IIITB ML Project: SFO Crime Classification

Team - moto users
Nikhil Pappu - IMT2016035
Puneeth Sharma - IMT2016018
Baswanth Modugu - IMT2016080

IIIT Bangalore

Introduction

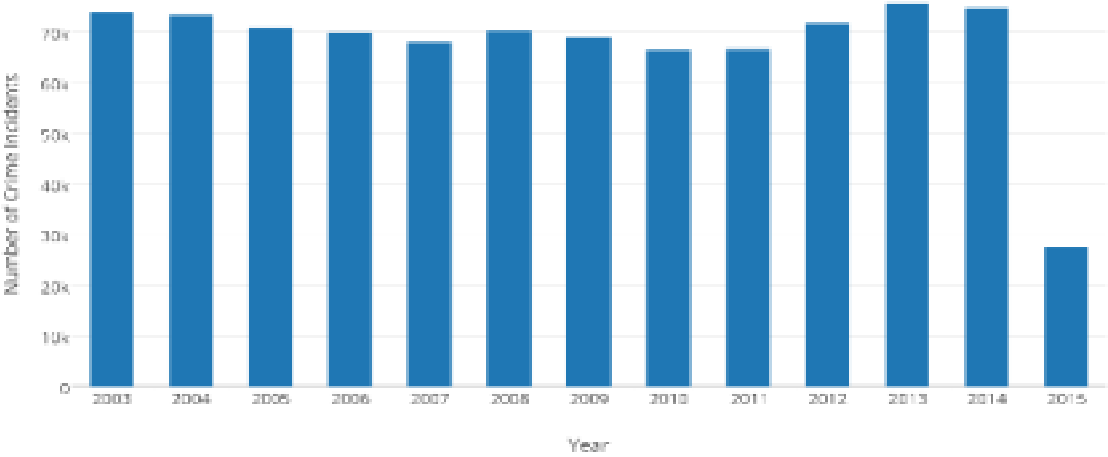
From 1934 to 1963, San Francisco was infamous for housing some of the world's most notorious criminals on the inescapable island of Alcatraz.
Today, the city is known more for its tech scene than its criminal past. But, with rising wealth inequality, housing shortages, and a proliferation of expensive digital toys riding BART to work, there is no scarcity of crime in the city by the bay.
The San Francisco Police Department published this dataset providing nearly 12 years of crime and their relevant record, to encourage people to mine more facts from it. Kaggle hosted a competition using this dataset at
https://www.kaggle.com/c/sf-crime.
A classroom competetion with a similar dataset was hosted at
https://www.kaggle.com/c/IIITB-ML-Project-sfo-crime-classification
the results of which are discussed here.

1. Dataset Exploration

1.1. Overview

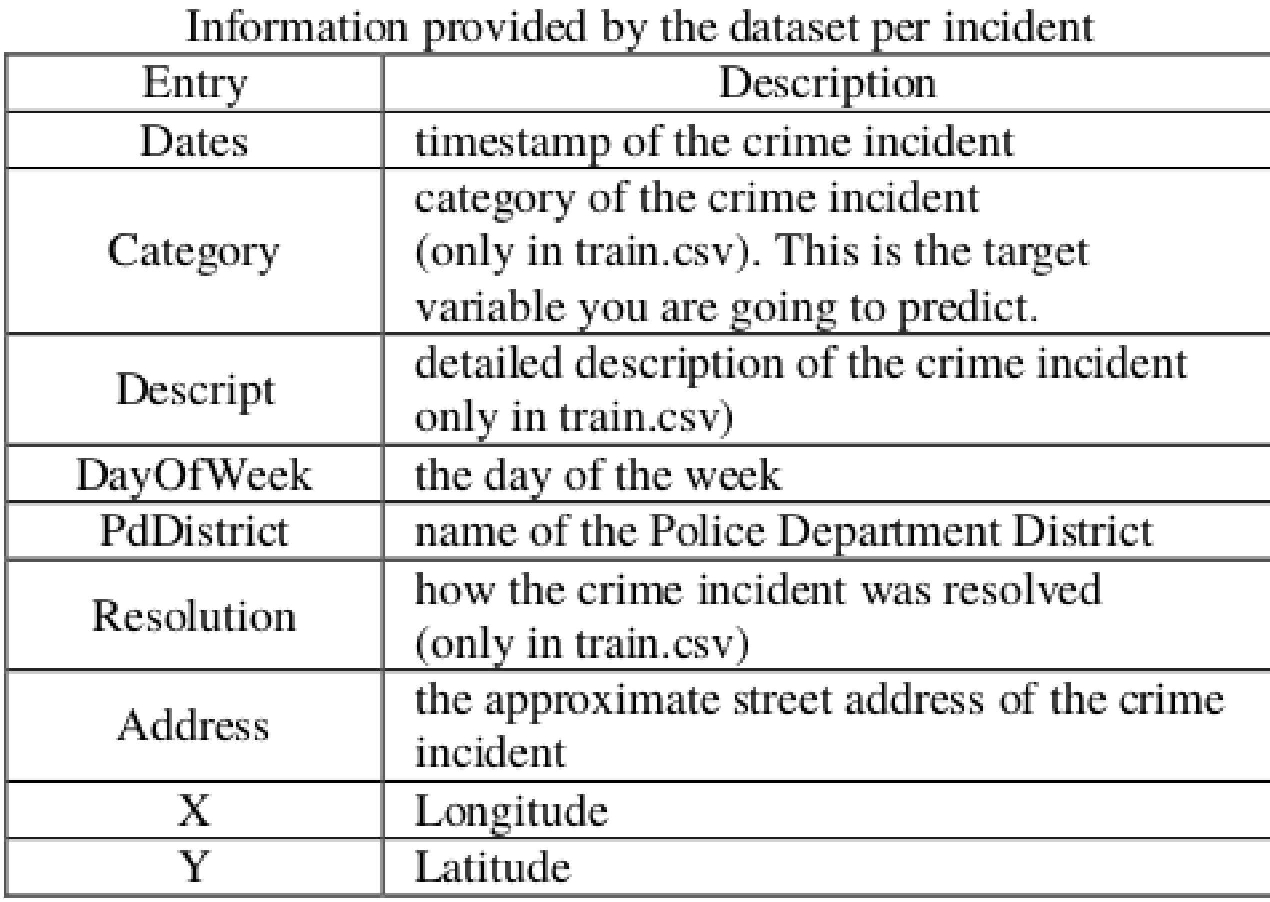
Time period the dataset covers is from 1/1/2003 to 5/13/2015 with 878,049 data points in total.Kaggle has divided dataset into train and teast datasets.Kaggle has divided all odd weeks belong to test set and even weeks belong to train test set.

Preprint submitted to IIIT Bangalore *March 21, 2019*



There are 39 categories of crimes in our dataset.

shows how the total number of crime incidents changes over years. A general trend of reduction can be observed from 2003 to 2011. However, the number of crimes starts to increase since 2012. Another thing to note here is the data of 2015 is not complete since it only covers 5 months record. In the dataset, each crime record has nine entries of information related to the incident which are shown in TABLE 1.



Information provided by the dataset per incident

Entry	Description
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

above figure shows how crime incidents distribute in different categories. It is clear that the distribution is uneven.Crimes such as Larceny/Theft, Other Offenses, Non-Criminal,Assault,Drug/Narcotic and Vehicle Theft take huge portions while at the meantime certain kinds of crime such as Sex Offenses Non-Forcible, Gambling, Pornography/Obscene Mat are extremely rare.

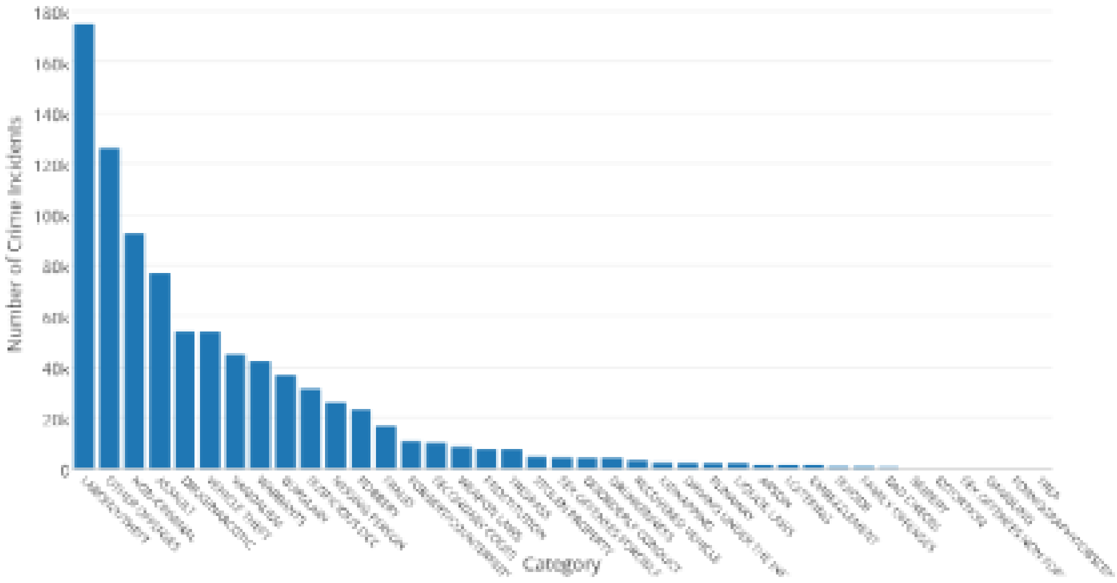
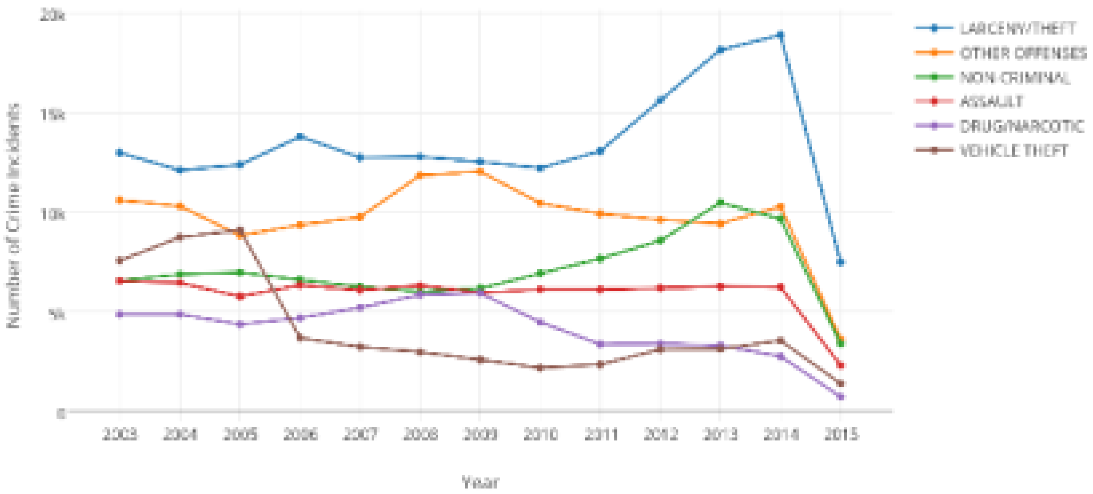


TABLE 1. The above image shows all the attributes given in the dataset.

1.2. Variable relation to time

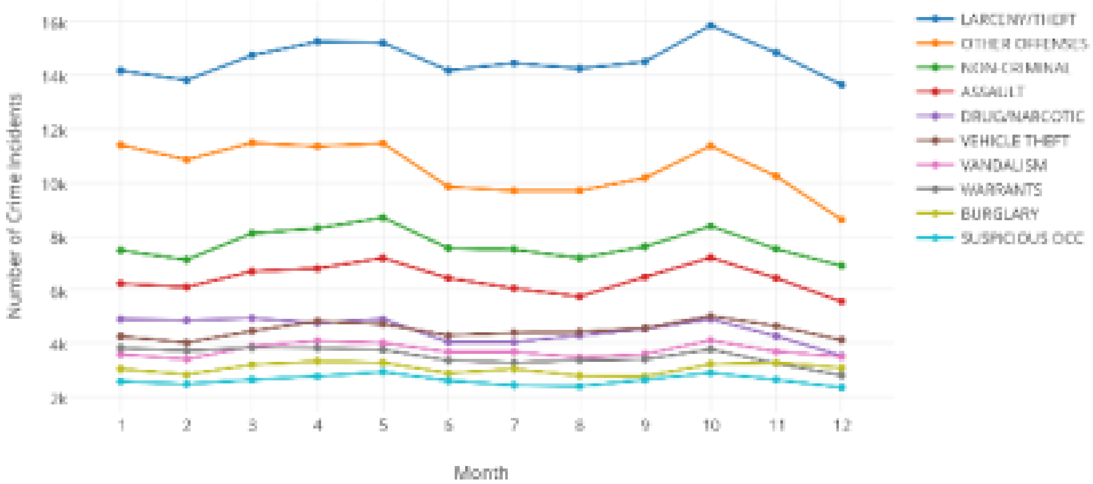
Time is always the first thing we mention when we report an incident. Here we want to figure out if time variables have effects on crime classification. Time variables related to a crime incident include information in the Dates entry and information in the DayOfWeek entry. The Dates entry in the dataset provides a incidents timestamp with a format of year-month-date hour:minute:second. Considering date, minute and second are trivial in effecting crime classification, only year, month and hour are researched here. The DayOfWeek provides on which day of week a incident happened, it is another time variable that is researched.

1.2.1. Year



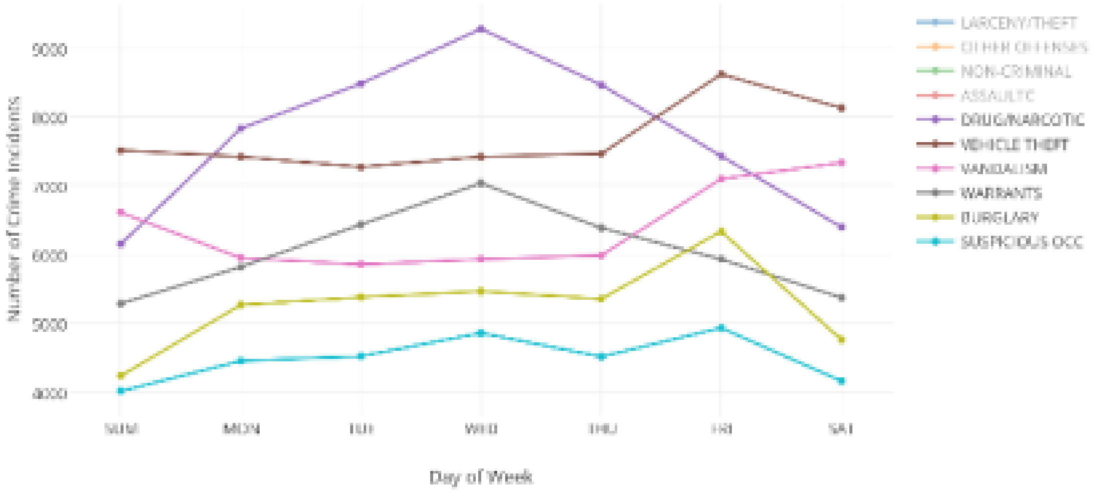
The above figure shows numbers of top six commonest crimes changes over years. From it we can see that in some degree, the proportion of different kinds of crimes changes in different years. For example, the proportion of Vehicle Theft drops to a low level since 2006, Other Offenses and Drug/Narcotic clearly take a bigger proportion in 2008 and 2009 compared to that in other years, the proportion of Non-Criminal incidents reduces from 2003 to 2008 then start to increase and it finally peaks in 2013. So we can suppose that the year variable could be a critical feature in crime classification model.

1.2.2. Month



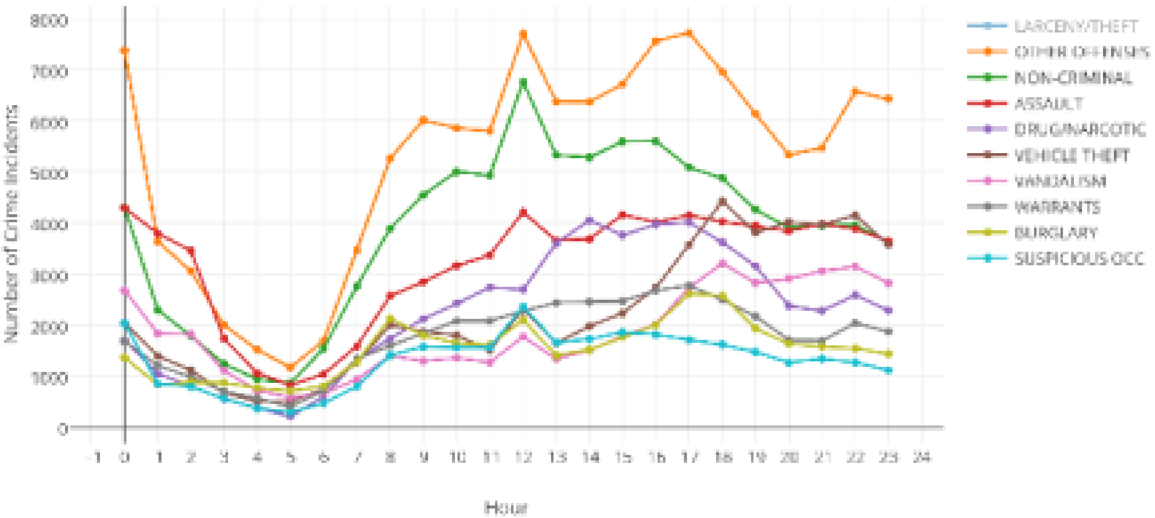
The above figure shows numbers of top ten commonest crimes changes over month. Variance of proportion among crime categories here is not very obvious since all categories of crime follow a roughly same trend over different months. So, month may not be a useful variable in the classification model. However, the trend all crimes follow should be noticed. Heres an interesting phenomenon that the total number of crime incidents tends to be lower in summer and winter but higher in spring and fall. We may infer that extreme whether can reduce crimina activities.

1.2.3. Day of week



The above figure we only let the fifth to tenth top commonest crimes show because their numbers are similar in scale so the change on proportion can be more easy to capture. It can be found that Drug/Narcotic and Warrants both peak on Wednesday and come down at the start and the end of a week while Vandalism is just the opposite. Also, the highest occurrence of Theft, Burglary and Suspicious activity are all on Friday. So day of week can be a critical variable in deciding the category of a crime incident.

1.2.4. Hour



The above figure shows number of top ten commonest crime with the 1st one shadowed because it is in a huge scale of number compared to others and it would make the change not so obvious. In the figure we can observe some proportion changes among different crime categories: Vehicle Theft rushes to a high level at 6 pm from a rather low one, Warrants and Drug/Narcotic start to drop from 5 pm and they are the only two crimes that do not peak at 12 pm while other crimes all tend to do so. Hence we may want to try hour variable in our classification model. We can also easily catch a fun phenomenon that in three to five oclock in the morning the occurrence of criminal incidents is the lowest among the whole day.

1.3. Variables Related to Location

Location is another important feature of a crime incident. It may have crucial effect on predicting crime categories too. In the dataset we have four entries which are PdDistrict, Address, X and Y describing the location of incidents. Here we will research how we can use these four entries to make effective predictions.

1.3.1. PdDistrict

PdDistrict stands for police department district.According to the dataset, therere ten of them.

1.3.2. Address

Information in the address consists of a street full name, and a street suffix abbreviation. Some good Information in the address consists of a street full name, and a street suffix abbreviation. Some good instances could be: 1400 Block of GOLDEN GATE AV, 200 Block of EVELYN WY, or MENDELL ST / HUDSON AV. In above instances, AV, WY, ST are all street suffix abbreviation. Its difficult to figure out a good use of the street full name for it could be redundant with the latitude and longitude variable or with the PD District variable. Also, if you treated it as a category feature, there would be a huge vector for there are too many of different street in a city and it could make the model very difficult to be optimized. However, in this dataset, we found that there are only 15 different street suffix abbreviations. According to the reference table of US PostalService[5], the meaning of these 15 abbreviation are

2. Prediction Task

2.1. Task Description

The predictive task is to predict the category of crime given the time and location information. We are to predict crime from a large number of classes making this a multi class classification problem.

2.2. Evaluation

We have used a train test split of 80-20 for our validation testing and used it for deciding between models and to tune hyperparameters. Each incident in the data set has exactly one true labeled class. For each incident in the validation set, we calculate probabilities of it belonging to every category. Prediction results are then evaluated using the multi-class logarithmic loss:

$$Logloss = -1/N \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} log(p_{ij})$$

where **N** is the number of cases in the validation set, **M** is the number of class labels, log is the nature logarithm, y_{ij} is 1 if observation **i** is in class **j** and 0 otherwise, and p_{ij} is the predicted probability that observation **i** belongs to class **j**.

2.3. Feature Extraction and Data Preprocessing

We have used one hot encoding for all but one of our features mentioned below which resulted in our feature space blowing up to about a 100 features. We tried dimensionality reduction methods like PCA and tried selecting few of the most important features but we found that this wasn't necessary as it didn't improve our results. So we went with all of the features after doing the one hot encoding. The features we have used are as follows:

1. **Year**: This was found to be quite an important feature which was not surprising because trends like these tend to change especially over a period of over a decade. We have therefore extracted this from the 'Dates' entry of the data set. We have taken this as a categorical feature as the range was only 12 years which resulting in a one hot encoding into 12 columns.

2. **Month**: This feature was also found to be quite important and was extracted from the 'Dates' entry as well. We have used a one hot encoding of 12 columns here as well.

3. **Hour**: Hour was also quite and important feature as a lot of crimes do not usually happen later in the day. We have extracted this from 'Dates' and used a one hot encoding of 24 columns.

4. **DayOfWeek**: This feature was given in the dataset. We found that more crimes tend to happen on Fridays and weekends. We have simply used a one hot encoding of size 7 here.

5. **PdDistrict**: This feature was given to us in the dataset and is an important location feature and we have used one hot encoding again.

6. **Street Type**: This was a feature we had extracted from 'Address' as it was one of the few things in 'Address' with a small range of possible values which represented the type of the crime location (street, boulevard, crossing etc). We have extracted this be taking the last two characters of this column and used one hot encoding once again.

7. **Simultaneous Crimes**: This is an implicit feature of the data set. This was an important feature as it separates crimes which go with

other crimes like 'Drunkenness' and ones which occur in isolation like 'Larceny/Theft'. We have summed up the values repeating for both 'Address' and 'Dates' for each data instance and used one hot encoding as the maximum number of simultaneous crimes was 5.

8. **Evening**: We have used this boolean column which differentiates based on Hour before or after 18 (6:00 pm). We had considered create 4-5 bins but it didn't make much sense as we already had an Hour feature. This feature did not help all that much due to its overlap with Hour.

9. **Season**: We have binned the Month feature into Summer, Autumn, Spring and Winter here. We then one hot encoded it. This one again didn't help all that much due to its heavy overlap with Month.

10. **Coordinate**: This was the only continuous or non categorical feature we had used. We initially used a standard scaler on the given 'X' and 'Y' latitude and longitude columns of the data set but later switched to using the `from_lat_lng`() function of the s2sphere library which properly encodes coordinates based on latitude and longitude. This gave us much better results.

3. Model Design

We had considered a lot of models some of which we dismissed outright and some others we tested against each other. We finally decided to go with XGBoost, Logistic Regression and Random Forests as they gave the best results. We did not consider any deep learning methods like neural networks because the problem at hand was a simple supervised learning problem with a good data set and we didn't find a need for these. So we considered only classical models, some of which are:

1. **Logistic Regression**: This was the first classifier we had considered. It gave us good results and converged much faster than other models. This showed us that the underlying data was fairly linearly separable and we needed to get the preprocessing right to get better results. We also validated a lot of the features on this model so that the grid search

didn't take too long.

2. **Support Vector Machines**: The SVM Classifier took surprisingly longer to converge and gave results similar to other models. We therefore went with the Logistic Regression Classifer as it converged faster and because the data was linearly separable.

3. **Random Forests**: This was our go to model until we found out that XGBoost did better. It was quite easy to determine the parameters as we had to tune lesser parameters compared to XGBoost the number of estimators and max depth. They took long to converge but gave us good results.

4. **Boosting Trees**: We tried both XGBoost and AdaBoost after tuning some parameters. XGBoost did better than AdaBoost and gave us the best score so we went with it.

5. **K Nearest Neighbours**: We wanted to see if we could use a non parametric model atleast as part of a pipeline but KNN wasn't very successful because it was quite slow and inaccurate due to the highly imbalanced dataset.

6. **Naive Bayes**: We had decent results with Naive Bayes but it was clearly performing worse when we added features which had considerable overlap with other features like 'PdDistrict - Coordinate', 'Month - Season', 'Hour - Evening' as it assumes that the features are all independent. We therefore didn't go with this.

4. Results and Conclusions

4.1. Optimization

We got a resultant logloss of 2.212 when we used XGBoost, 2.228 with Random Forest and 2.240 with Logistic Regression using K-fold cross validation on our 80-20 train test split and K = 5. . We used Grid Search Cross Validation in scikit-learn to tune our hyperparameters. We found that a value of C=1 ($\lambda = 0.1$) worked best for Logistic Regression, number of estimators = 300 and max depth = 20 for Random Forests. For XGBoost we used max depth = 5, eta = 0.4 and sub sample = 0.9.

4.2. Coclusions

We went with all of the features we had mentioned above as feature reduction and PCA didn't help us. We found that using s2sphere for encoding the latitude and longitide worked better. The implicit feature Simultaneous Crimes was quite effective and so was the Street Type which we extracted from the Address. Some features like Season and Hour helped only a little bit as they had considerable overlap with other features. This was a reason we didn't use Naive Bayes as mentioned above. We also didn't find much use for non parametric methods like KNN or ANN here. XGBoost clearly outperformed the other two models although not by a lot. This showed that data preparation and preprocessing were a lot more imporant for this problem than the choice of the models. We got a final score of 2.215 on the competition using XGBoost and ranked 7th out of 21 teams on the leaderboard of the competition.