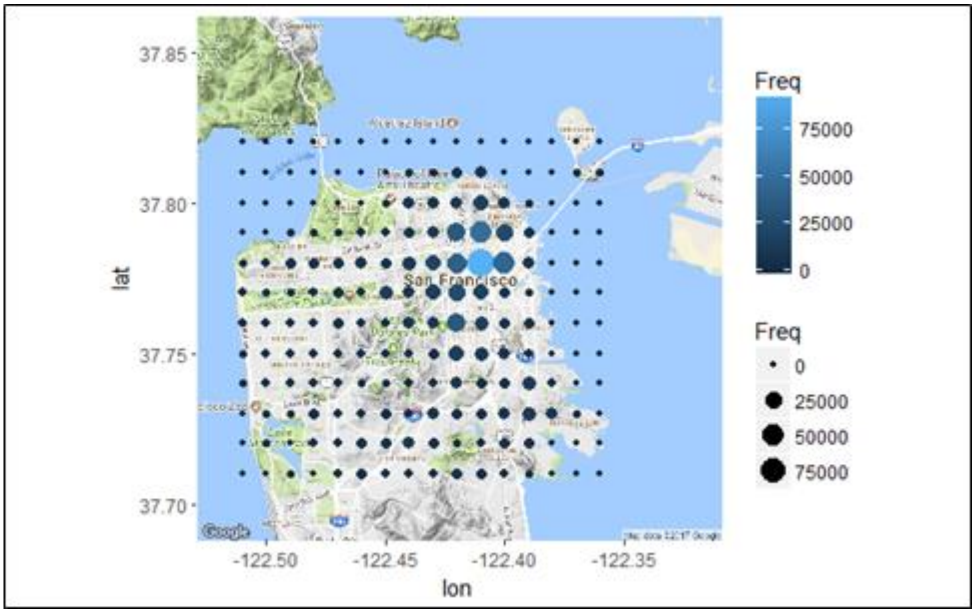
San Francisco crime classification: Descriptive, Predictive, and Prescriptive analysis



San Francisco Area under Radar

1. Abstract

Predicting the crime and the crime rate is one of the essential factor in improving the efficiency of police department and reducing threat for the public. We are working on San Francisco Crime classification Data from Kaggle which was collected from SF Police Department reporting system. We analyzed the data from 2003 to 2015 with more than 800,000 observations in training data set and around 880,000 observations in testing dataset. Using data modeling approaches such as Support Vector Machine SVM, Random Forest, XG Boost we created models that can be used to classify category of crime given the location and time.

2. Introduction

San Francisco first boomed in 1849 during the California Gold Rush. The city then expanded both in terms of land area and population. As a result, the crime rate and civil problems also proliferated. However, San Francisco of today is different than what it was at it’s beginning. Now it is well known for the Silicon Valley and the Tech giants than that for its criminal history.

With the increase in crime rate it is very difficult to predict the crime and prevent it from happening. Though with the help of data mining and other tools the prediction of crime can be done. This doesn’t mean that the crime will be completely controlled but to some extent the SFPD can provide helpful information to prevent crime. The San Francisco Crime classification data with 800,000 observations has following features:

· Dates—timestamp of the crime incident

· Category—category of the crime incident

· Descript—detailed description of the crime incident

· Day of week—the day if the week

· PdDestrict—name of Police Department District

· Resolution—how the crime incident was resolved

· Address—Approximate street address of the crime incident

· X—longitude

· Y—latitude

The X and Y essentially gives the location parameter.

Date is clubbed together as a date, time and day of week. This is further split for further use in following sections.

3. Structure of DATA:

Data consists of 800k observation of 9 variables.

There are total of 39 categories of the crime. The Data seems widely distributed in terms of frequency of the occurrences. As the top 4 categories of the data are composed of almost 53% of the data and the rest of the 47% is comprised of the remaining categories. The data has the highest frequency of 174900 for larceny and just 6 for TREA (trespassing and loitering in industrial area). We have added following new variables to the data set for better understanding by using strip time function from caret package.

Year, month, day, hours, weekday/ weekend and day of month.

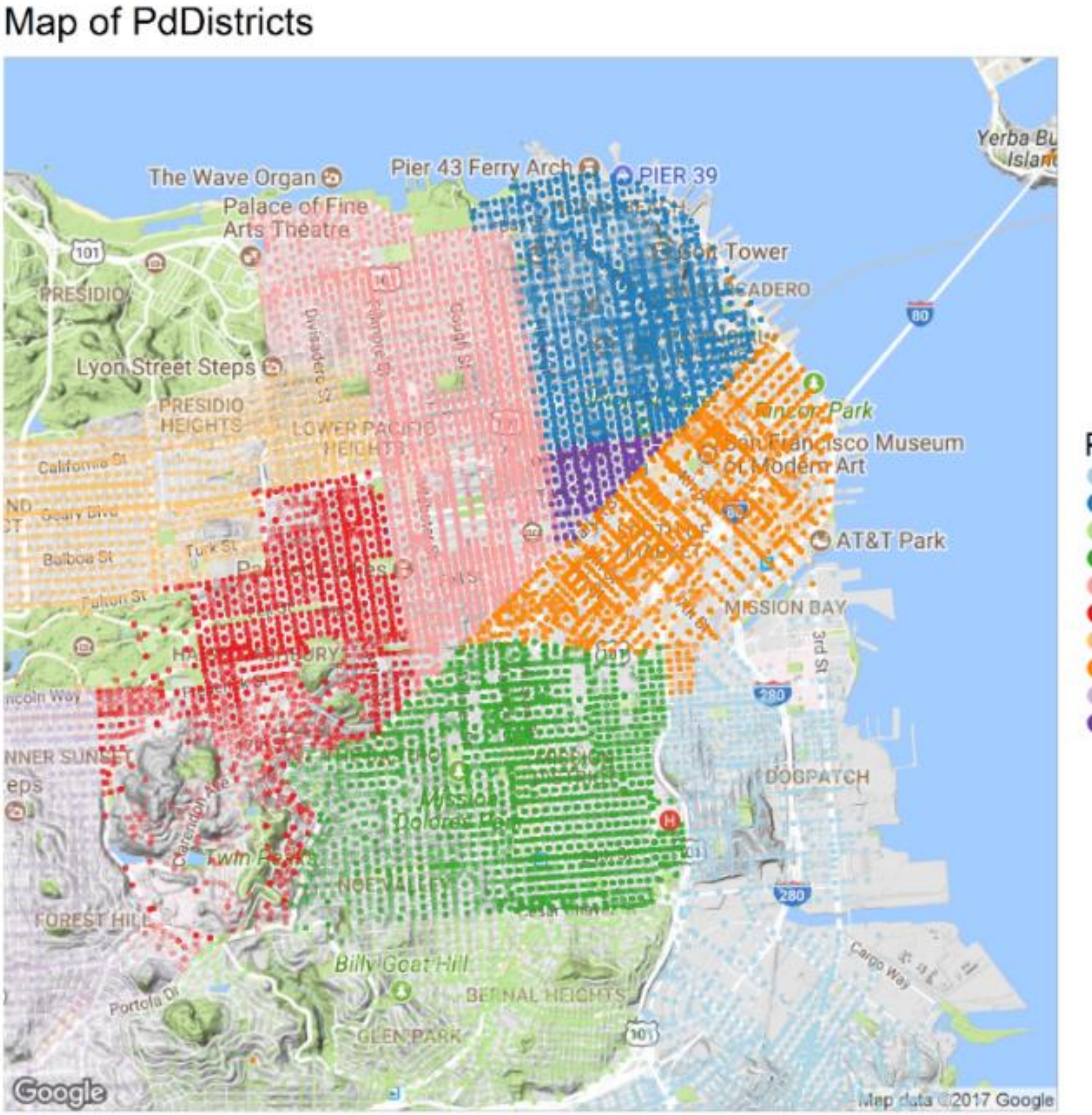
Further the address type is also split to understand if the crime incident was an intersection of two roads.

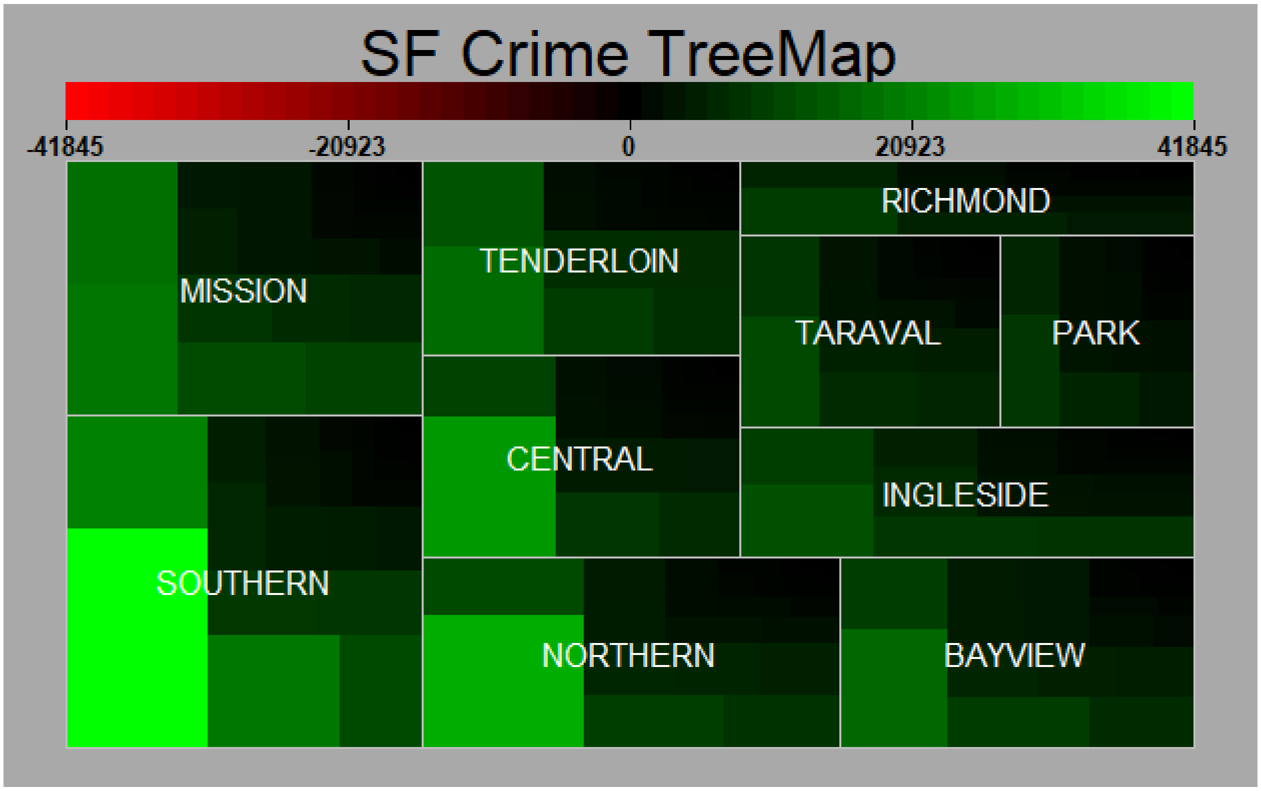
1. Exploratory Analysis:

San Francisco is one of the busiest cities around the world and it is incredibly difficult to know what is going on. Given the strength of police departments we cannot just deploy the police equally around the city for policing. Some of the parts have a high crime rate while others are not as high.

To clarify the crime incidents, we super imposed the crime incidents' longitude and latitude on the San Francisco map.

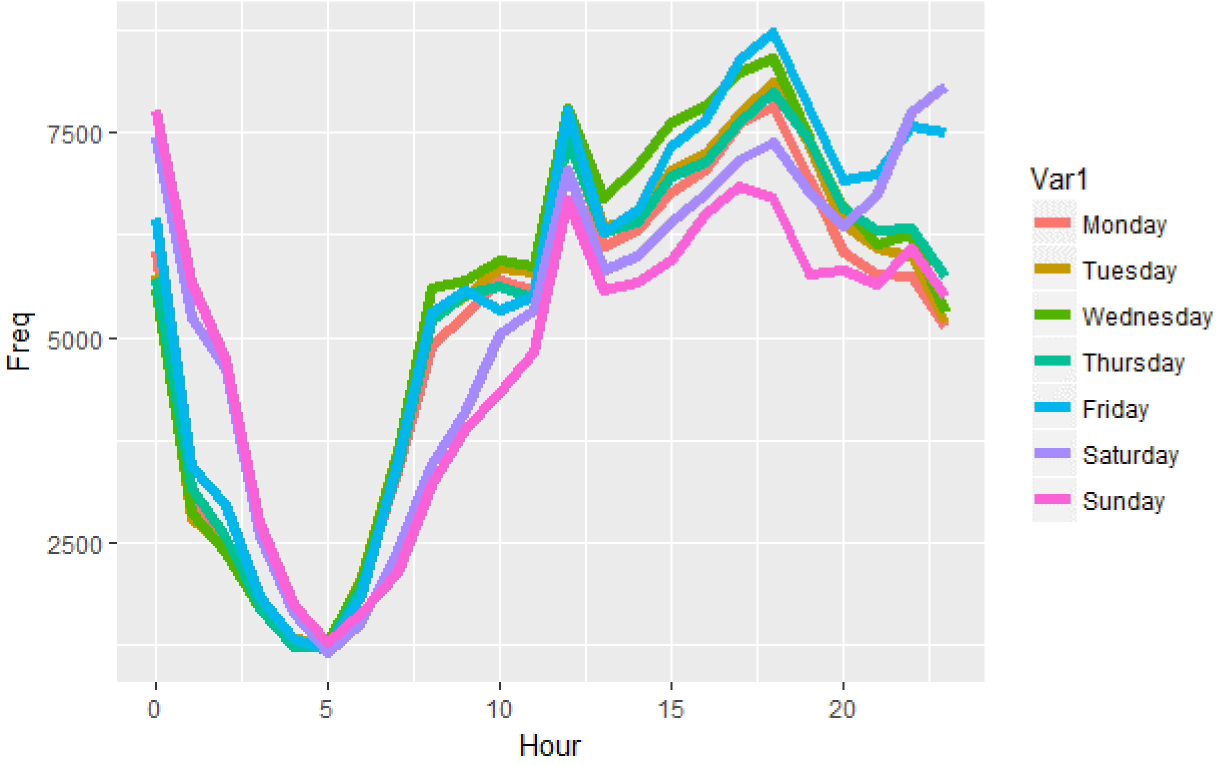
4.1 Distribution of crime incidents:



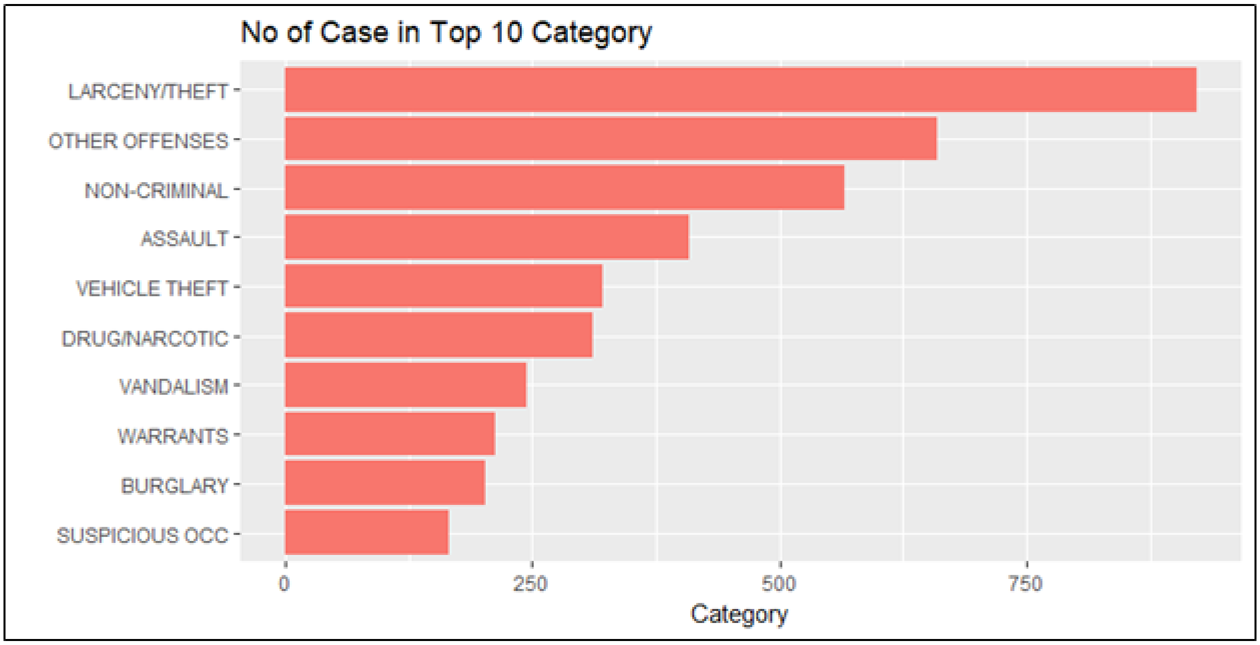


Area wise crime distribution

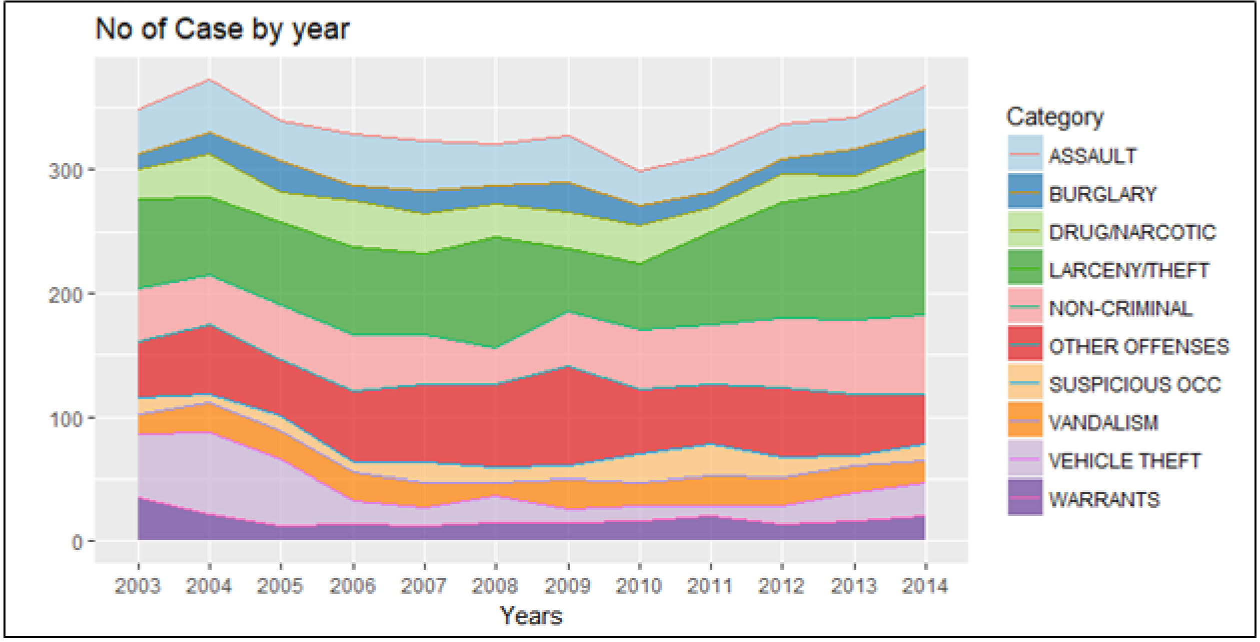
Distribution of crime events with timestamp



The graph tells us that the crime rate is significantly higher from 15:00 hours to 20:00 hours on all days and decreases as the time passes but the crime rate on weekends (Friday night and Saturday night) falls shortly and increases again.



Prevailing Crimes



Prevailing Crimes with respect to Years

4.3 Influential parameters:

Since there are 16 variables influencing the categories of the crime committed we need to narrow down our search and check if all the parameters are influential. Using Variable importance function from Caret package in R we narrow down this search.

The influential variables are Longitude, Latitude, Police department district, year of crime, week number in the entire year, day of week and month. The graph indicates that the X and Y combined have an effect of over 30% influence on the crime category.

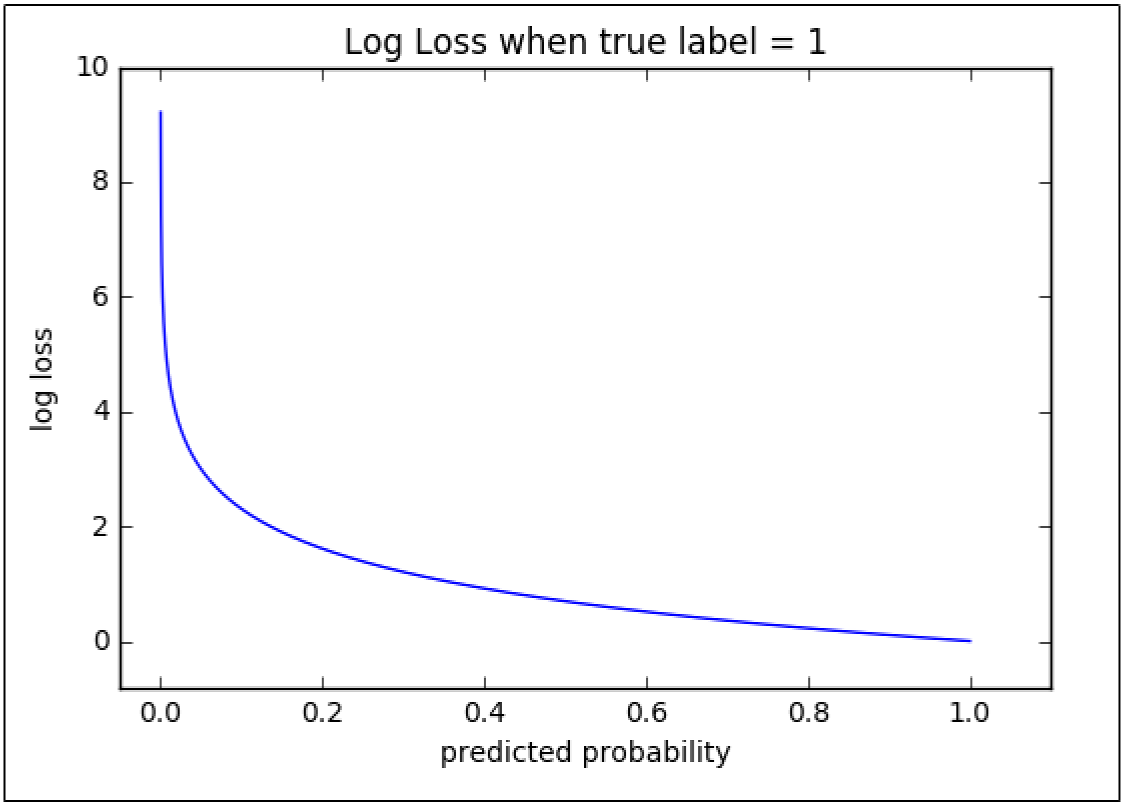
5. Goal

The goal of the project is to classify the given crime given the time and location of the incident.

In this process, we exploded the date column which had a lot of data clubbed together. We included following parameters in our model which were relevant to classification, PD district, hours, years, X, Y.

Ultimately, we will be calculating the log loss for the classification. Log loss calculates the wrong classification of the model and penalized the model for each wrong classification. This eventually indicates the entropy of the entire model. The higher the miss classification of the categories the higher is the entropy in the model. We can say that there is excessive noise in the system which leads to wrong classification.

The Leaderboard in Kaggle for this competition had a logloss value of 1.95936. We would try to reach as close to the value as possible.



Log Loss with respect to Accuracy

LogLossBinary (1, c(0.5))

0.69315

LogLossBinary (1, **c**(0.9))

0.10536

LogLossBinary (1, **c**(0.1))

2.3026

5. Models

5.1 Decision Tree:

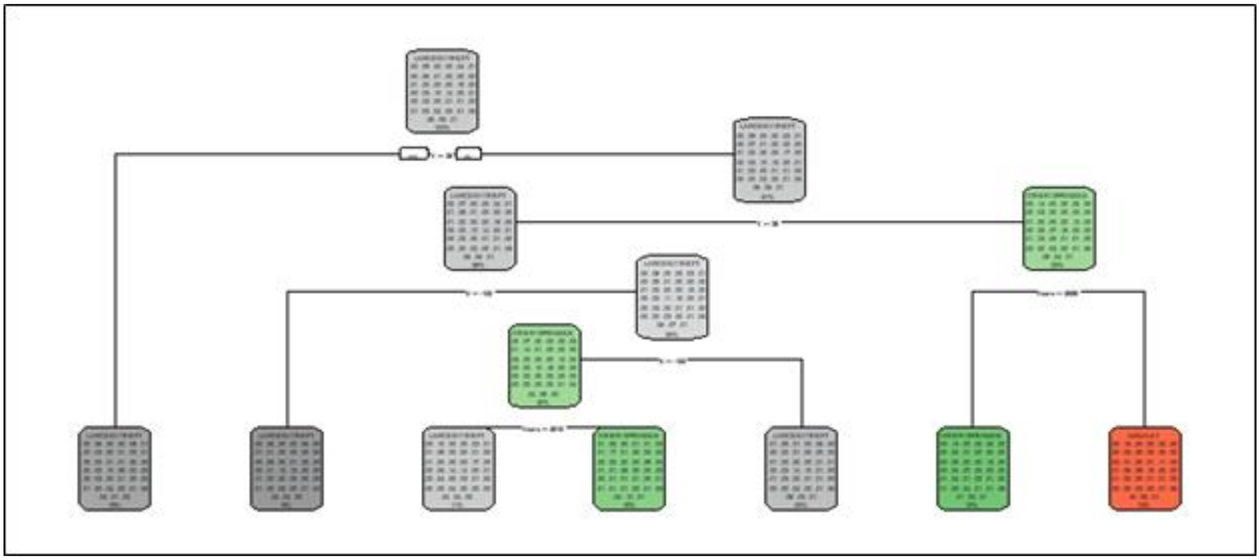
The decision tree algorithm works by splitting the dataset recursively, which means that the subsets that arise from a split are further split until a predetermined termination criterion is reached. At each step, the split is made based on the independent variable that results in the *largest possible reduction in heterogeneity of the dependent variable*.

Here we can see as described above the Larcency/Theft which occupies the 30 percent of our total dataset has been predicted by the model the most. The three classes out of total 39 classes predicted by decision tree comprises of 68 percent of the dataset.

To develop this model, we chose a split for a given subset that minimizes entropy in the subset's, as the higher entropy got us higher accuracy but the misclassification error also increased. Thus, we computed the weighted average over all sets resulting from the split

Recursive partitioning is implemented in “rpart” package and plotted the conditional tree graph with respect to category.

We used the observations in the subset, apply statistical test of independence between each feature and the labels. The model helped us to understand the best feature for predicting the labels on our subset.



5.2 Random Forest:

Random Forests is a very popular ensembling learning method which builds many classifiers on the training data and combines all their outputs to make the best predictions on the test data. Thus, the Random Forests algorithm is a variance minimizing algorithm that uses randomness when making split decision to help avoid overfitting on the training data.

We used random forest to rank the features based on their importance to predict the labels.

In our study, we came across that random forest do not work well with negative values thus we took the absolute of the longitude(X), it didn’t make any difference on the dataset but slightly improved the performance of the model.

Error estimate came up to 74.77% the model did really bad on it. But the evaluation metrics used by the Kaggle is the log loss which is a probabilistic prediction scale we scored 2.75.

5.4 XG—Boost:

Extreme gradient boosting was applied by transforming the data type numeric for Police Dept. District and DayOfWeek. The parameters for the model was tuned after the first run to improve the performance of the model, the parameters like eta, gamma was increased, and max delta step was increased as the dataset was highly imbalanced.

The model was tested with 100 iterations and the minimum error was found at 92th iteration.

As earlier, we calculated both accuracy and logloss for our reference.

Parameter Tuning in XgBoost:

Here we set the objective to multi:softprob and the num_class to mlogloss.

These two parameters tell the XGBoost algorithm that we want to probabilistic classification and use a multiclass logloss as our evaluation metric.

The parameter multi:softprob objective also requires that we tell the number of classes we have with num.class which is 39 for us.

The other parameters of note are nrounds and prediction. The nrounds parameter tells XGBoost how many times to iterate.

The learning rate was tried for different values between 0 and 1 and we got our best result at 0.2.

Maximum depth was increased to 8 even though we were aware of the overfitting of the model but it gave us a better accuracy but the performance level deteriorated for the logloss evaluation.

We also altered the maximum delta step from 1: 6 as the dataset is highly imbalanced and it helped us to weigh the lower frequency classes and improved the performance of the model.

```
iter train_merror_mean train_merror_std test_merror_mean test_merror_std

1: 95 0.6227765 0.001673217 0.7429598 0.003165538

2: 96 0.6217300 0.001696940 0.7430197 0.003131105

3: 97 0.6206335 0.001766637 0.7430798 0.003186837

4: 98 0.6196170 0.001793703 0.7432298 0.002993724

5: 99 0.6182865 0.001795452 0.7431100 0.002873621

6: 100 0.6171802 0.001681101 0.7431998 0.003097023
```

```
>table(train$Category == train$pred)

FALSE TRUE

349011 529038
```

Accuracy

```
prop.table(table(train$Category == train$pred))

FALSE TRUE

0.386584 0.613416
```

5.5 Evaluation of the model:

Comparing various models we see that Decision tree model is highly biased with the majority values in the Larceny, Assault and non-criminal offenses. Hence the model is over fit for these categories while other categories are not predicated accurately given the less number of occurrences.

Model using Random forest algorithm is better than the decision tree and we have an error rate of 73% with a log loss of 2.75.

XG boosting has given better results compared to other three algorithms, the accuracy we achieved was 0.613416 which is much higher than other models. The model also predicted 529038 observations correctly.

6. Conclusion:

During the classification, we have seen that the classification has least log loss in Xg-Boosting and we would use this model on test set. We have received an 61% accuracy in the given set and log loss of 2.44.

Better results can be obtained by augmenting the data in to 4 segments viz- White collar crimes, Blue collar crimes, violent crimes and non-violent crimes. This results would be more easily predictable and we expect the accuracy to increase.