REAL-TIME CRIME PREDICTION ON A 911 CALL

IEOR 242: Final Project

Team Members:

Aman Tripathi
Mit Dhami
Tejas Baindur

Table of Contents

BACKGROUND

The screen signaling backlog of calls blinks red, continuously at San Francisco's 911 emergency call center. With a few clicks, the call comes through and the operator, hastily switches among computer displays, inputting a string of commands to dispatch help.
The seconds it takes him/her to answer and dispatch a call can mean the difference between life and death. The faster they can get help to the man, the better his chances of survival.

With a click, Operator would answer the next one. Click: a stabbing. Click: a pocket dial — false alarm. Click: a suicide attempt. The delay in answering those calls has the potential to leave people facing health emergencies or other dangers without the help they need in the time they need it.

In 2009, the San Francisco Police Department (SFPD) responded to 59,037 dispatched Priority A calls for in average response time of **3 minutes and 49 seconds**. Six years later, in 2015, officers responded to 81,342 dispatched Priority A calls for service in an average of **4 minutes and 59 seconds**.

Thus even a small aid, which can classify the type of crime from the call by just looking at the location through which the call came and the date and time when it came, can be very helpful for the operator. For example arson and lethal crimes can be immediately flagged and can help in rerouting and give visual aids to the operator and cut down the dispatch time. Even if it is a false positive call, the aid can reach quickly and check firsthand the situation.

Thus the motivation is to look through the location, extract features, extract the locality and run through the historical crime records and also predict the most probable crime type. There have been instances where the location is not accurately recognized but those cases are few, and thus for current scope we consider that there is no delay to transmit and recognize the location from which the call came through.

Also we can see in Figure 1 that response times are not meeting the standard response time, and national prescribed averages we do hope that our models would help in that aspect as well.

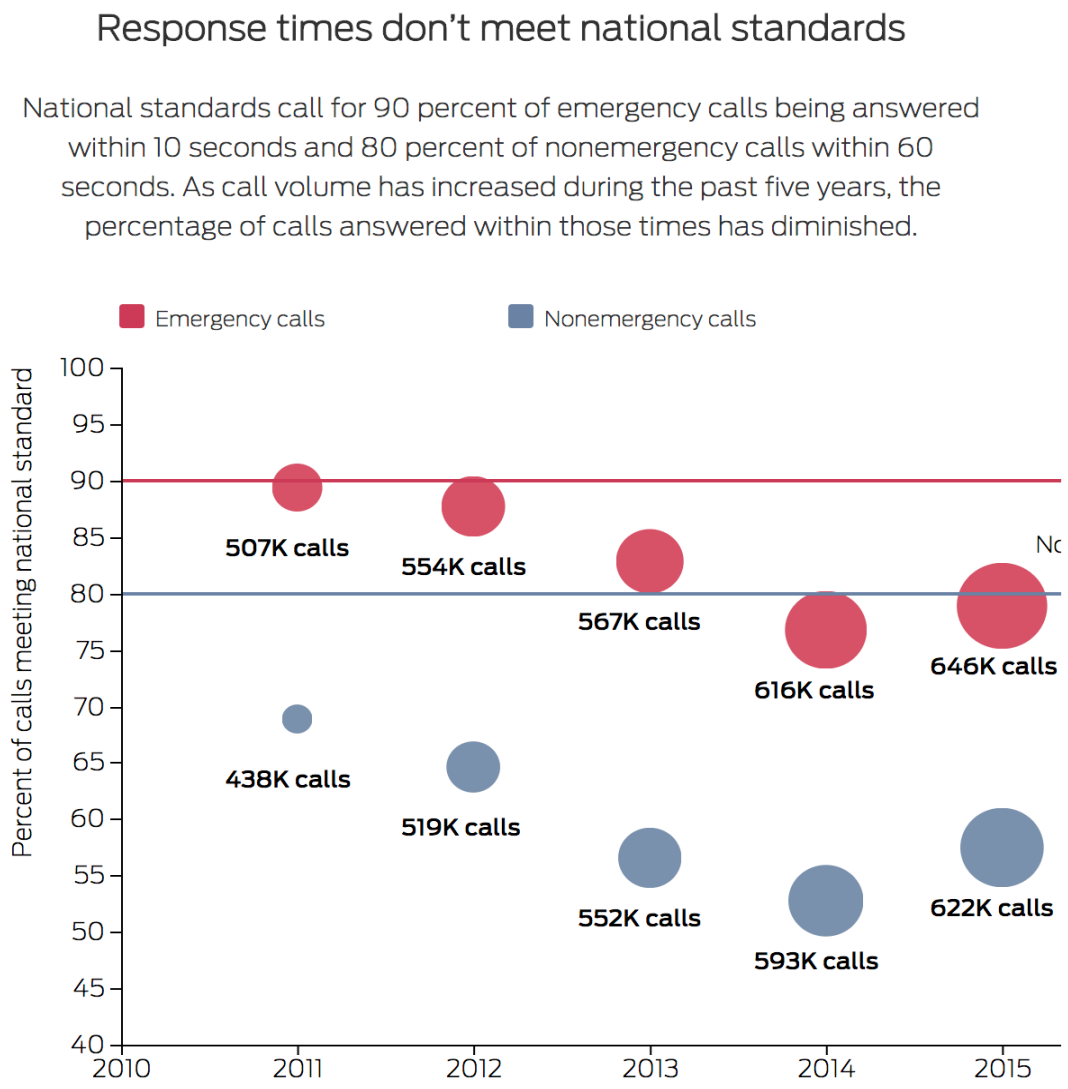

Figure 1: Percent of calls meeting the response time through the years

PRELIMINARY DATA

Preliminary Data is obtained from *data.sfgov.org* and *datasf.org.*
Following data is obtained for the city of San Francisco:

1. Neighborhood Analysis (Geographic)
2. SFPD Incident Reports
3. Business Registrations by areas (Proxy for richness, visitors and outside activity)
4. Crime Heat Map
5. Eviction Notices (Proxy for Ease of living, and neighborhood)
6. Land Use (Proxy for commercial/residential activity)

These all data files have either zip code, or location coordinates, which were turned into zip code by google API.

Also the features like population, density, wages and other proxy for financial well-being are used and extracted through reverse geocoding. Other cleaning, dropping null value and ensuring data sanity steps are in the notebook.

EXPLORATORY DATA ANALYSIS

Let us try and identify features and see if there is any natural cyclicity in time of type of crimes, their locations and any correlations or patterns between them. Also let us visualize the heatmaps, which would help us understand the patterns and see any variable trend throughout the day.

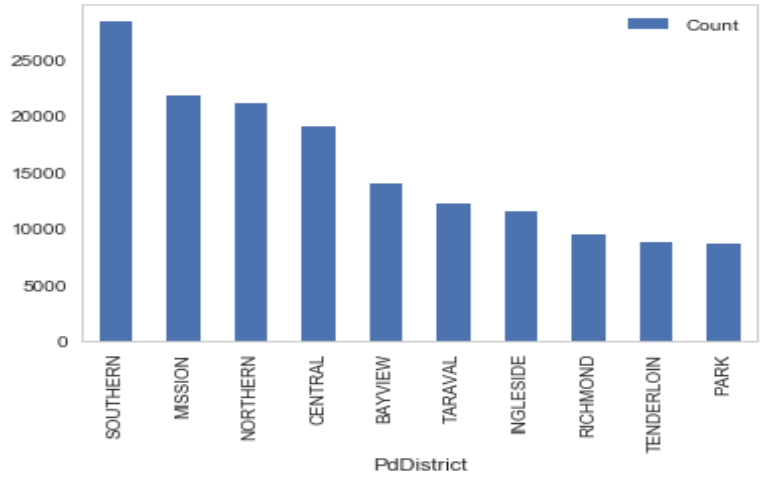

Figure 2: Number of crimes per district in San Francisco

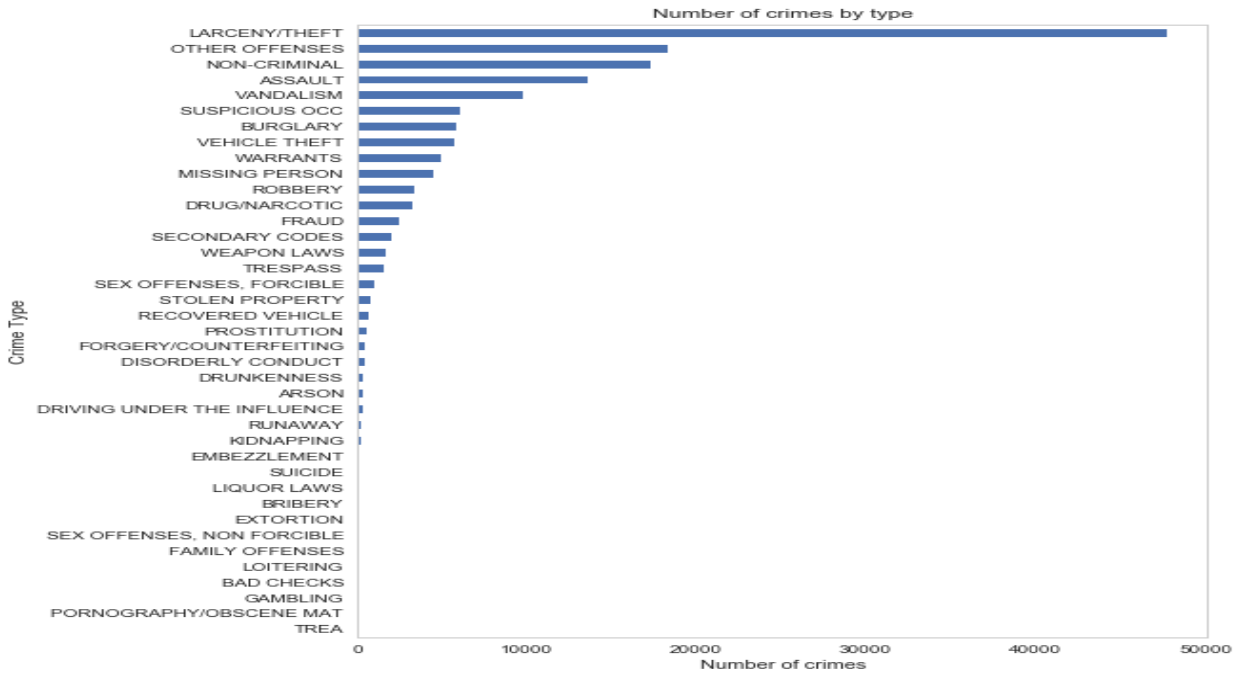

Figure 3: Number of Crimes by Type of Crime

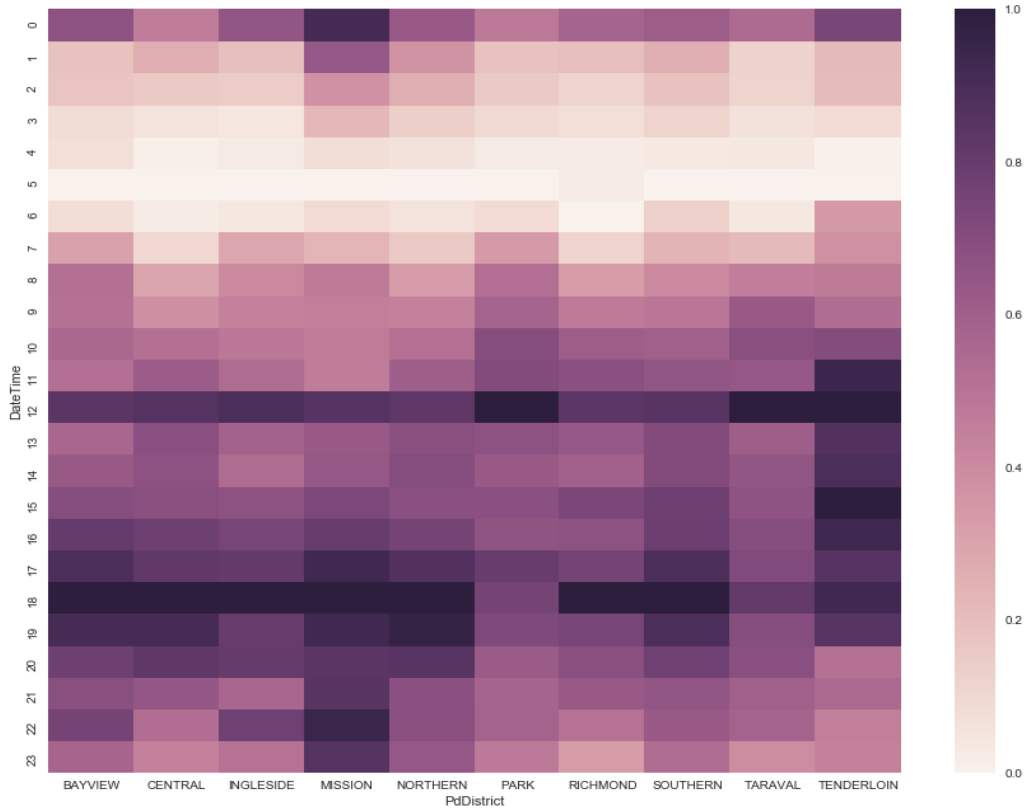

Figure 4: Crime Heatmap, for Time of Day vs District

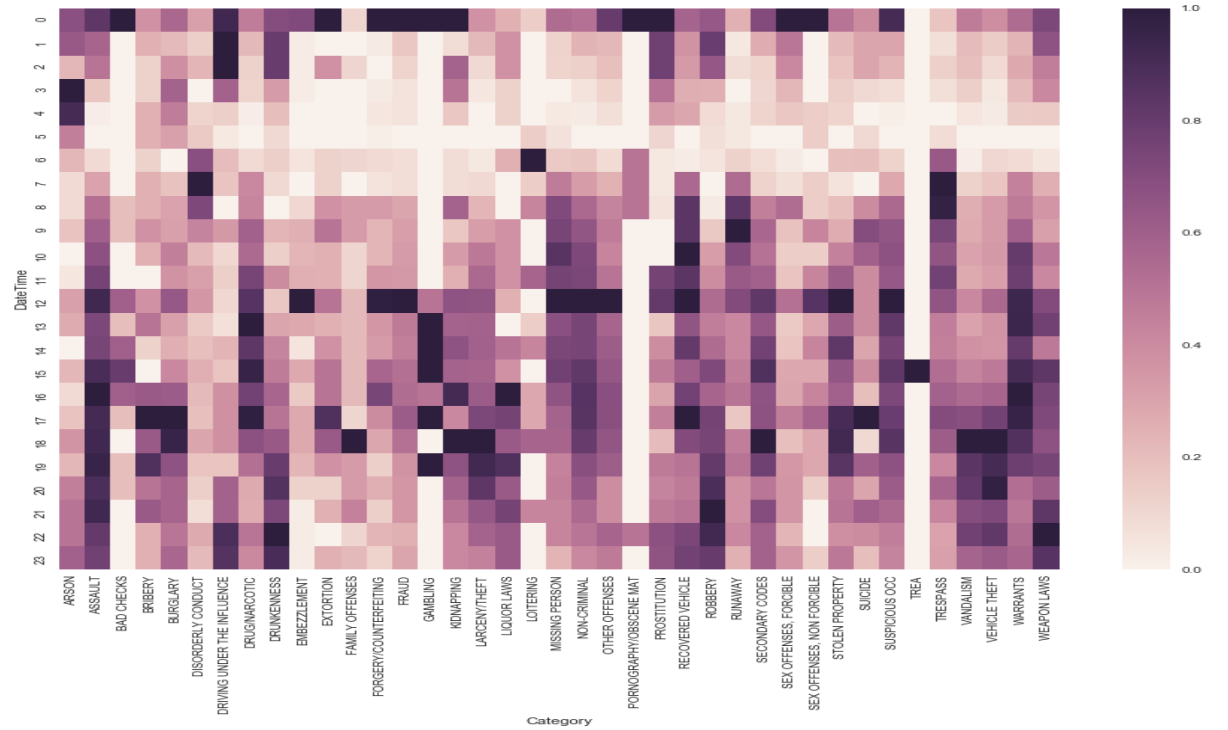

Figure 5: Crime Heatmap: Time of Day vs Type of Crime

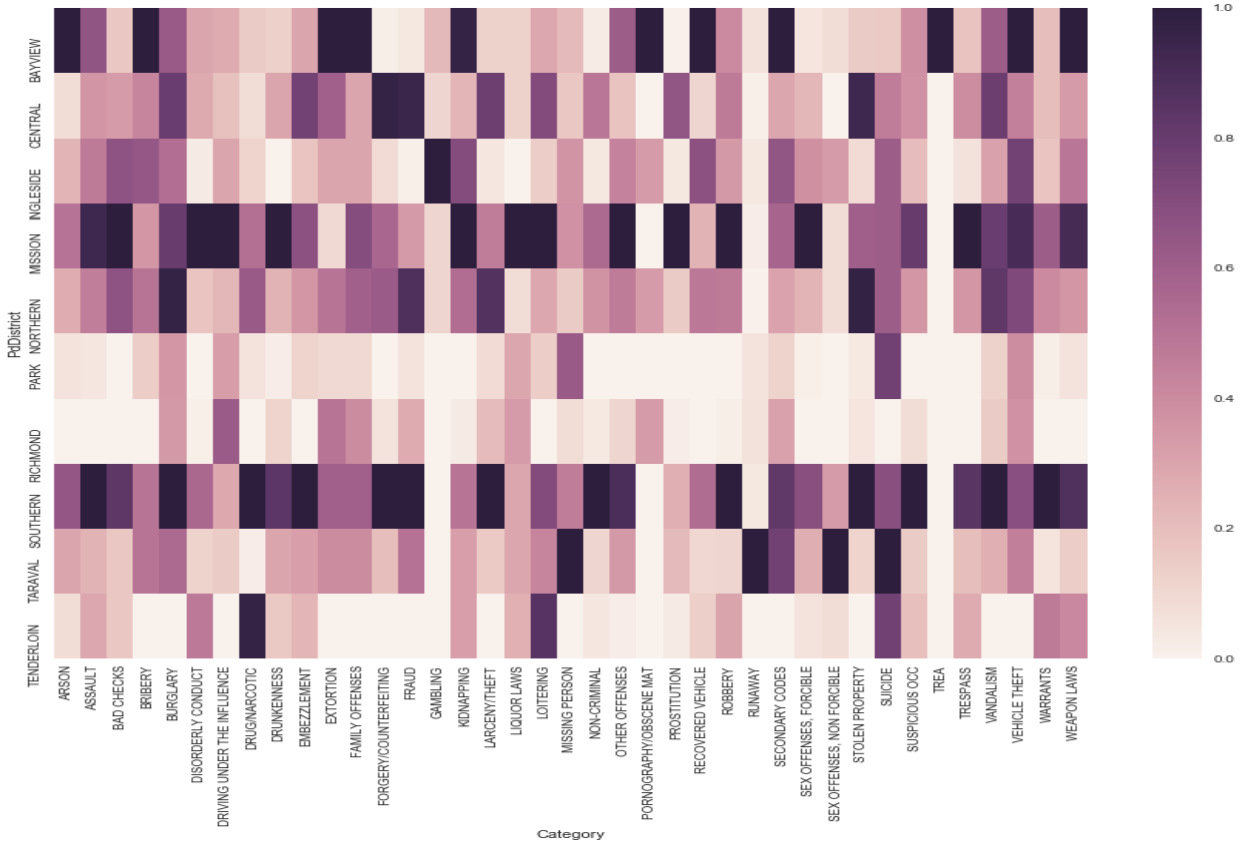

Figure 6: Crime Heatmap: District vs Crime Type

As we can see from the above and other graphs attached in code and appendix, there is fairly visible amount of trend, and crime type can be pretty much different during different time of day, day of week, month of year, part of San Francisco and also has a seasonal component within it. Let us also expand the feature space, and drop the correlated factors such as wealth, since that remains almost same by zipcode, and thus is not a pretty good proxy. We can instead just retain land use and businesses as counts, and make zipcode/district a categorical feature and other numerical to capture trend.

Also to avoid high dimensionality we must make sure not to one hot encode the timeofday, and other features since those could potentially explode the feature space. Let us try two different method to include continuous variable such as time

1. Binning
Let us create bins according to time of day (3 hours bins) since almost the pattern is same for this time. This would then be one hot encoded to avoid explosion of feature space

2. Continuous variable as sum of sine and cos
NCoeff1| timeofday = A sin(2*π*timeofday/23) + Bcos(2*π*timeofday/23)
NCoeff2|dayofweek = A sin(2*π*dayofweek/7) + Bcos(2*π*dayofweek/7)

We would continue the analysis with first method, as it is more convenient and can give us a better indication rather than a continuous variable.

REVISITING THE PROBLEM STATEMENT

As we can see from the data, the multiclass classification for all crimes would be pretty tough and thus the natural outcome is to breakdown problems into approachable parts
1. Binary Classification
- Theft or Not Theft
2. Multiclass Classification
- Classify among Theft, Other Offences, Non – Criminal, Assault and Rest

RESULTS AND INFERENCES

A. BINARY CLASSIFICATION

There is a huge class imbalance among the data set. Thus there can be two approaches to the problem, wherein we train the models directly or use balanced class weighted method wherein both classes are given equal weight.
In both cases the baselines would be pretty different. In case 1, the baseline is to predict not a theft always and it is around 73% accurate. However TPR is 0, and thus the focus is to improve this. Second baseline would be different since we are factoring in the class imbalance.

Below are the results for the different models, with class balancing.
We have only included Gradient Boosting without balancing, just to have an graph of relative feature importance of the boosted trees, so that we can compare it with xgboost plots and see if there is any scope to improve around.

XGBoost works well since there is a lot of trends unexplained and the residuals are fitted very well and many trends though intricate are captured well.

MODELS	*ACCURACY*	*PRECISION*		*RECALL*	
		No Theft	Theft	No Theft	Theft
Logistic Regression	60.45%	81%	36%	60%	62%
DTC	60.55%	81%	36%	60%	62%

RFC	60.76%	81%	36%	61%	61%
LinearSVC	60.32%	81%	36%	60%	60%
XGBoost	67.32%	79%	40%	76%	44%
Gradient Boosting (No balanced class)	73.28%	73%	49%	100%	0%
PCA + RFC	60.93%	81%	36%	61%	61%

As we can see from the Table above, XGBoost is pretty accurate for this task, and has interestingly a very good recall as well. However the recall for theft is low, which suggests that an ensemble method with RF and XGboost can be worked around to get pretty accurate. Moreover PCA is not much helpful and we can see that from the PCA plot attached which doesn't differentiate much anyway. Let us see XGBoost in detail and figure out what was most important while deciding the crime type. We can see that districts and time of days are pretty important while splitting.

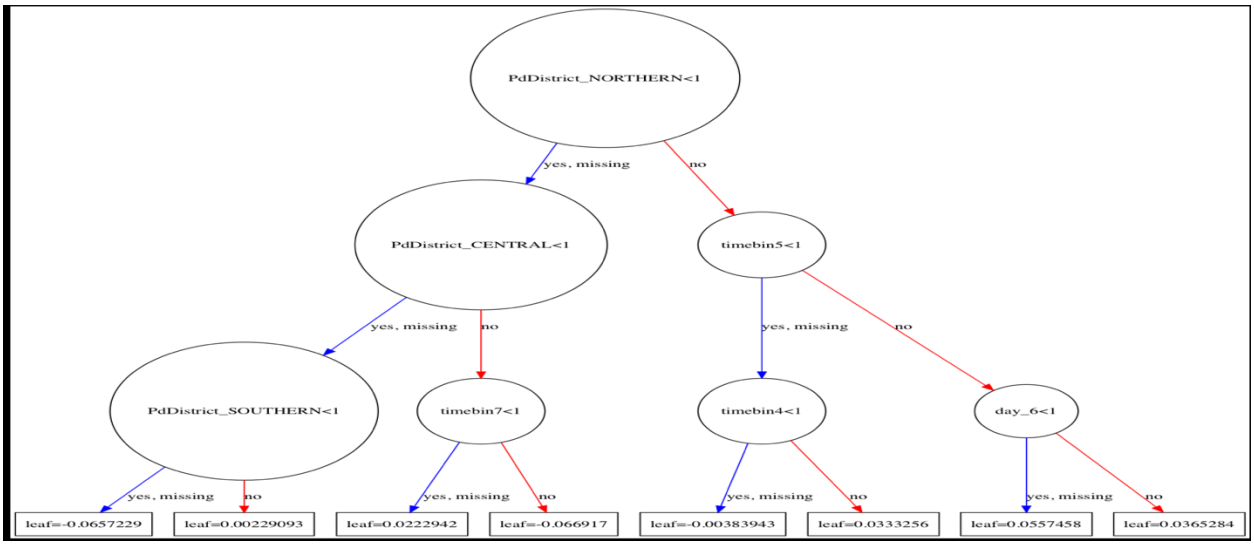

Figure 7: XGBoost Tree

Below is the confusion matrix for the algorithm results as well

	precision	recall	f1-score	support
0	0.79	0.76	0.77	168132
1	0.40	0.44	0.42	61292
avg / total	0.68	0.67	0.68	229424

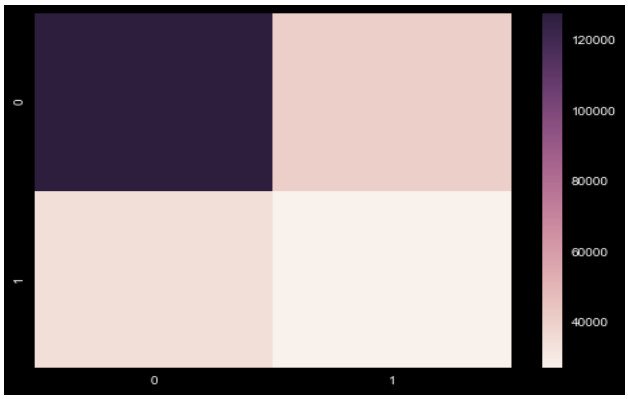

Figure 8: Confusion Matrix for XGBoost

Also, we can see similar features in the relative feature graph of Gradient Boosting algorithm.

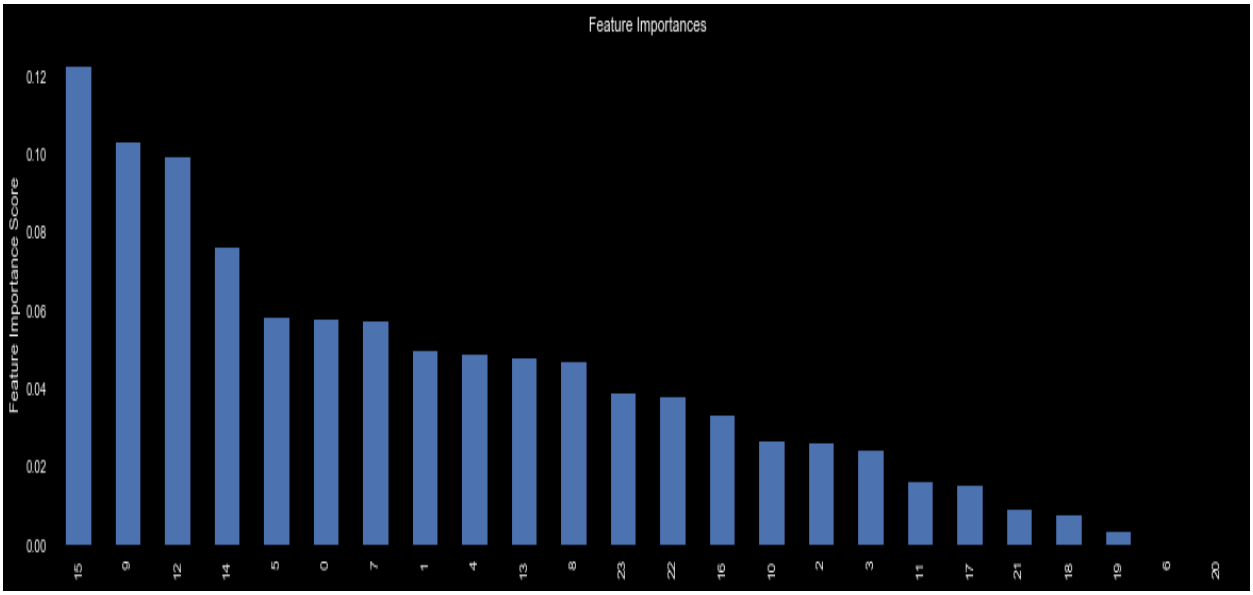

Figure 9: Relative Feature Chart in Gradient Boosting

B. MULTICLASS CLASSIFICATION

Let us create 5 classes, and try multiclass among this dataset: Theft, Other Offences, Non – Criminal, Assault and Rest

MODELS	AVG. PRECISION	AVG. RECALL
Logistic Regression	34%	25%
RFC	33%	26%
XGBoost	28%	42%

Results do look promising, but they have to be improved upon. More preprocessing and sampling techniques, as well as well-defined multiclass, multioutput definition can be worked upon as well.

CONCLUSION AND IMPACT

We have significantly improved the TPR in case of binary classification and thus it can be helpful to predict when theft is occurring. This can be expanded upon and violent crimes can be predicted in an more efficient way. Moreover the hyper parameters were not optimized fully and also the models were just balanced by default library class weighing methods. The balancing, up sampling down sampling and other techniques can also be tried out. Also the dataset was large and thus it couldn't run kernel svm and other cv methods. Moreover ensemble methods

could be applied as well. In case of multiclass the imbalance is a major issue, and weighing methods have to be improved. We can see recall for 5 crimes is decent but there is a lot of similarity and correlation and thus the misclassification rate is high. Even a smaller but accurate model and high TPR would help the operator take note and attend distress early. More over the analysis can be improved by collecting more features of the calls, such as arrest rate, lethal crimes and also nature of it. We did also drop out huge chunks of specific locational data by just using the districts, which is to be noted. If we divide the blocks into smaller groups models can be improved.