks
- Use of Jupyter Notebook for collaboration during analysis and implementation proved challenging as it fails to support collaborative efforts on a single notebook

CONCLUSION

Based on the models created and the tuning and evaluation strategies selected, we found that the RandomForest model providing an F1-Score of 0.69 worked best for the considered dataset. Although kNN with k = 100 was close with a score of 0.68, we have used the RandomForest model as the predictor for the Slack application.

PLAN AND TASK DISTRIBUTION

The sub-modules along with their owners and respect responsibilities have been provided below:

1.Dataset gathering/Exploration: [Owner: Team]

Responsibilities:

- Explore public datasets to find those that can be used to help provide crime occurrence information
- Select relevant datasets from the dataset pool
- Explore each dataset to identify valuable features
- Identify the features to be used with the crime dataset

2. Data Analysis: [Owner: Sudheer Sammeta]

Responsibilities:

- Use the selected datasets to come up with visualizations that depict the relation between crimes and the features selected
- Eliminate features that fail to show convincing relations
- Provide insight into patterns in the occurrence of crime using the crime dataset

3. Data Cleaning: [Owner: Aashish Subramanian]

Responsibilities:

- Use the exploration information to clean the datasets
- Change features to desired data types
- Save each of the cleaned datasets

4. Data Preprocessing: [Owner: Chintan Vachhani]

Responsibilities:

- Eliminate irrelevant features from the crime dataset
- Perform transformation of column data to obtain granular features

- Transform categorical features using label encoding
- Binarize features
- Combine all cleaned datasets to obtain the final dataset
- Save final dataset for model build step

5. Build model: [Owner: Team]

Responsibilities:

- Use the preprocessed data to find suitable models that provide accurate predictions of threat level

6. Tuning/Evaluation: [Owner: Chintan Vachhani]

Responsibilities:

- Use cross validation techniques and Grid Search to obtain a tuned model
- Experiment with different hyperparameters

7. **Integration with frontend (Slack):** [Owner: Aashish Subramanian]

Responsibilities:

- Build Slack app to use the final tuned model to provide predictions for a user
- Integrate application with a Slack channel to test predictions

8. Documentation/Slides: [Owner: Team]

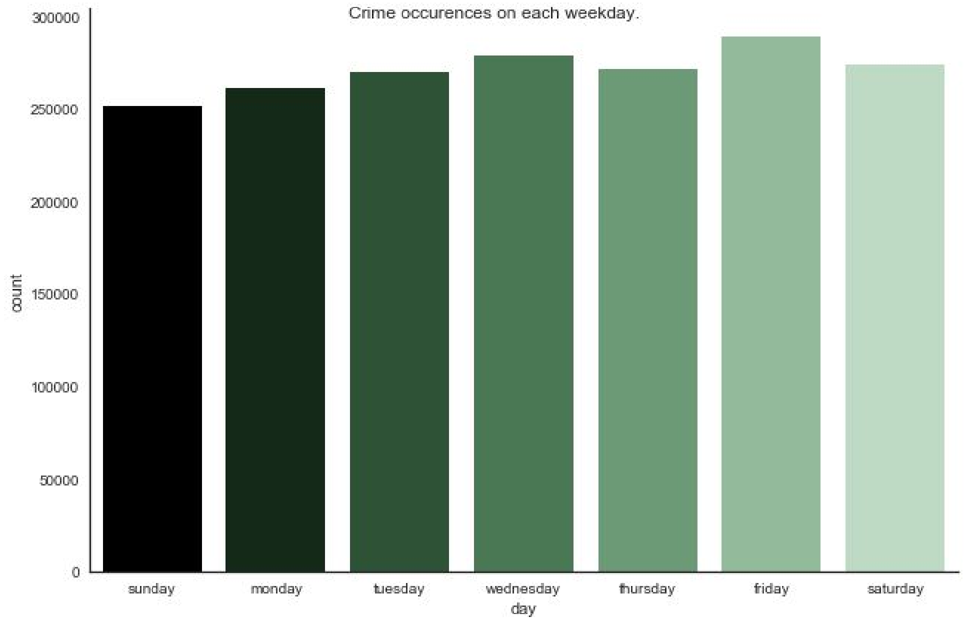
Responsibilities:

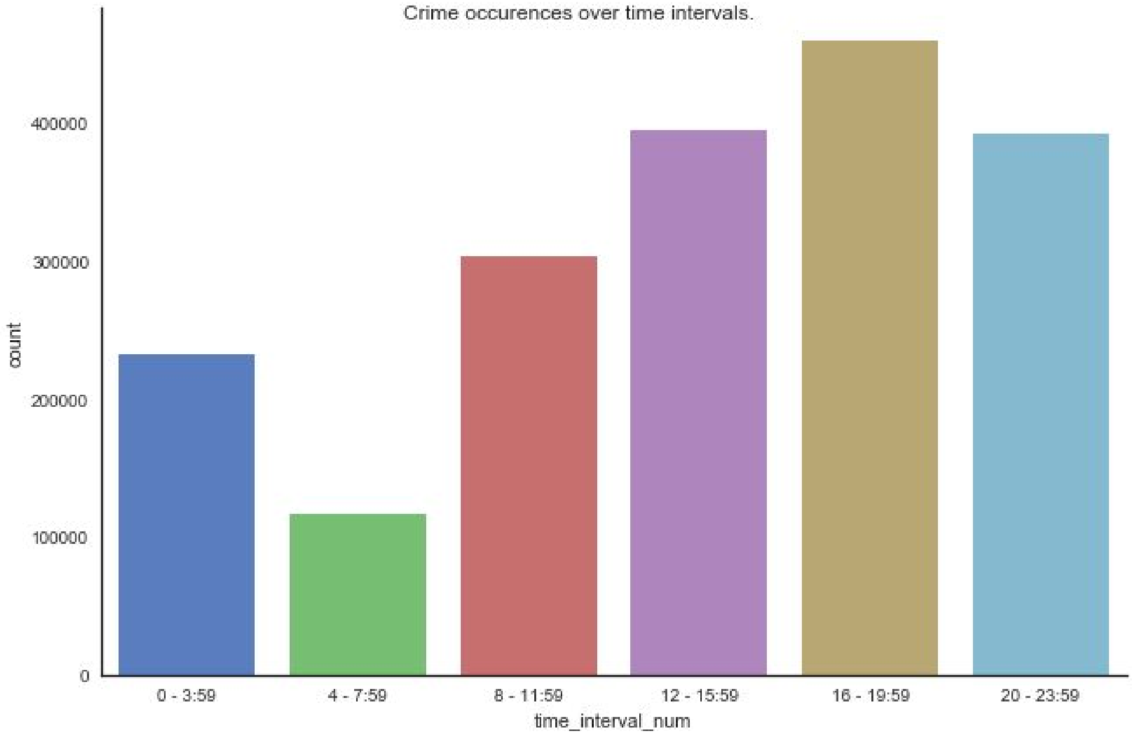
- Complete the report as per the requirement
- Complete the presentation slides

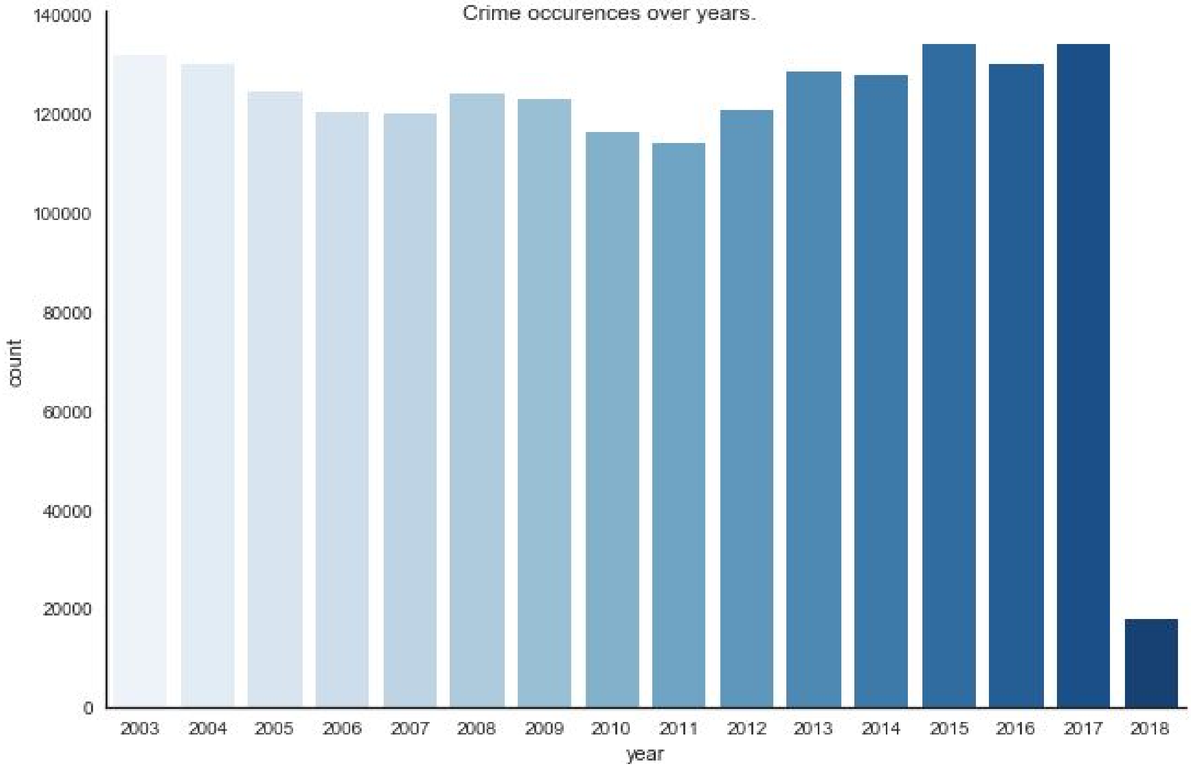
Each of the above tasks were successfully accomplished by the owners. In addition, the team worked together and helped each other when required.

APPENDIX

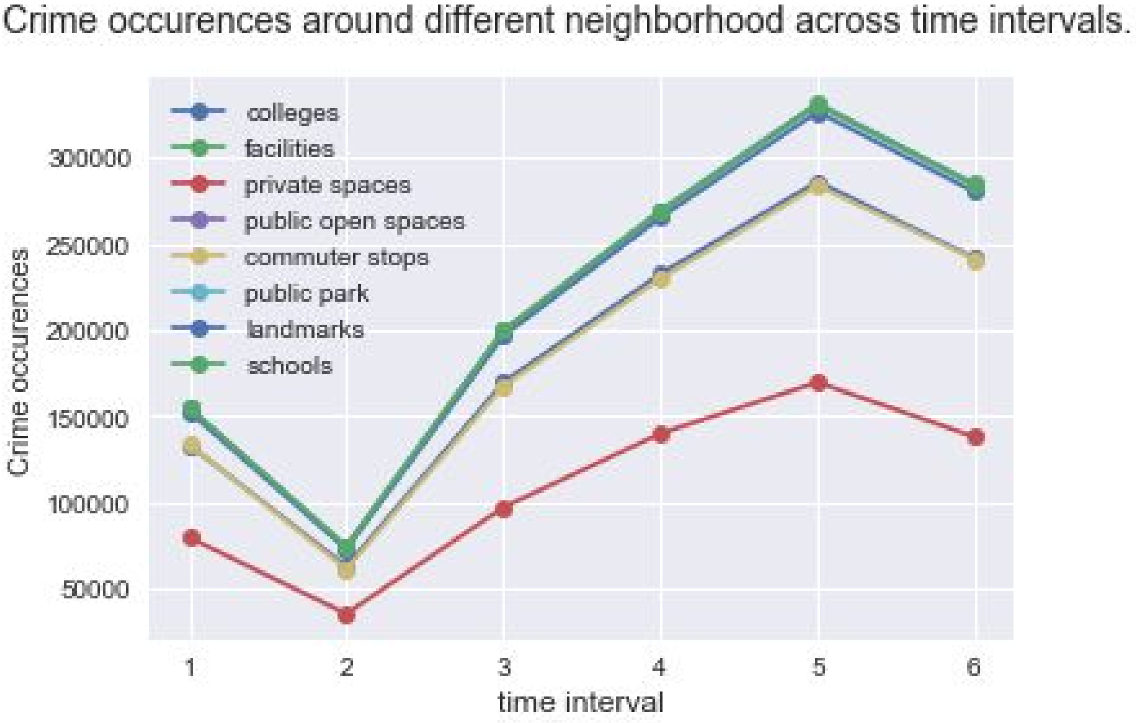
Data analysis visualizations have been provided below.



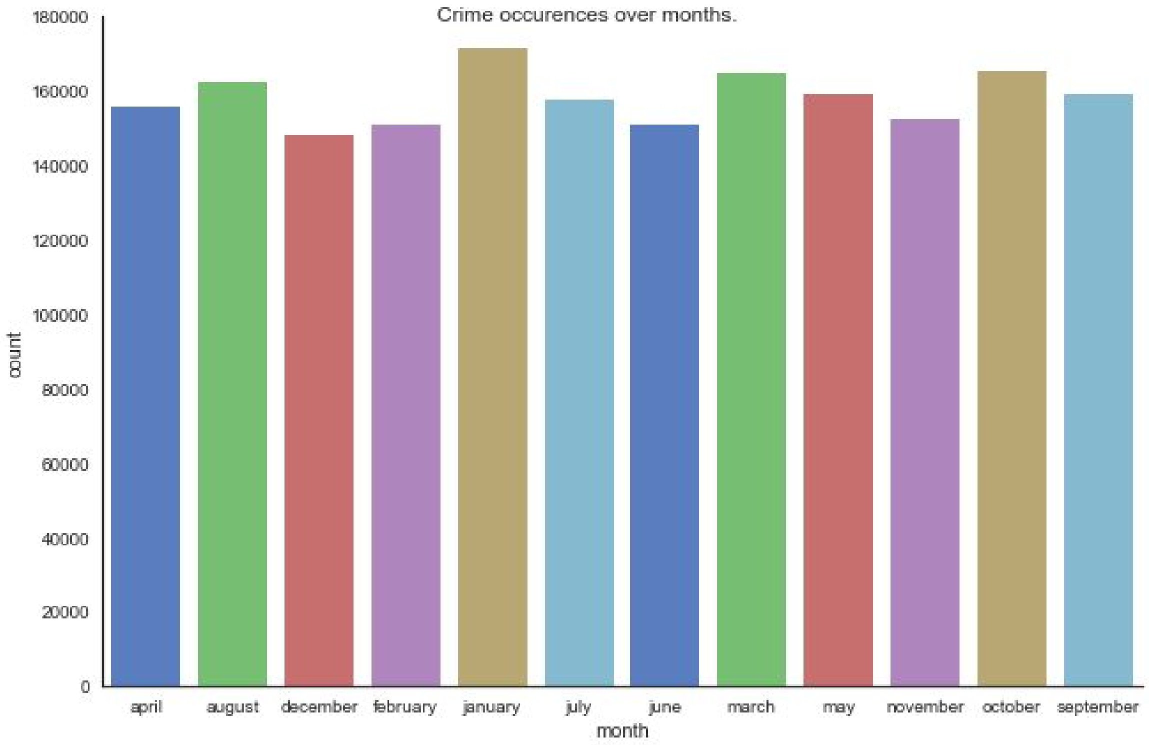


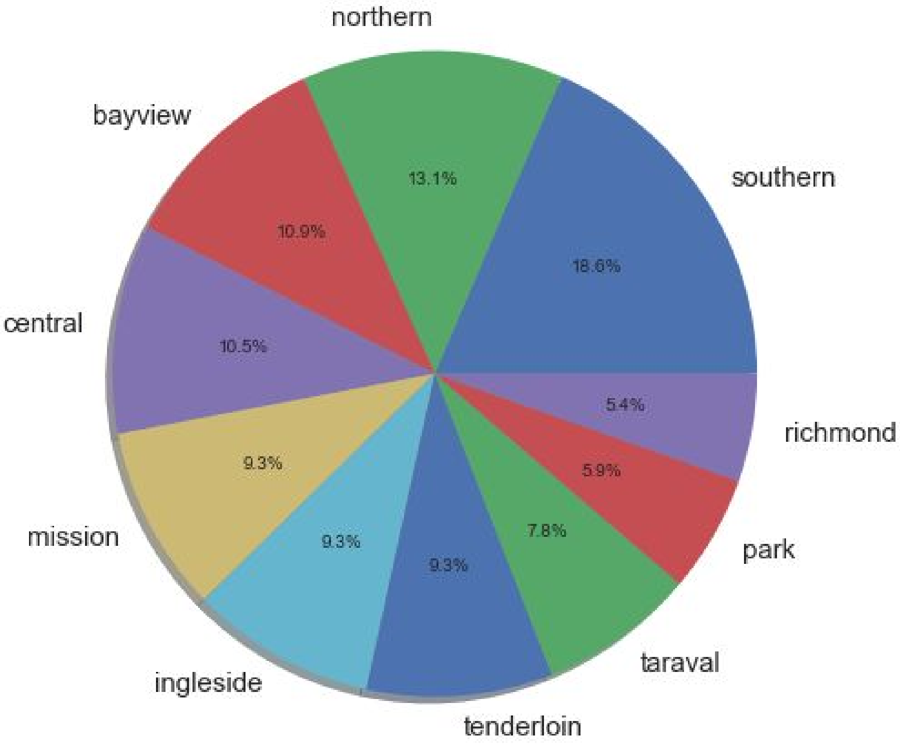


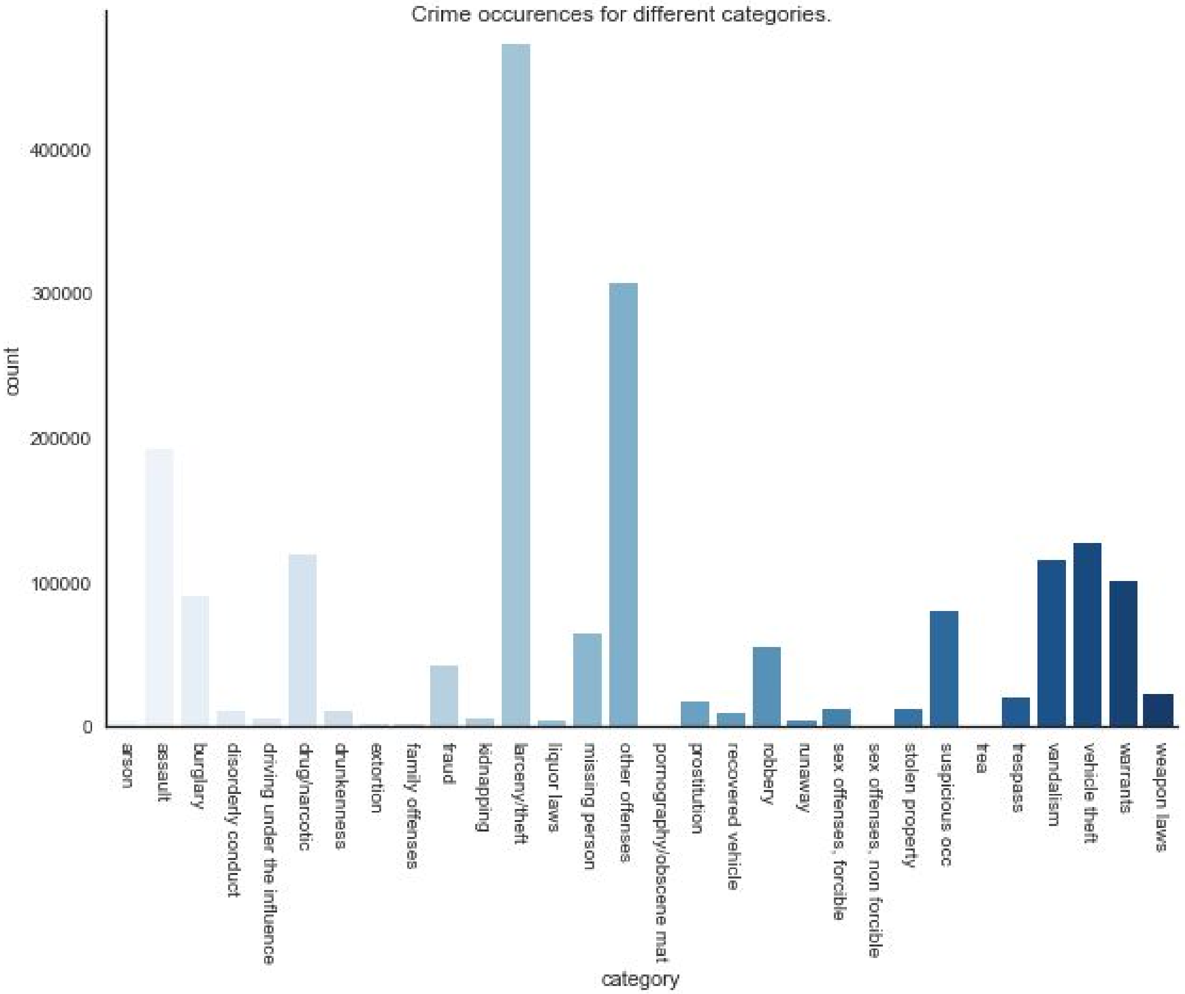
Crime occurences over years.



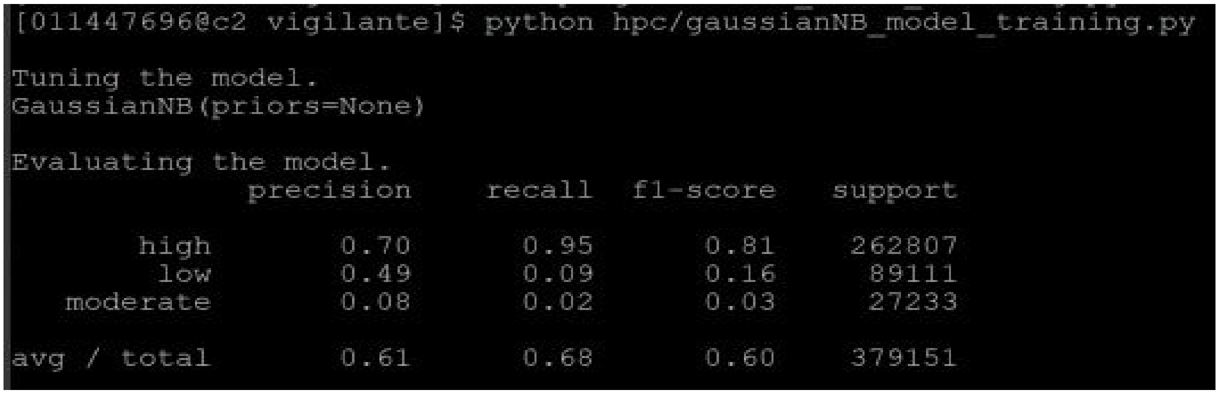
Crime occurences around different neighborhood across time intervals.


Crime occurences over months.
count
180000
160000
140000
120000
100000
80000
60000
40000
20000
0
april
august
december
february
january
july
june
march
may
november
october
september
month


northern
13.1%
bayview
10.9%
southern
18.6%
central
10.5%
5.4%
richmond
5.9%
9.3%
mission
9.3%
7.8%
park
9.3%
ingleside
taraval
tenderloin



Classification report for all the models have been provided below.



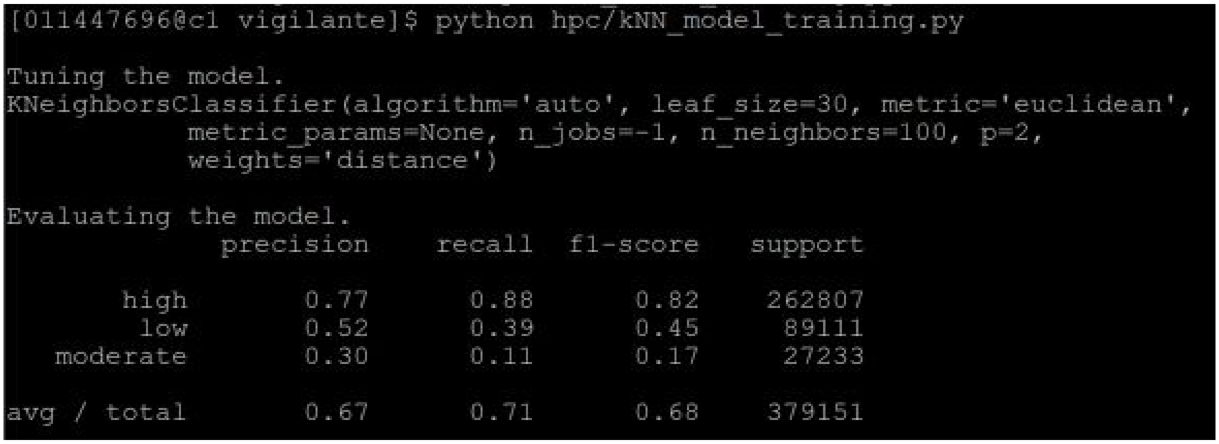
```
[0114476960c2 vigilante]$ python hpc/gaussianNB_model_training.py

Tuning the model.
GaussianNB(priors=None)

Evaluating the model.
             precision    recall  f1-score   support

       high       0.70      0.95      0.81    262807
        low       0.49      0.09      0.16     89111
   moderate       0.08      0.02      0.03     27233

avg / total       0.61      0.68      0.60    379151
```



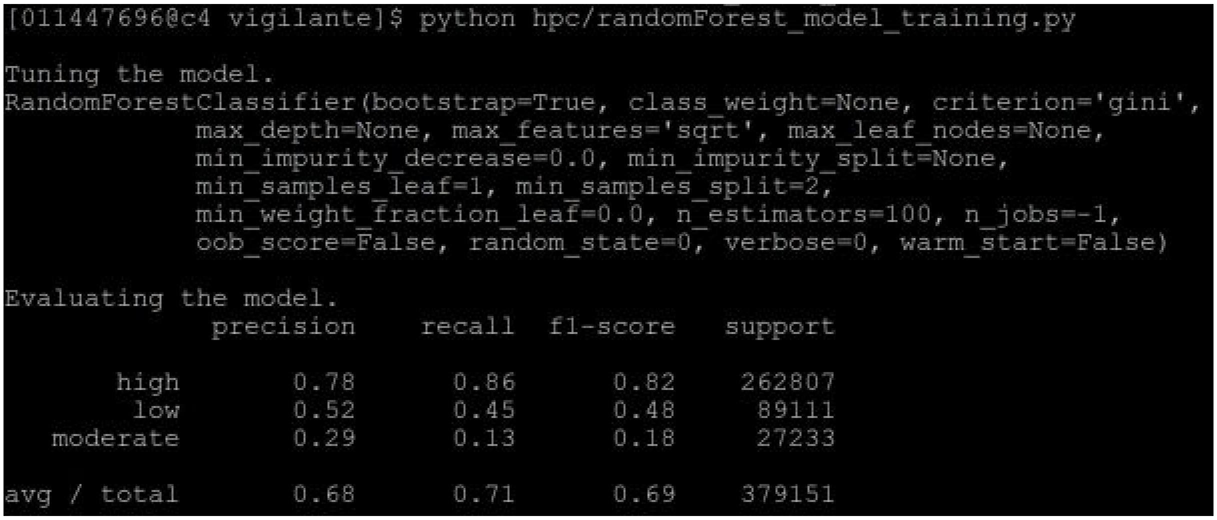
```
[0114447696@c1 vigilante]$ python hpc/kNN_model_training.py

Tuning the model.
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='euclidean',
           metric_params=None, n_jobs=-1, n_neighbors=100, p=2,
           weights='distance')

Evaluating the model.
             precision    recall  f1-score   support

       high       0.77      0.88      0.82    262807
        low       0.52      0.39      0.45     89111
   moderate       0.30      0.11      0.17     27233

avg / total       0.67      0.71      0.68    379151
```



```
[0114447696@c4 vigilante]$ python hpc/randomForest_model_training.py

Tuning the model.
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='sqrt', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=-1,
            oob_score=False, random_state=0, verbose=0, warm_start=False)

Evaluating the model.
             precision    recall  f1-score   support

       high       0.78      0.86      0.82    262807
        low       0.52      0.45      0.48     89111
   moderate       0.29      0.13      0.18     27233

avg / total       0.68      0.71      0.69    379151
```