Capstone Project Report

San Francisco Crime Classification

Mentor: Srdjan Santic

By Anushree Srinivas

Problem to be solved and motivation:

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.

From Sunset to SOMA, and Marina to Excelsior, this competition's dataset provides nearly 12 years of crime reports from across all of San Francisco's neighborhoods. The objective of this project is to predict the category of crime that occurred at a given time and location.

Client:

City and County of San Francisco is trying to make the city and the neighborhoods safer. The City authorities as well as the San Francisco Police Department will be the client for whom the conducted data analysis as part of the capstone project will be beneficial.

Data:

The data was obtained from:
https://www.kaggle.com/c/sf-crime/data

There are **two** data sets in this project which I worked on:

1) **Train**.csv: The data set on which I worked on and deriving insights. This data set has **9** columns: Dates, Category, Descript, DayOfWeek, PdDistrict, Resolution, Address, X(latitude) and Y(Longitude).

2) **Test**.csv: The data set for which predictions of the crime category are to be done.

Important fields and Information in Dataset:

The date column has dates ranging from 2003-2015 with the time that the crime took place. Category column briefly states the category of crime as given by the Police Department. There are 37 categories in this data set. Descript gives an in-depth description of the category of the crime that was committed.

PdDistrict involves the districts in the San Francisco area which belong to different police departments. There are 10 different PD districts. Address column has two type of address one is either an intersection which is given by a '/' and residential location which is specified by 'of'.

Data Cleaning and Wrangling:

The first thing I did was to check for missing values in all the data sets. I used the **isnull()** function to check for null values in the data set and also tried to get the percentage of the null values in train data set.

To find the missing values I first found the **tota**l for each data set using the. **isnull() and .sum()** functions. This gave me the number of null values for each column in the train data set. There were no null values in this data set so therefore, there was no cleaning required to be done. I then proceeded to do the exploratory data analysis.

Exploratory Data Analysis and Insights generated:

I first started by looking at the distribution of the number of crimes occurred vs the day of the week. Using, **value_counts()** and **sns.countplot()**, I plotted the graph. Then since the data has dates, I converted them into hour, month and year using the **to_datetime()** function. Now using this, I decided to check the trend of crime activities with respect to the hour, month and year (2003-2015).

The lowest number of crimes is in 2015 which is because we have data only until May 2015. The number of crimes reached an all-time high in the span of 12 years in 2013. The number of crimes was high at 2003 after which it was decreasing until 2008 where we can see a spike again after which it decreased until 2012-2013. We can see that every 5 years there's a spike in crime rate.

The crime rate is maximum on Friday and Saturday and least on Sundays. The months of May and October recorded a high number of crimes. In terms of the time, the maximum crime activity is around 12pm and in the 5-7pm window the maximum being at 6pm.

I then went to explore how the crime activity affects the pdDistrict. So using **value_counts()** I plotted the graph of the crime activities vs regions(pdDristrict). The maximum crime rate was recorded for the Southern region followed by the mission region. Then I moved on, to find the top 5 crime categories.

Since in the top 5 categories, there were “other offenses” and “non-criminal” categories which doesn’t give us much insight so I decided to drop using the **drop ()** those two rows to get a better idea about the top 5 categories. The 5 categories being Larceny/Theft, Assault, Vandalism, Drug/Narcotic, Vehicle/Theft.

Since the crime rate was maximum for Friday, Saturday and Wednesday and lowest on Sundays, I decided to look at the hourly distribution of the crime activities specifically on these day. From the graphs for Saturday and Sunday I saw that, the crime activity increases in the weekend at night on Saturday from 10pm-12AM. Whereas on Friday the crime activity is most at 6pm. This might be due to the fact that the city is overly crowded with people coming to the city and leaving the city around this time.

Then to get a better idea about the district specific crime category, I used a stacked bar graph to plot the top 5 crimes across the pd Districts.

Finally, I proceeded to see the frequency of the number of crimes for the top 5 categories in terms of day of the week and hour of the day. The burglary rate is maximum on Fridays so checking the hourly activity on Friday, I saw that 5pm on Friday is most prone to burglaries.

Although the overall crime recorded is the least on Wednesday the number of crimes due to drug or narcotic is most on Wednesday and specifically at 2pm. The larceny category has most number of crimes recorded on Saturday followed by Friday. Looking at the hourly distribution I saw that for Friday its maximum at 6pm whereas on Saturday it's at its peak at 11pm. The maximum crime activity recorded due to vandalism is on Saturday at 10pm.

Inferential Statistics:

For inferential statistics, I performed two different normality tests for each of the following variables: hour, year, month, day_num, X,Y. Shapiro-Wilk test couldn't be conducted since the sample size is > 5000). So I proceeded to do the Anderson-Darling, and the D'Agostino tests. To further check if any of the variables are correlated, I tried a correlation plot for the train dataset.I used **sns.heatmap()** and the **corr()** functions to get my corresponding correlation plot. There was a high positive correlation between X and Y of 0.56. The correlations between other variables were not that significant.

Feature Engineering:

I developed some features from the date that was given and tried to couple them to get some extra features. Using the **pd.to_datetime** function I extracted the day(day of month as Day), hour, minute, dayofweek(day_num), weekofyear, dayofyear, year, month etc.

Since the dataset already has the dayofweek column and since the crime rates are max on Friday and least on Sunday. I derived a weekend feature. Then by defining the **get_season** function, I extracted the season and categorized them into fall, winter, spring or summer to get better insights.

I then proceeded to define an address function called **define_addresss** which categorizes the address. If the address given is an intersection, then the add_num variable would be 1 and 0 if it's a residential location. To know if it's an intersection or a residential location I used / and of as indicators. If the address has '/' and 'of' is not in it, like for example OAK ST/LAGUNA ST then it would be a 1 and the address 1500 Block of LOMBARD ST which has an 'of' it would be a 0.

For the latitude's and longitudes, X and Y I standardized them using the **sklearn preprocessing function** and standardizing it with the latitudes and longitudes of San Francisco which means for the X(latitude) I added 122.4149 and for the longitude(Y) I subtracted by 37.7749.

Then for the category and PdDistrict variable, using the **pd.categorical** function, I categorized them into a numeric value. Then, I standardized the year by subtracting them with 2003(taking 2003 as the baseline).

The final features were:

- X
- Y
- District_num (PD District)
- add_num (Address)
- minute
- hour
- DayOfYear
- Year
- Month
- Day
- Day_week
- WeekOfYear
- Fri
- Sat
- Sun
- Weekend
- Fall
- Winter
- Spring
- Summer

Algorithm and Findings:

After the initial exploration and findings were concurrent with the claims that the features given in the dataset as well as the new features obtained through feature engineering would help us achieving good accurate results in predicting the category of crime in San Francisco. The next step was to build machine learning models and obtain predictions on the test set. The first machine learning algorithm I applied was the RandomForestClassifier

My goal was not only say if an object belongs to one of the 37 categories, but to also provide the probability that it belongs to these classes. Log Loss quantifies the accuracy of a classifier by penalizing false classifications. The whole idea is to minimize the log loss which will lead to maximum accuracy on our classifier. So if we say, the classification is neutral and assign equal probability to each classes, the log loss would be less for e.g. around 0.8. In the second case, let's

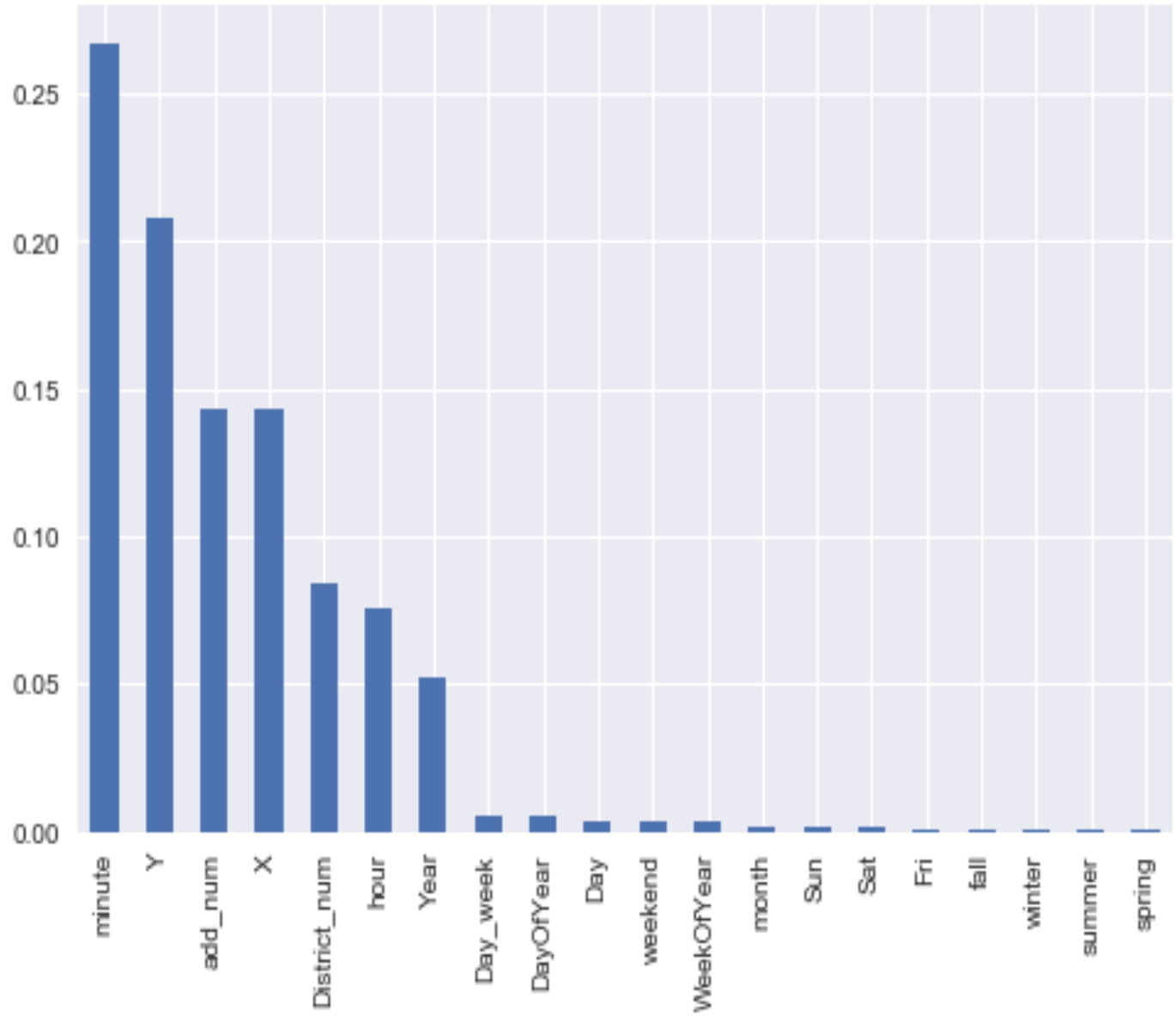
say the classifier is relatively confident in the first class. Since this is the correct classification the Log Loss is reduced to say 0.10. The third case is an equally confident classification, but this time for the wrong class. The resulting Log Loss increases to say 2.5. We would ideally want a lower log-loss score since that would mean it's a better model. Keeping this concept in mind, I went ahead with my first algorithm.

Random Forest Classifier: I perfomed a Grid Search cross validation for the RandomForestClassifier and got the best parameters with max_features="log2", n_estimators=24. The log-loss score for this model was 2.41 using all the features that I had derived.

Then by using the another set of features which was a subset of the features already derived and used it to run the next random forest model. The features being, X,Y, District_num, add_num,minute,hour, Year,month,Day, with the same parameters. The log-loss score for this model using these set of features decreased to 2.385 (a decrease of 0.025).

AdaBoost Classifier: The next model I implemented was the AdaBoostClassifier, using all the features and the parameter for this model was n_estimators=24. The log-loss score that was obtained was 3.51 which was higher than both the random forest models. To get a better log-loss score for this model, I then did a cross-validation to tune the parameters and use the best parameters. I used the best parameters obtained from the cross validation and got the log-loss score of 2.83, which is still higher than the random forest model but for the AdaBoost Classifier there was a decrease in the log-loss score by 0.68.

Feature Importances: To get a better idea of which features are the most significant ones I used the feature_importances_ attribute to get the top 5 important features as minute, Y co-ordinate(Longitude), Address, Latitude and the PdDistrict. The graph of the entire list can be seen below.



Recommendations:

1) Knowing the category of the crime, we can better equip the police force, district wise for better response time.
2) Residents could be made conscious along with encouragement to take steps and measures to be secure.
3) Policymakers can put forth local policies to ensure the safety of the districts and neighborhoods.
4) For law enforcement agencies that are considering adopting predictive policing tools, situational awareness can be increased.
5) The police force can be distributed according to the day and type of crime in a particular neighborhood thereby reducing the extent of the damage.
6) Advertisement and awareness campaigns can be conducted in different areas for the area specific crime to involve the citizens and work with them.

Future Research and Work:

The parameters can be tuned more to get a better log-loss score. Better algorithms which work well with multi-category prediction can be used to get more accuracy. To obtain a model that achieves a better log loss score than above, we can try running nonlinear models. The three models that were implemented here were all linear models. A nonlinear model could provide us with better predictions on our test dataset as compared to linear models, that's something I would really like to take a look at. Though the features That were created and used allowed me to get decent predictions, I could create new features in order to see if better results can be achieved. New features created could help us generalize better on the test dataset. In conclusion, these are the two aspects which I would like to work on to achieve better results and a more accurate prediction.