San Francisco Crime Classification

Article · July 2016

CITATIONS
0

READS
269

1 author:

Yehya Abouelnaga
Technische Universität München
5 PUBLICATIONS **12** CITATIONS

SEE PROFILE

San Francisco Crime Classification

Yehya Abouelnaga
School of Sciences and Egnineering, Department of Computer Science and Engineering
The American University in Cairo, New Cairo 11835, Egypt
devyhia@aucegypt.edu

***Abstract*—San Francisco Crime Classification is an online competition administered by Kaggle Inc. The competition aims at predicting the future crimes based on a given set of geographical and time-based features. In this paper, I achieved a an accuracy that ranks at top %18, as of May 19th, 2016. I will explore the data, and explain in details the tools I used to achieve that result.**

I. Introduction

San Francisco first boomed in 1849 during the California Gold Rush, and in the next few decades, the city expanded rapidly both in terms of land area and population. The rapid population increase led to social problems and high crime rate fueled in part by the presence of red-light districts. However, the San Francisco of today is a far cry from its origins as a mining town. San Francisco has seen an influx of technology companies and their workers. While this has resulted in the city being acclaimed as a technological capital, the gentrification of its neighbourhoods have not been entirely well-accepted. It comes as no surprise that a tech-savvy city like San Francisco have decided to publicly release their crime data on their open data platform, and this data is part of an open competition on Kaggle to predict criminal occurrences in the city [1].

II. Dataset Analysis

The San Francisco Crime Classification dataset contains the following set of features:

- **Longitude** – X coordinates on the map where the crime occurred.
- **Latitude** – Y coordinates on the map where the crime occurred.
- **Address** – The address of the crime incident.
- **Day of Week** – The day of the week (i.e. Thursday)
- **Date** – The date of the crime in the following format: *YYYY-mm-dd hh:MM:ss*. Thus, you can deduce the following: Year, Month, Day, Hour, Minute, Second.
- **District** – Police district to which the crime is assigned.
- **Resolution** – The resolution taken to address the crime.
- **Category** – The type of the crime. This is the target/label that we need to predict.

The previous features are provided in the training set. However, in the data set you don't have **Category**, and **Resolution** columns. You do have an extra **Id** column for the purpose of Kaggle submissions.

In our training, we will remove the **Resolution** column because it's associated with the Category (our label/target). Thus, it's not provided in the test dataset, as well. Now, we will explore the features one by one.

A. Data Distribution

1) Hour: After experimenting, this turned out to be the most important feature. It provides the highest correlation with the crime category (our target label). This is extracted from the feature *Dates*.

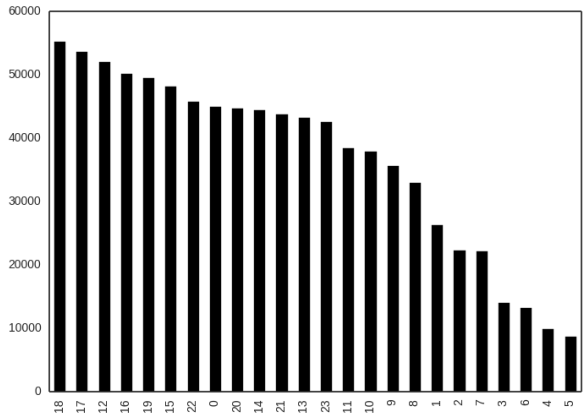

Fig. 1. Crime Distribution Per Hour

2) District: There're 10 districts in San Francisco. Some of them are more endowed with crimes than others.

District	Number of Crimes
SOUTHERN	157,182
MISSION	119,908
NORTHERN	105,296
BAYVIEW	89,431
CENTRAL	85,460
TENDERLOIN	81,809
INGLESIDE	78,845
TARAVAL	65,596
PARK	49,313
RICHMOND	45,209

3) Day Of Week: Crimes seem to be almost evenly distributed across days of the week. They increase on Fridays, though. Friday night parting culture might have an impact on that spike.

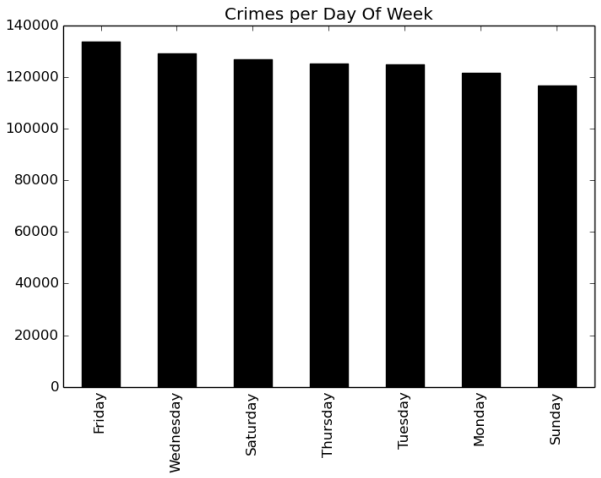

Fig. 2. Example of a parametric plot $(\sin(x), \cos(x), x)$

4) Address: This feature is very sparse among the entire dataset. There are 23,228 unique address in the training dataset. Thus, it's hard to encode these values. The most frequent address is *800 Block of BRYANT ST*. We notice that most of the popular crime places contain the word "BLOCK" in them. Thus, when used in classification, it gave better results.

5) Category: This is the target label/crime we want to predict. We've 39 crime categories (i.e. classification classes). The distribution of training sample is very skewed as you can see in the table below.

District	Number of Crimes
LARCENY/THEFT	174,900
OTHER OFFENSES	126,182
NON-CRIMINAL	92,304
ASSAULT	76,876
DRUG/NARCOTIC	53,971
VEHICLE THEFT	53,781
VANDALISM	44,725
...	...
EMBEZZLEMENT	1,166
SUICIDE	508
FAMILY OFFENSES	491
BAD CHECKS	406
BRIBERY	289
EXTORTION	256
SEX OFFENSES NON FORCIBLE	148
GAMBLING	146
PORNOGRAPHY/OBSCENE MAT	22
TREA	6

6) Street Number: This is a feature extracted from the Address. About 300,000 of the address have no street number, thus, got a zero value. The remaining 550,000 crimes have street numbers distributed over 80 different numbers. Note that, we had about 26,000 unique addresses. Meanwhile, we have only 80 unique street numbers. This feature increased the classification accuracy.

7) Block: This is another feature extracted from the Address. It only has binary values: block or non-block. There're 617,231 block-based crimes. And, other 260,818 non-block-based crimes. This feature increased accuracy as well.

III. Evaluation Metric

The metric used to evaluate the quality of the classifier is the multi-class logarithmic loss, as indicated below.

$$logloss = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} log(p_{ij})$$

IV. Principal Component Analysis

In order to increase the classification accuracy, and avoid overfitting, we used PCA to reduce the dimensionality. We noticed that the PCA performs best with the 3 components.

In the following table, we tested the classification accuracy after using PCA, with 1, 2, and 3 components.

# of Components	Log-loss
1	2.5103915
2	2.5094789
3	2.5056236

V. Classification

In this section, I will explore different classification models, and compare their accuracy.

A. Features

We used the Hour, Month, District, Day of Week, Longitude, Latitude, StreetNo, Block, and 3 components of the PCA that preserve the highest variance.

B. k-Nearest Neighbors Classifier

This classifier achieved high results only under high values of k.

k	Log-loss (Validation)
39	6.341270679
100	3.966297590
500	2.837714675
1000	2.703951916
2000	2.645995828
3000	2.628225854
4000	2.621394788

C. XGBClassifier

This classifier is a very robust regressor. It's very fast, and achieves good results. Here are the results with changes in the *max_depth*.

max_depth	Log-loss (Test)	Log-loss (Validation)
3	2.57896	2.573599576
4	2.57896	2.573666882
5	2.57428	2.568169744
6	2.57428	2.565219931

The results show that the change of depth didn't achieve much better results.

D. Decision Tree Classifier

Decision Tree Classifiers are very fast compared to the previous ones, and surprisingly, more accurate for this classification problem.

max_depth	Log-loss (Validation)
4	2.60898971
5	2.592237171
6	2.585709647
7	2.584480513
8	2.597382051
9	2.634747421
10	2.536769

It's noteworthy that *DecisionTreeClassifier* performs best at $max\_depth = 10$. We proceeded to experiment with the number of elements in the tree, while fixed $max\_depth = 10$. And, we found that the it performs best when $max\_depth = 10$, and $n\_elements = 256$. This set of parameters yields a log-loss of 2.50849507.

E. Bayesian Classifier

This classifier didn't prove to be the best for this set of features. It sets a baseline of 2.64923294 on the validation set.

F. Random Forrest Classifier

Random Forrest Classifier proved to be the best for the job. After parameter tuning, at $max\_depth = 13$, it performs best. Now, we have to tune the $n\_elements$ parameter.

n_elements	Log-loss (Validation)
10	2.441049707
20	2.422189106
30	2.424566886
40	2.420590797
50	2.416913452
100	2.412998032
150	2.411652466
190	2.411337273
200	2.410574498
200 + StreetNo	**2.370072457**
200 + StreetNo + Block	**2.366175731**

VI. Kaggle Scores

After submitting the results to the Kaggle competition, our best classifier placed at 388 out of 2077 (as of May 19, 2016), with a log-loss of **2.39031**. This is among the **Top 18%** on the leaderboard. This result is higher than those achieved in [1], [2], [3], and [4].

VII. Future Work

We still think that we can achieve much higher accuracy when we employ more feature engineering on the Address field. [5] suggested that we use Learning by Counting, explained in [6], in order to generate a log odds feature that might be useful as [5] claims. [5] also suggested that we use a Neural Network to train on the data. He achieved a much higher accuracy with the use of a Neural Net with 512 hidden layers. We need to experiment with the concept of classifier fusions, mentioned in [7].

VIII. Conclusion

In this paper, I explored a wide spectrum of possible classifiers that might be a good fit for solving the San Francisco Crime Classification problem. We achieved a log-loss metric that was higher than most of the published solutions, with subtle feature engineering, and classifier parameter tuning.

References

[1] S. T. Ang, W. Wang, and S. Chyou, "San Francisco Crime Classification," *University of California San Diego*, 2015.

[2] J. Ke, X. Li, and J. Chen, "San Francisco Crime Classification," *University of California San Diego*, 2015.

[3] A. Chandrasekar, A. S. Raj, and P. Kumar, "Crime Prediction and Classification in San Francisco City."

[4] C. Haskell, "Kaggle Competition: San Francisco Crime Classification," 2015. [Online]. Available: https://brittlab.uwaterloo.ca/2015/11/01/KaggleSFcrime/

[5] C. Papadopoulos, "Predicting Crime Categories with Address Featurization and Neural Nets," 2015. [Online]. Available: https://www.kaggle.com/c/sf-crime/forums/t/15836/predicting-crime-categories-with-address-featurization-and-neural-nets/103225

[6] Microsoft, "Data Transformation: Learning with Counts," 2016. [Online]. Available: https://msdn.microsoft.com/en-us/library/azure/dn913056.aspx

[7] D. Ruta and B. Gabrys, "An Overview of Classifier Fusion Methods," *Computing and Information Systems*, vol. 7, pp. 1–10, 2000.