San Francisco Crime Classification

INF - 552 Project Report

By:
Shilpa Gulati
Madhu Ramiah
Aravind Ram Nathan
Sheetal Mahesh

Table of Contents

1. Problem Description

From 1934 to 1963, San Francisco was infamous for housing some of the world's most notorious criminals on the inescapable island of Alcatraz. With rising wealth inequality, housing shortages, and a proliferation of expensive digital toys riding BART to work, there is no scarcity of crime in the city.

So the problem here is, given the crime data report for 12 years, we should be able to predict the category of crime based on time and location.

2. Resources and Tools Used

2.1 Dataset

This dataset is taken from Kaggle which contains incidents derived from SFPD Crime Incident Reporting system.The dataset was provided by SF OpenData.

- The data ranges from : **1/1/2003-5/13/2015**
- Train dataset size : **878050**
- Test dataset size : **884263**
- Crime Categories : **39**
- Number of features : **9**

The data fields are described below:

- **Dates** - timestamp of the crime incident
- **Category** - category of the crime incident (only in train.csv). This is the target variable you are going to predict.
- **Descript** - detailed description of the crime incident (only in train.csv)
- **DayOfWeek** - the day of the week
- **PdDistrict** - name of the Police Department District
- **Resolution** - how the crime incident was resolved (only in train.csv)
- **Address** - the approximate street address of the crime incident
- **X** - Longitude
- **Y** - Latitude

2.2 Tools

Scikit-learn: It is a machine learning python library. It features various classification , clustering and regression algorithms. It is designed to interoperate with the python numerical and scientific libraries NumPy and SciPy.

Pandas: Software library written in python for data manipulation and analysis. Data structure called data frame is used in this project for loading and generating csv files.It helps to read and write to csv files.

Matplotlib: It is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It helps in plotting the graphs.

Seaborn: Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.

3. Methodology

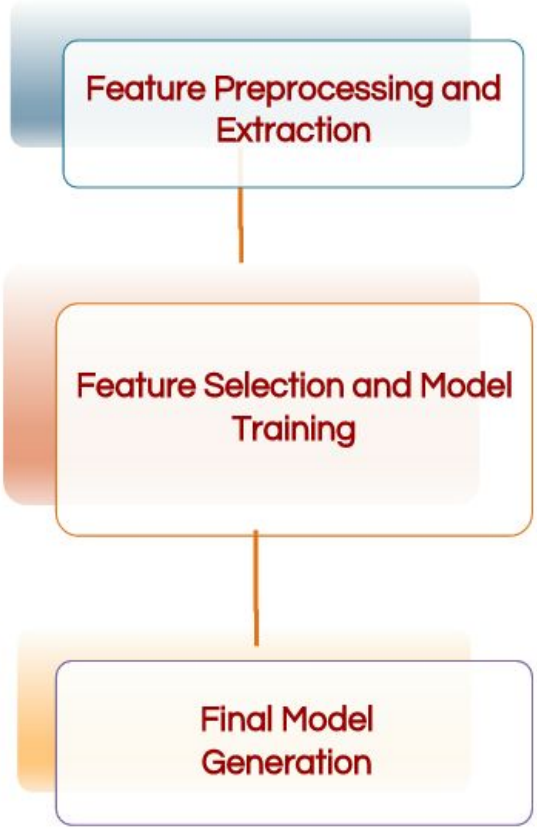

Figure 1

3.1 Feature Pre-Processing and Extraction

The following features were extracted:

- **Extraction of date and time:** The month, year, day of the week, hour and minute was extracted from the date.

- **Mapping of hour to Intervals:** We mapped the time into intervals: Morning , Noon, Evening and Night. The possibility for having a crime during the evening or night could be more and hence it seemed as a convincing division.
- **Mapping days as weekday or weekend:** Week has been divided into weekend and weekdays. As we thought that could also be good feature to study the pattern of certain kind of crime.
- **Dividing month into 3 slots:** Dates of month has been divided into three slots-first, second and third indicating first 10 , next 10 and last 10 days of month. This was done assuming crimes like robbery are more when people get salary or end of month leads to no money and increases robbery crimes.
- **Mapping months to season:** The months were mapped into Spring, Summer, Fall and Winter

3.2. Feature Selection

The two methods that were used to select the features were based on visualization and intuition.

3.2.1. Based on Visualization

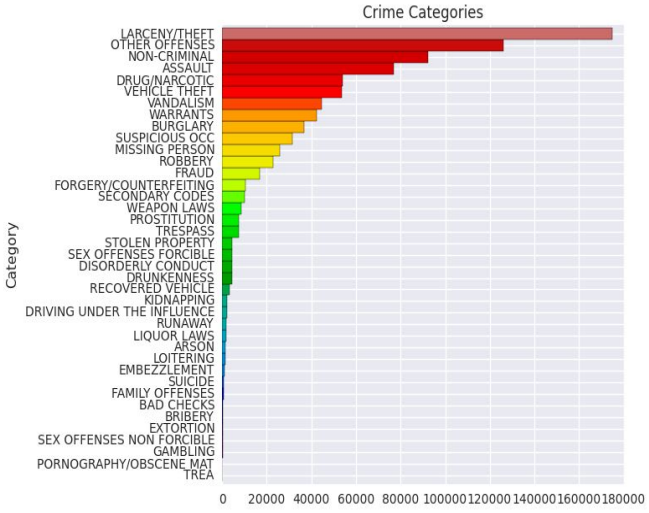

Figure 2

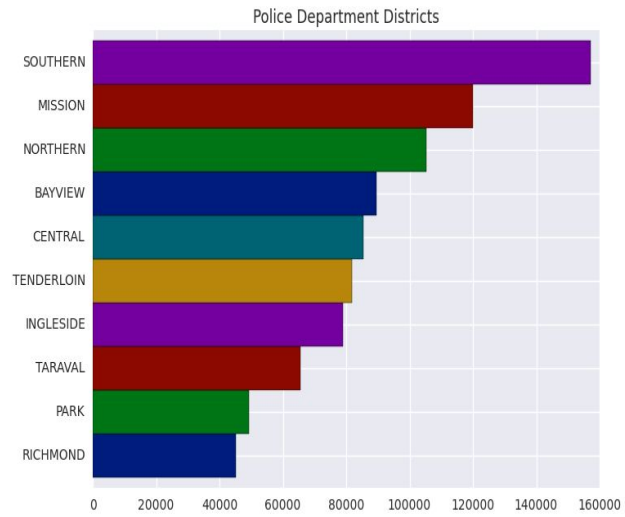

Figure 3

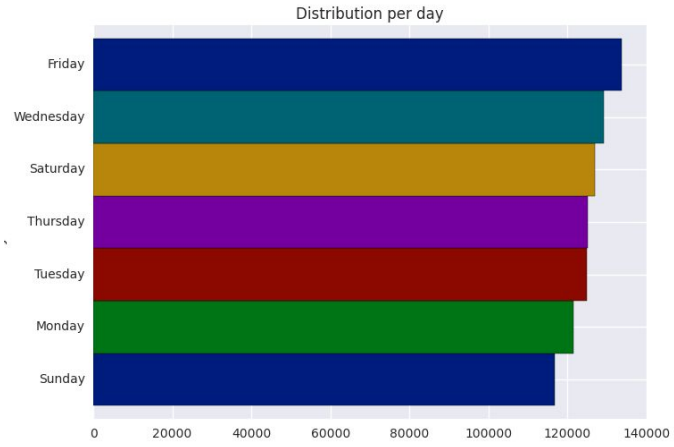

Figure 4

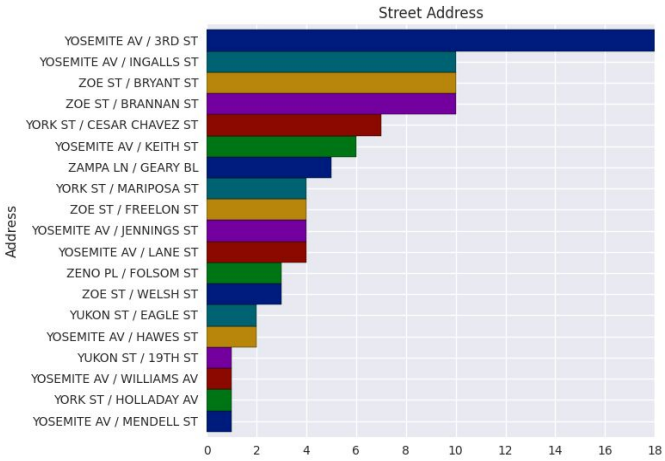

Figure 5

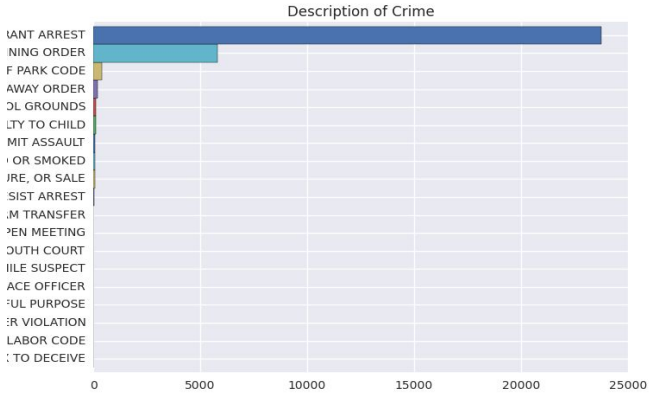

Figure 6

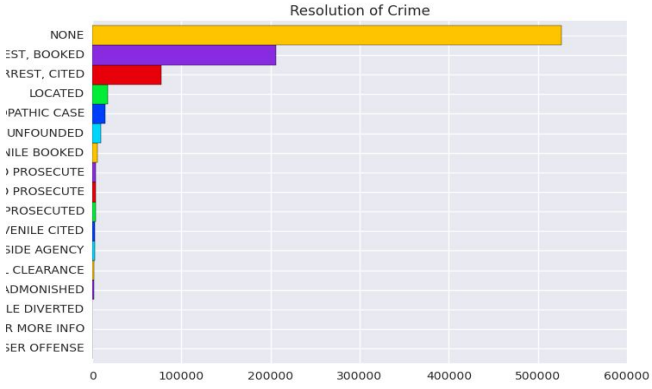

Figure 7

Figure 2 describes the various crime categories happening in San Francisco
Figure 3 shows the number of crimes happening in 10 different districts of San Francisco.
Figure 4 represents the number of crimes that occurred on each day of the week.
Figure 5 depicts the street address of where the crime took place in San Francisco.
Figure 6 is the description of the various crimes that happened in San Francisco.
Figure 7 describes how the crime was resolved.

After analysing the visualizations, we decided not to consider the description, street address of the place where the crime took place and the resolution of the crime since they did not show any significance in predicting the crime.

The district and day of the week features were chosen based on this method of selection.

District - 10 districts
Days - 7 days

3.2.2. Based on Intuition

After analysing the results using extracted features, we came up with following set of features which gave good predictions.

Hour- 0 to 23
Minute- 0 to 59
Season- Spring, Summer, Fall, Winter
Time of the month- 1st-10th, 11th-20th, 21st- 30th/31st

We predicted the category based on various subset of features:

- District
- District+Days
- District+Days+Hour
- District+Days+Minute
- District+Days+Minute+Hour
- District+Days+Minute+Hour+Season
- District+Days+Minute+Hour+Season+Time

3.3 Implementation

In this project we have used four different algorithms to predict the categories of crime in San Francisco and we have developed our own implementation of the algorithms using scikit-learn and pandas libraries.

3.3.1 Decision Tree

Decision tree is known for giving perfect splits. Being a classification algorithm, it was one of the choices to pick for this problem. As expected we got a good training accuracy(84%) but the validation accuracy was still low like 21 % . That's how we came to know the decision tree is resulting into overfitting. So with large data sets and and too many impacting features, it does not seems to give good split and hence no good results.

3.3.2 AdaBoost

This one ensemble method we have picked to run on our problem. It was a combination of decision tree as a weak classifier which was boosted based on weights to focus on more complex problems. This Classifier has tend to give acceptable rate to choose it as one of the classifier to give class labels.

3.3.3 Naive Bayes

Naive Bayes was best choice for such a problem because for some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers has worked quite well in many complex real-world situations. The independent feature model was derived using Naive Bayes probability model.probability Then the naive Bayes classifier combines this model with a decision rule to give the posterior probabilities for each class.

3.3.4 Logistic Regression

Based on the recent research papers on crime classification it was seen that Logistic regression has given good results for such classification problems Few features of logistic regression also has made it a good choice as a classifier:

- It is intrinsically simple, it has low variance and so is less prone to overfitting.
- Logistic regression can optimize the multi-class (multinomial) problems directly.

Since our dataset also has multiple classes it was suitable to use this algorithm.

4. Visualizations

4.1 Density of different crimes happening in various Police Department Districts

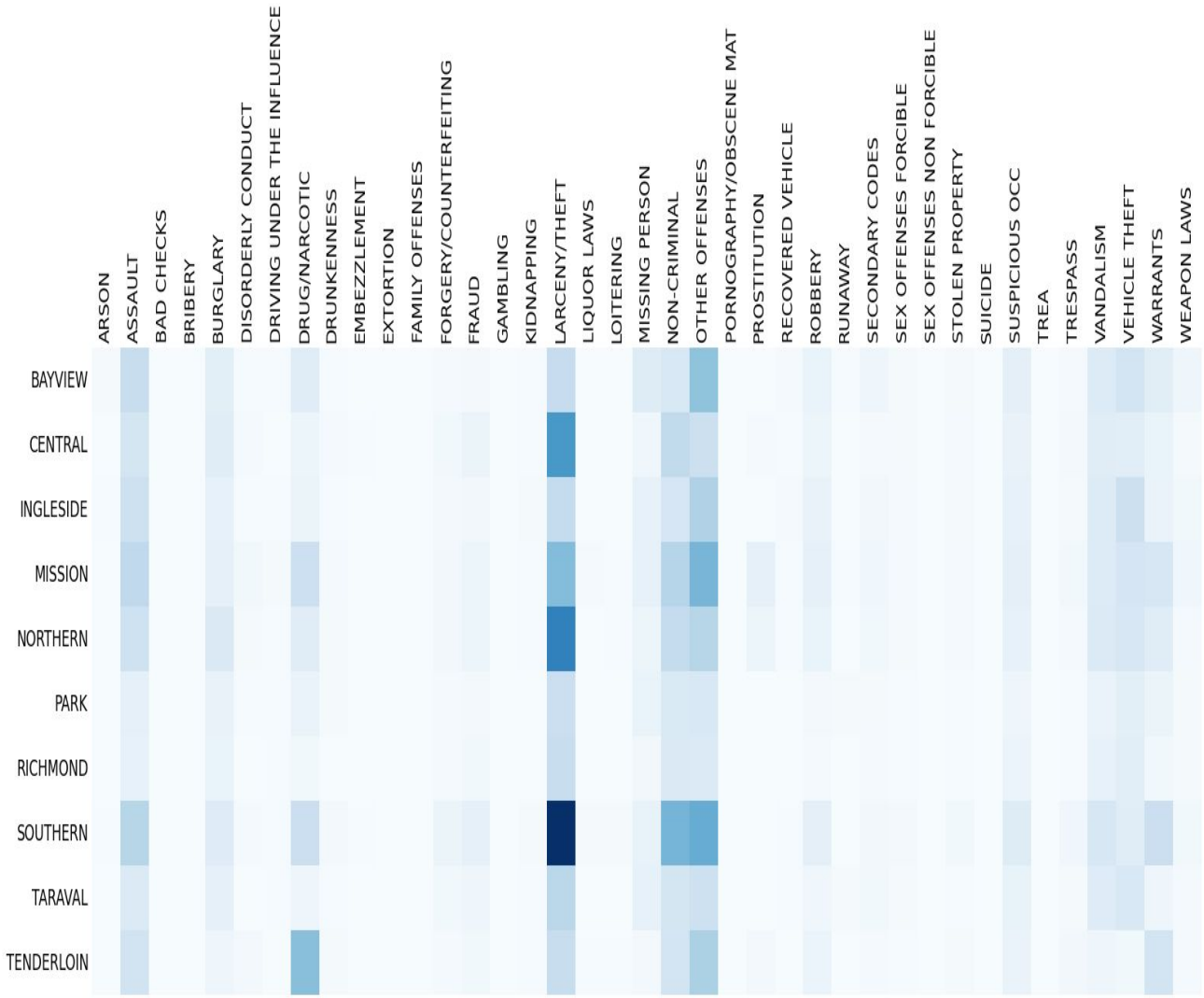

Figure 8

Figure 8 is a heat map which is plotted based on the various crimes that occur at each district in San Francisco. The more darker the region is in the heatmap implies more number of crimes have occurred in the corresponding district. As seen in the graph, the crime **Larceny/Theft** for the **Southern** district is dark, which means that the crime rate is high. The crime larceny/theft for the northern district is the next darkest region, which means the crime rate is high.

4.2 Density of different crimes happening in each month

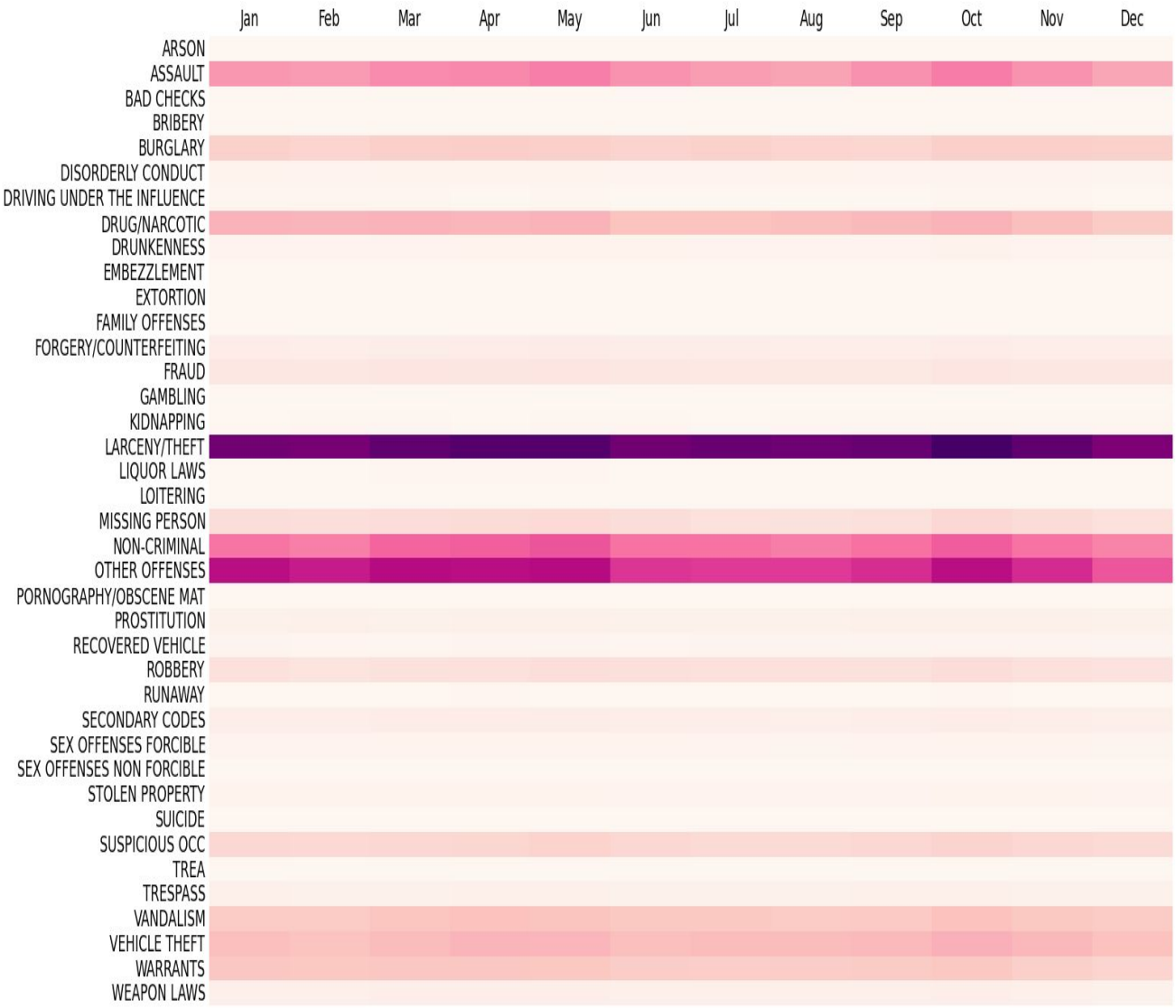

Figure 9

Figure 9 depicts the various crimes occurring during each month of the year. The **Larceny/Theft** has the darkest values in the graph throughout the year, which means that it has the highest crime rate and it also implies that theft is the most common crime that happens throughout the year in San Francisco. The next highest crime is the **Other Offenses** which is the next darkest color in the graph.

4.3 Density of Crimes in different Police Department Districts from 2003 to 2015

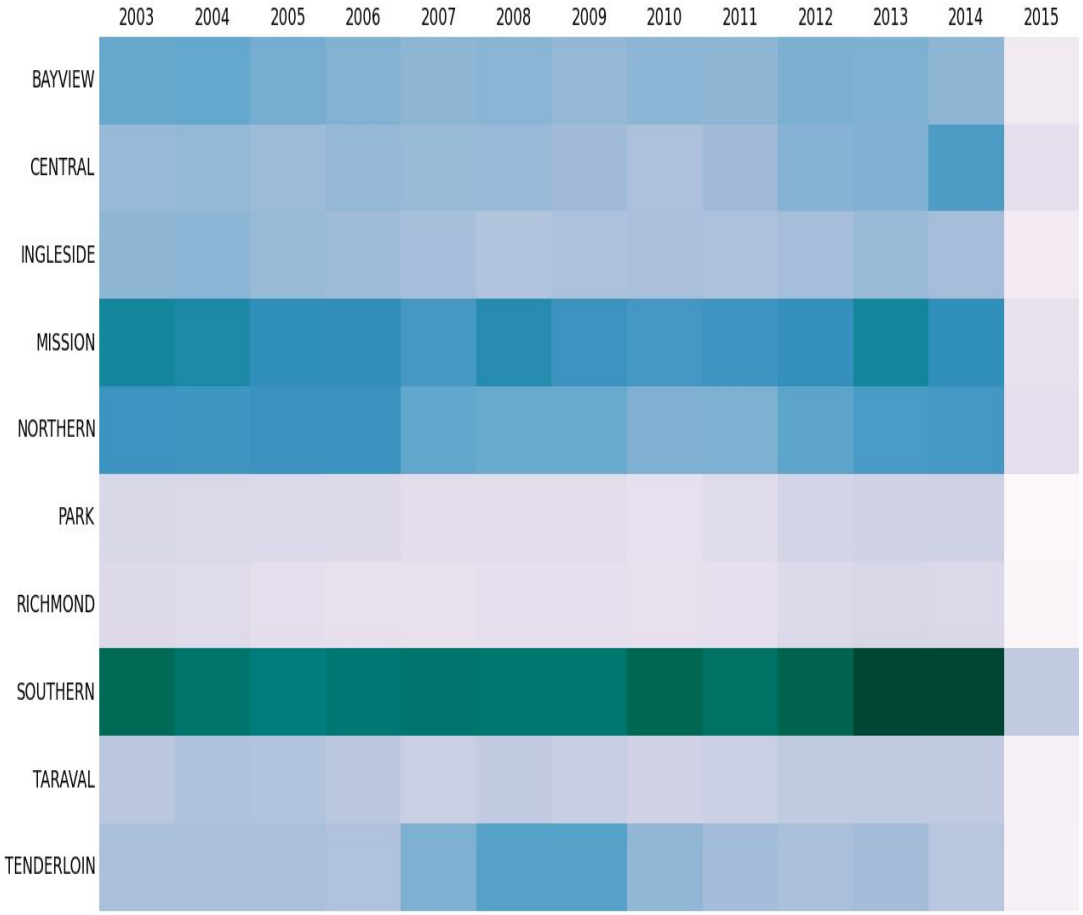

Figure 10

Figure 10 is a heat map plotted for the crimes occurring in each district for the different years ranging from 2003 to 2015. The highest crime was seen in **2013 and 2014** in the **Southern** district. It can be said that throughout the years, Southern district has had the highest crime rate.

4.4 Density of Crimes happening in different Police Department Districts in each month

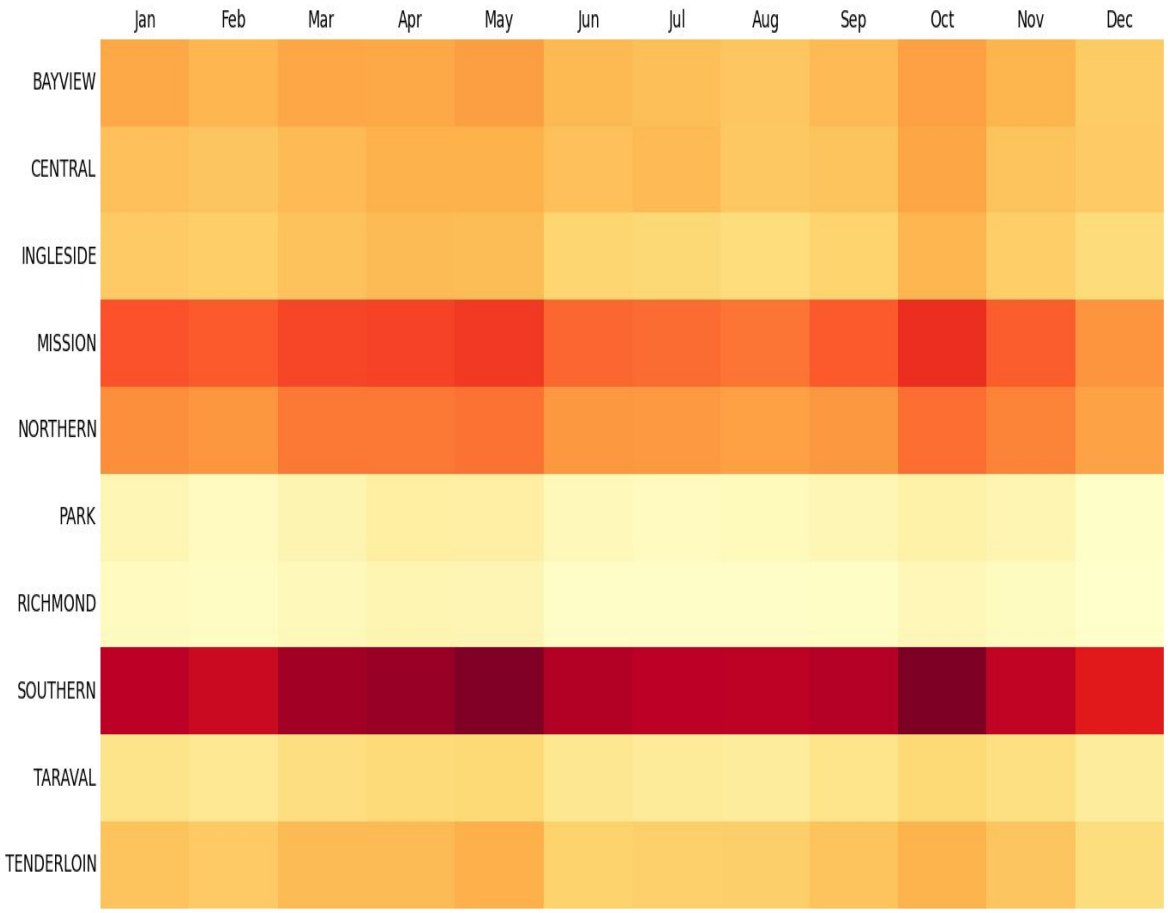

Figure 11

Figure 11 is a heat map plotted for the crimes occurring in each district during each month of the year. The red color regions represent the region with most number of crimes happening in a district followed by orange and yellow color regions. The highest crime was seen in the months of **May and October** in the **Southern** district since the red color region is more darker in those months. It can be said that throughout the years, Southern district has had the highest crime rate and the other districts like Northern and Mission have the next highest crime rates.

4.5 Theft Density Plot

Figure 12

The heat map in Figure 12 depicts San Francisco and the regions that have the highest crime rate. As shown in the figure the southern region of San Francisco is affected the most.

5. Evaluation

5.1 Performance Metric

The metric that we use for evaluation is Log Loss. It is the cross entropy between the distribution of the true labels and the predictions. The metric used in kaggle for evaluation is log loss.

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\log(p_{i,j})$$

Where N is the number of observations, M is the number of class labels, log is the natural logarithm, yi,j is 1 if observation i is in class j and 0 otherwise, and pi,j is the predicted probability that observation i is in class j.

5.2 Results

The table describes the Log loss values for each algorithm for different training data sizes.

Training Size	Decision Tree Log Loss Values	Adaboost Log Loss Values	Naive Bayes Log Loss Values	Logistic Regression Log Loss Values
0.1	21.4458122203	5.6705170776	2.50766161231	2.54732653728
0.2	18.4424939658	5.15462678977	2.49755018581	2.51476998675
0.3	16.6030055107	5.01294382118	2.49638014317	2.50320609131
0.4	15.3057479882	4.67375972256	2.49556548392	2.49795796978
0.5	14.3542091956	4.57544699794	2.49231120244	2.49234510059
0.6	13.4983202545	4.50266710522	2.48962181966	2.48860305124
0.7	12.8190914845	4.41211901222	2.4915695719	2.48734418229

0.8	12.230428915	4.30239172936	2.4903840861	2.48484732522
0.9	12.4566787889	4.20137725234	2.48696325938	2.48045027258

Figure 13 represents the log loss of different algorithms. Naive Bayes and Logistic Regression have very similar values. It is because of this reason, Naive bayes is seen overlapping with the values of Logistic Regression, hence it shows only Logistic Regression values in the graph. Any algorithm which has a log loss value closer to zero can be considered as a good classifier. It can be seen from Figure 13 that Naive Bayes and Logistic Regression have log loss values closer to 0 and hence can be considered as very good classifiers for the given crime dataset. However, decision tree classifier seems to produce a higher log loss value compared to all other algorithms. Hence, we cannot conclude that it is not a suitable classifier for classifying crimes of many categories. This observation is also supported in [2].

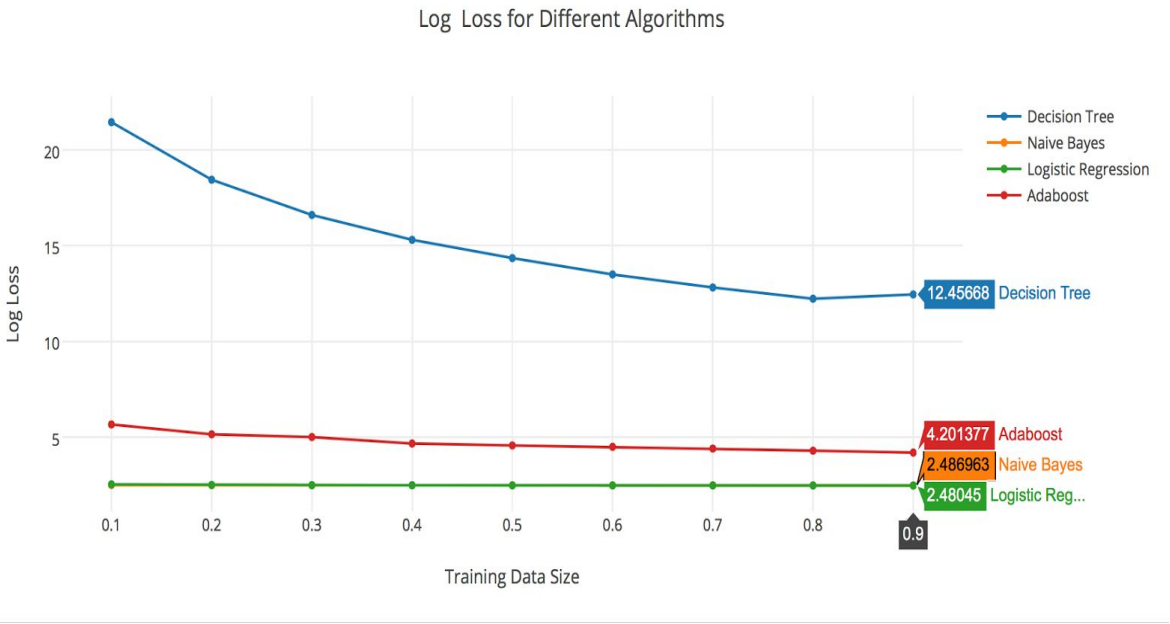

Figure 13

5.3 Runtime comparison

Figure 14 depicts the runtime of each algorithm in seconds.

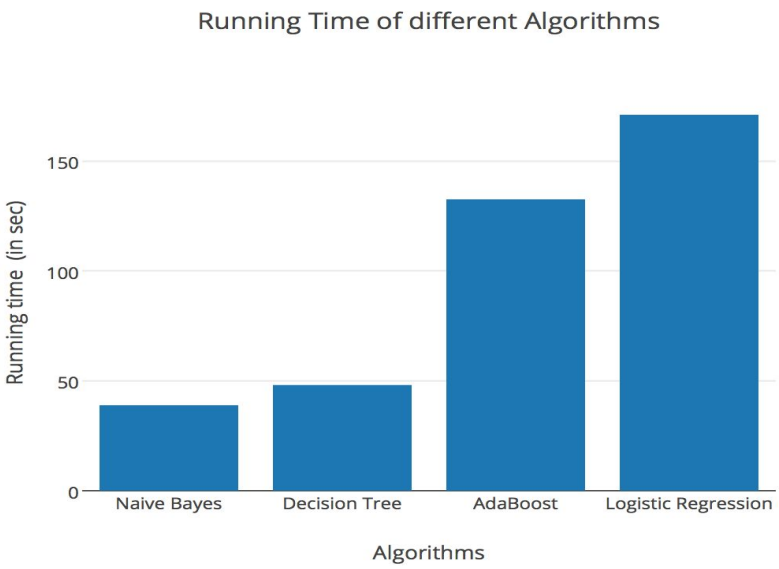

Figure 14

Algorithm	Run time (in sec)
Decision tree	**42.108**
Adaboost	**132.60**
Naïve Bayes	**38.92**
Logistic Regression	**171.08**

The above table indicates the running time of each algorithm in seconds computed for the same training data size.

5.4 Prediction of test data

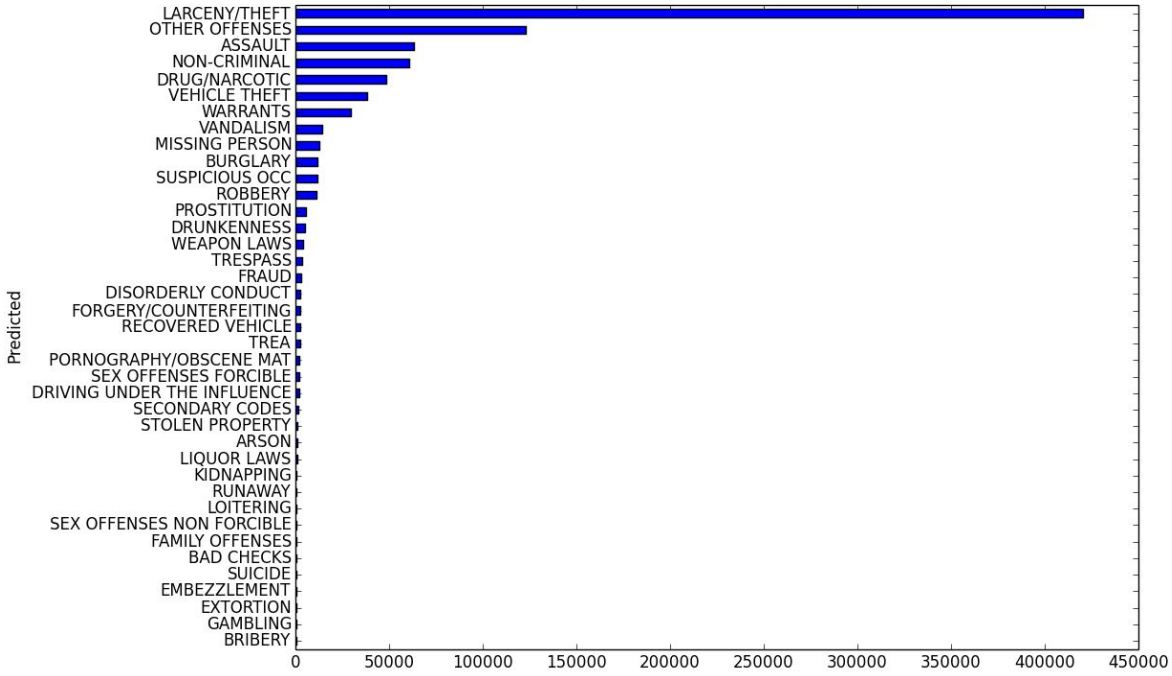

Figure 15

Figure 15 describes the prediction of crimes on the test dataset. It can be seen that out of the **884263** entries in the given test set, around **420000** crimes have been classified as **Larceny/Theft** followed by other crimes like **Assault, Non-Criminal, Drug/Narcotic** and so on. The crime was predicted based on the probability values computed using the 4 algorithms and the crime corresponding to the highest probability was the predicted category for each entry in the test dataset.

5.5 Rank in Kaggle

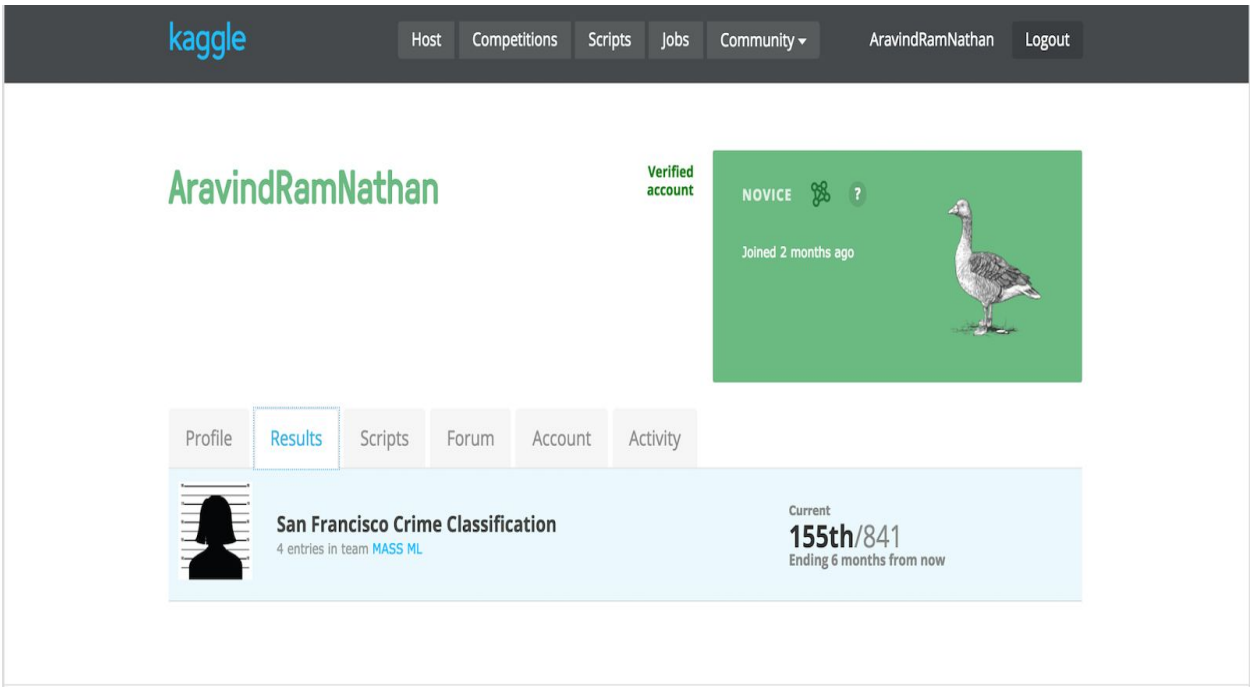

Figure 16

Figure 16 shows the current rank of our team in Kaggle. The current score of our team is 2.4854. The top score in this competition is currently 2.15453

6. Future Work

The next goal is to visualize data from different perspectives and come up with better set of features which improves the performance of our crime prediction model so that we get a better log loss value and our score in the competition goes up thereby increasing our rank in the competition.

7. Acknowledgments

This project required more dedicated time for the feature extraction and feature selection phase. Since the dataset was large, extracting and selecting only the desired features was important to build a proper model which would give good results and also would take less time to run. So, we decided to divide the work equally for the feature extraction phase. For the feature selection phase, out of the 6 visualizations (which were plotted by writing a python script using Pyplot library), 3 were plotted by Aravind and 3 were plotted by Sheetal. Shilpa and Madhu wrote a python script to extract additional features like Season, Month, Hour and Minute which at the end gave us good results. For the implementation phase, each of us worked on one algorithm each using the scikit-learn and pandas python libraries. Sheetal wrote the script for building the model using decision trees. Shilpa built the model using Adaboost algorithm. Madhu built

the model using Logistic Regression. While, Aravind built the model using Naive Bayes algorithm. The heatmap visualizations and density plots were plotted using Pyplot by Aravind. The evaluation of the model using the log loss metric was done by Madhu and Shilpa and comparing the performance based on the running time of each algorithm was done by Sheetal. The python script used to the compute the probabilities and predict the crime categories on the test data was written by Madhu and Aravind. The report work was split equally among us, with each of us explaining their respective tasks. Apart from these major tasks in the report, section 6 was written by Shilpa and sections 7,8 were written by Aravind.

8. References

[1] Nasridinov, A., & Park, Y. H. (2014). A Study on Performance Evaluation of Machine Learning Algorithms for Crime Dataset. Advanced Science and Technology Letters, Vol.66 (Networking and Communication 2014), pp.90-92

[2] Shojaee, S., Mustapha, A., Sidi, F., & Jabar, M. A. (2013). A Study on Classification Learning Algorithms to Predict Crime Status. International Journal of Digital Content Technology and its Applications, 7(9), 361-369.

[3] McClendon, L., & Meghanathan, N. (2015). Using Machine Learning Algorithms To Analyze Crime Data. Machine Learning and Applications: An International Journal, Vol.2, No.1.

[4] Saeed, U., Sarim, M., Usmani, A., Mukhtar, A., Shaikh, A. B., & Raffat, S. K. (2015). Application of Machine learning Algorithms in Crime Classification and Classification Rule Mining. Research Journal of Recent Sciences, Vol. 4(3), 106-114.