San Francisco Crime Data Analysis and Classification

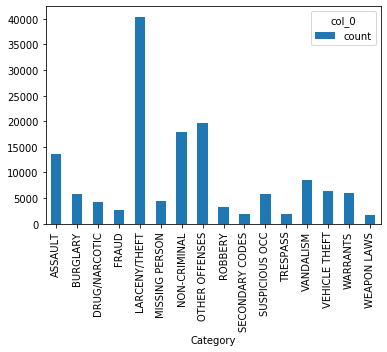
SF Crime Analysis for 2016

Author: Dylan Karman

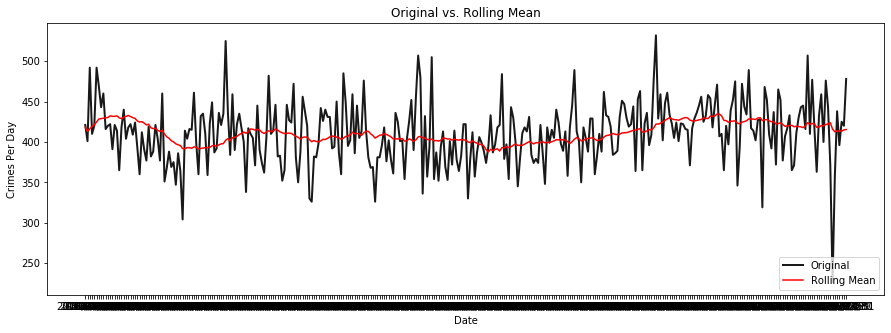
Link To Project: https://github.com/dylankarman/Capstone

Having grown up in Northern California, I was always fascinated by San Francisco. I am from a small town and SF seemed so massive to me. The buildings were tall, the driving was crazy, and there was way more people there than I thought could be possible. However, with more people comes more problems, and more specifically, more crime. Having a Bachelors Degree in Criminal Justice interested me in this aspect of SF, and got me thinking, ‘How can SF better allocate their resources within the Police Department so that they can spend more time on being proactive, instead of reactive?’ In order to this, I obtained the SF Crime data for the year 2016 from Kaggle. The data can be found here: <https://www.kaggle.com/roshansharma/sanfranciso-crime-dataset>

I first performed an analysis of the data using pandas, a built in data analysis library within Python. There was only one missing data point in the entire DataFrame, so I was able to fill that easily using the .ffill() function and begin the analysis. There was no risk of contaminating the data with faulty data points because it was only one value. By graphing the data with the .value\_counts() function, I was easily able to group together the different values within the column and see what the data represented visually. This allowed me to make some assumptions: In the year 2016, San Francisco saw 150,000 reported crimes. 27% (40409/150000) of the crimes reported were in the larceny/theft category. Not surprisingly, Friday and Saturday (the weekend) saw the most amount of crime, accounting for 30% of the crime in only 28% of the week (2 days/7days). New Years Day is the single day with the highest amount of crimes by a large margin. New Years Day accounts for 0.3% (558 crimes) of all crimes throughout the entire year, which is quite a lot considering that if you averaged out the 150000 crimes by 366 days (leap year), it equals 0.2% (409 crimes). Almost 150 more! 12:00 (noon) saw the most amount of crime happening. "Most crimes commited in big cities occur during the day, while more violent offenses happen more frequently at night, according to a new study." This makes sense seeing that the vast majority of crimes are minor larceny/theft. This study can be found here: <https://www.usnews.com/news/cities/articles/2019-06-12/study-finds-crime-in-big-cities-is-more-likely-during-the-day>. When it comes to the resolution, or what the officer does when they get to the scene, is very one sided. Out of 150000 crimes reported in the year, over 100000 of them end up in the 'None' column. Officers are called to the scene and are sent away without having done anything with a legal outcome. October is the month that has the most reported crimes. All of these columns within the DataFrame offer an in depth look into the where and when the most amount of crime is going to happen. By looking at this data, Police Departments can better allocate their resources to fit the needs of the community if they know that a high crime rate day is on the horizon.

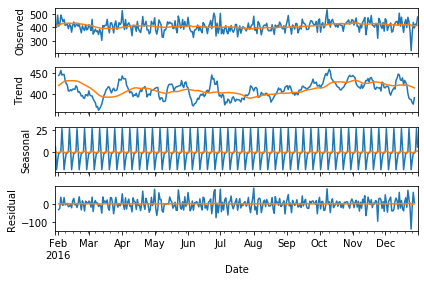


\*An example of a .plot(). This particular example is for the category of crimes. Using the graph, you can easily see what crimes happen the most.



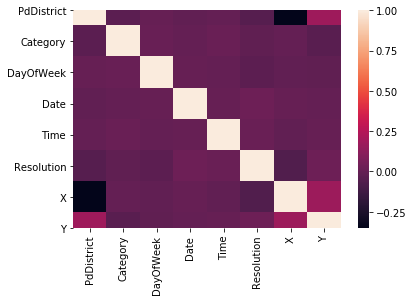
\*The original data (black) plotted with the rolling mean (red).

A seasonal decompose algorithm was used on the data to see if there was any seasonality to it. The seasonality is important because it determines whether there is some sort of pattern in the data. I calculated the rolling mean (orange line) in order to capture the trend of the data, which is very flat. An 'additive' model was used in the seasonal decompose algorithm, because the data was not trending in an upwards or downwards fashion. When the seasonal decomposition was finished, there was a very high seasonal trend shown with the graph and the dramatic spikes. The residual plot is filtering out the noise from the data. If there was no noise, the plot would be flat.



\*The seasonal decompose model

Classification is a way to predict what may happen in the future based on the correlations made between the data and a target variable. This can be useful knowledge when trying to predict where crimes might happen, and placing necessary resources there before it happens. The classification was done on the PdDistrict and the, 'Category', 'DayOfWeek', 'Date', 'Time', 'Resolution', 'X', and 'Y' columns were used as the variables. The categorical values (data represented with words) had to be changed into numbers so that they can be graphed, using a .LabelEncoder(). The data was then seperated into two DataFrames. One DataFrame only had the target variable (PdDistrict), and the other DataFrame contained the variables. The data is then run through a train test split where 70% of the data was trained on a logistic regression model and 30% was tested on it. When that is complete, the training data is compared with the testing data and a classification report is produced. In information retrieval, precision is a measure of result relevancy, while recall is a measure of how many truly relevant results are returned. A number close to 1 is ideal. The F1 score is an average of the both of them. The accuracy of the model when predicting where crimes will happen in the future was 87%.

\*A heat map is used to determined the correlations amongst the target variable and the other variables.

precision recall f1-score support

0 0.91 0.93 0.92 4291

1 0.93 0.88 0.90 5300

2 0.89 0.89 0.89 3478

3 0.87 0.89 0.88 5851

4 0.90 0.81 0.85 6030

5 0.78 0.86 0.82 2610

6 0.87 0.89 0.88 2677

7 0.96 0.81 0.88 8533

8 0.92 0.91 0.92 3397

9 0.63 0.98 0.77 2983

micro avg 0.87 0.87 0.87 45150

macro avg 0.87 0.89 0.87 45150

weighted avg 0.89 0.87 0.88 45150

Accuracy Score: 87.3 %

\*The classification report is a great way to see if the model is accurate. Numbers closer to 1, are ideal.