San Francisco Crime Classification

Wenbin Zhu
Computer Science and Engineering
wez180@eng.ucsd.edu

Wenjie Tao
Computer Science and Engineering
w1tao@eng.ucsd.edu

Yuchen Wang
Computer Science and Engineering
yuw520@eng.ucsd.edu

ABSTRACT

Crime classification helps police to reasonably allocate resources and keep the city safe. In this paper, we aim to predict the category of past crime incidents given time and location from San Francisco crime report dataset. We extract time and geographical features with Google's S2 library and represent them with one-hot encoding. Besides a baseline constant classifier, Naïve Bayes, logistic regression and random forest classifier are applied and find-tuned. Random Forest gives the best performance of logarithm loss 2.29 on testing best.

KEYWORDS

Crime Classification, Naïve Bayes, Logistic Regression, Random Forest, S2

1. INTRODUCTION

Today San Francisco is famous as a technology center rather than a criminal hotbed, but there is still no scarcity of crimes by the bay. Based on past 10 years' crime reports of San Francisco's neighborhoods, we want to predict the category of crime that occurred given time and location.

In section 2, a brief introduction of the dataset and findings from exploratory analysis is provided. Besides basic statistics, we visualize crime distribution over time and space to help discover potential patterns. In section 3 we formally describe the predictive task and logarithm loss function to evaluate prediction. A naïve constant classifier is used as the baseline. We talk about features tried and selected in section 4. Section 5 lists three models we employed for the classification task, including Naïve Bayes, Logistic Regression and Random Forest classifier. Experiment results and comparison between classifier are given in section 6. Related work about crime prediction and conclusion are respectively in section 7 and 8.

2. DATASET

2.1 Basic Information

The dataset we use is from "San Francisco Crime Classification" contest on Kaggle. It contains crime records derived from SFPD crime incident reporting system ranging from 1/1/2003 to 5/13/2015. The training and testing set rotate every week, which means incidents in week 1, 3, 5… belong to the testing set, incidents in week 2, 4, 6… belong to the training set.

In specific, there are 878049 records in training set and 884263 records in testing set; all records are sorted by time. Each record in training set has 9 fields:

Dates - timestamp of the crime incident

Category - category of the crime incident

Descript - detailed description of the crime incident

DayOfWeek - the day of the week

PdDistrict - name of the Police Department District

Resolution - how the crime incident was resolved

Address - the approximate street address of the crime incident

X - Longitude

Y - Latitude

We randomly shuffle the original training data and split it into training set, validation set and testing set with size 70%, 15%, 15%.

2.2 Exploratory Analysis

As the first step, we count the number of incidents of each crime category. As can be seen from figure 1, the most common 6 types, theft, other offenses, non-criminal, assault, drug and vehicle theft account for about 66% of all crimes committed. Crime incidents satisfies long tail distribution and it makes sense to pay more attention to controlling these 6 types.

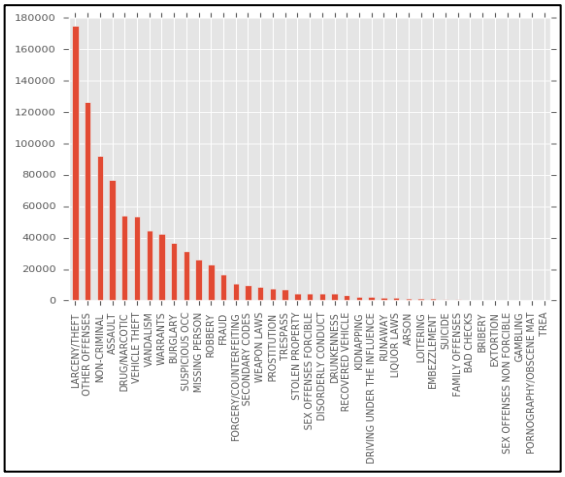

Figure 1 Occurrence of Each Category of Crime

By intuition, time is an essential factor to predict

category of crime incident. For 6 most common crime categories, we visualize how their occurrences vary with year (Figure 2), month (Figure 3), day of week (Figure 4) and hour (Figure 5). Though month and day of week don't seem to be useful—distribution is roughly uniform, year and hour lead to some interesting findings. For example, number of drug use and vehicle thefts is decreasing in recent years while theft keeps growing. In an ordinary day, crime rate falls to lowest point at early morning (5:00 AM). There are two peaks every day, a smaller at noon (12:00 AM) and a bigger at dusk (18:00 PM).

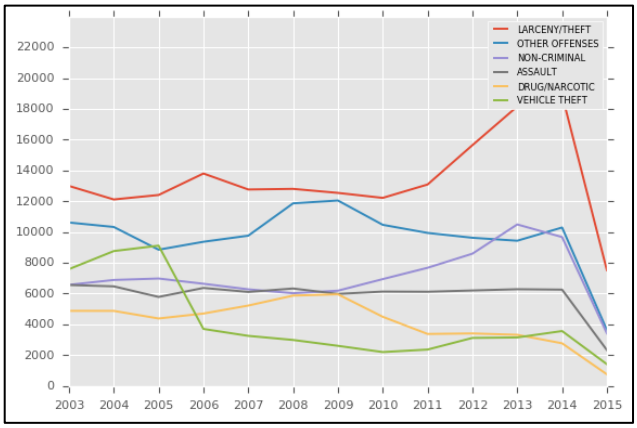

Figure 2 six most common crimes with year

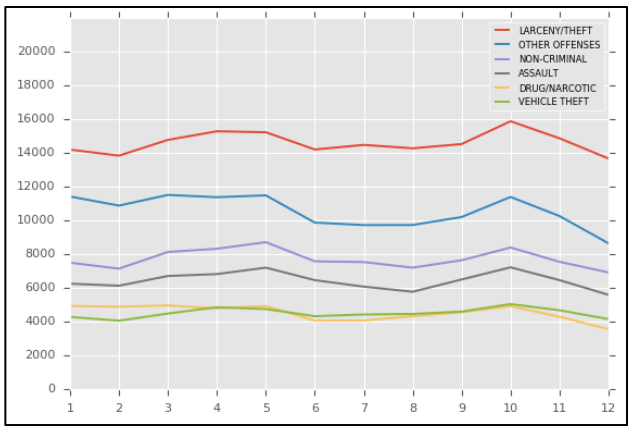

Figure 3 six most common crimes with month

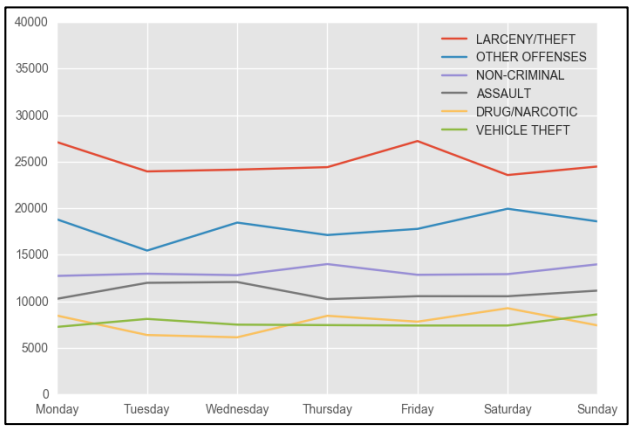

Figure 4 six most common crimes with day

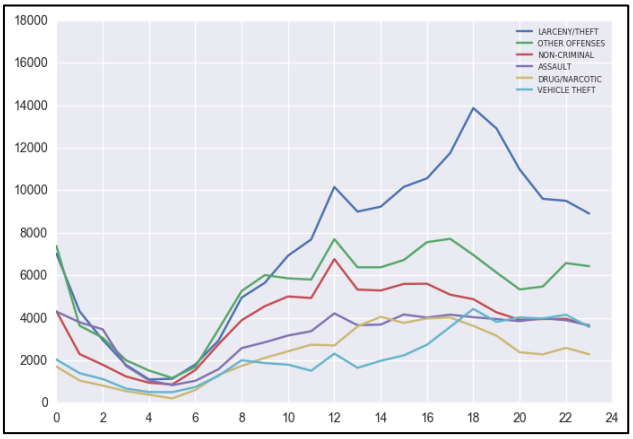

Figure 5 six most common crimes with hour

Moreover, we study PdDistrict and X/Y to exploit geographical information. As in figure 6 below, MISSION police department deals far more drug-related crimes than other departments.

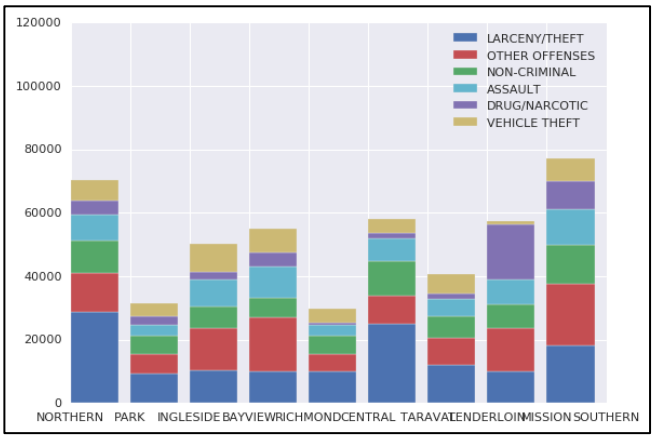

Figure 6 six most common crimes with police district

Meanwhile, different crimes tend to lie in different "regions". For instance, we use seaborn's kdeplot to fit a bivariate spatial distribution for theft and drug and plot them on the street map of San Francisco. Obviously, theft is scattered across a wide region while drug is relatively in a small area.

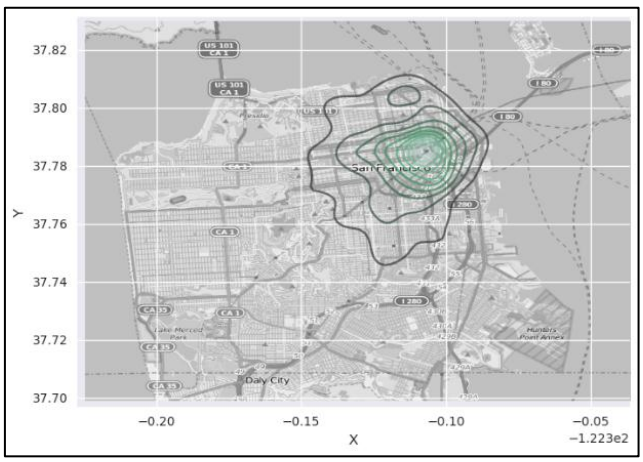

Figure 7 spatial distribution of theft

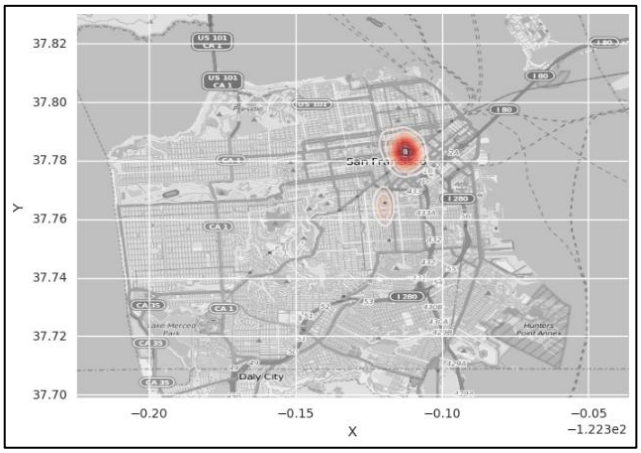

Figure 8 spatial distribution of drug

3. Predictive Task

3.1 Evaluation

Since the task is to predict the category of crime given time and location, we adopt the multi-class logarithm loss used by Kaggle to evaluate prediction. In specific, for each record we calculate a set of predicted probabilities denoting the record belongs to certain categories.

Suppose there are N cases in testing set and M categories of crime. Let y_{ij} be 1 if observation i is in category j and p_{ij} be the predicted probability of observation i belongs to category j, then loss function is:

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

To avoid the extremes of the log function, predicted probabilities are rounded as in Kaggle system:

$$\text{output} = \max(\min(p, 1-10^{-15}), 10^{-15})$$

3.2 Baseline

We used a naïve constant classifier as the base line. It always predicts the category as theft, which is the most common crime category. Its logarithm loss is 26.916843.

4. FEATURES

4.1 Time related features

There are two fields in dataset related to time, Dates and DayOfWeek. From these two fields, we extracted year, month, day of week and hour as our features. Instead of using the numerical values directly, one-hot encoding vectors are used to represent all these features. For instance, Tuesday is [0 1 0 0 0 0 0] and November is [0 0 0 0 0 0 0 0 0 0 1 0]. We then concatenate all these vectors to get a complete time vector.

We also evaluated models with numerical values as features. Unsurprisingly, one-hot encoding representation outperforms numerical value version.

4.2 Geographic features

PdDistrict: we use one-hot encoding representation to represent the 10 police department districts.

XY: We use Google s2sphere package to convert XY pair into cell id. Cells here serve as a hierarchical decomposition of the sphere into compact representations of regions or points.

There are four reasons why we use cells as features. First, they are compact. Second, they have resolution for geographical features. Third, they are hierarchical (they have levels, and similar levels have similar areas). And last, the containment query for arbitrary regions is fast. After converting to cell id, we also use one-hot representation to convert it into a 72-d vector.

Address: We compute the crime distribution over 39 crime categories at each address to form a 39-dimension vector, with each value representing the number of crimes of a specific category which happened at the address. For address only appearing in testing set, this vector is set to be the mean vector of all addresses.

5. MODELS

5.1 Naïve Bayes Classifier

Naive Bayes algorithm is a classification technique based on Bayes' Theorem with an assumption that each feature is independent with each other. It's a conditional probability model that computes the probability for each of possible labels of output given each instance features.

For our crime classification problem, we didn't use one-hot representation since there may be correlation between entries in one-hot represented feature vector. Instead, we just use their numerical values and for string type features like PdDistrict and Address, we map each string value into an integer value. For instance, for PdDistrict field, we map BAYVIEW to 0, CENTRAL to 1 and so on.

5.2 Logistic Regression Classifier

Logistic regression is a regression model where the output is a binary variable. The logistic function is given below:

$$\text{f(x)} = \frac{1}{1+e^{\theta x}}$$

f(x) represents the probability of the output y

equaling to one.

For our crime classification problem, it is a multi-class problem. We use the one-vs-rest (OvR) scheme to train our model. One-vs-Rest is an estimator. It uses a basic classifier, and create 2-class problems for k classes respectively. The 2-class classifier for class i predicts whether it belongs to i or not. Finally, by evaluating on k 2-class classifiers, we take the result of the classifier which has the highest confidence level.

5.3 Random Forest Classifier

Random forest is an ensemble learning method for classification. It constructs many decision trees when training and choose the result class which is the mean prediction or the mode of the classes of all trees. Compared with Decision tree algorithm, Random Forest could avoid the overfitting problem which is common in Decision tree.

The training algorithm for random forests applies bagging technique to tree learners. When splitting at each node, the tree learning algorithm selects a random subset of the features, which is called feature bagging. The reason of this is that if we don't randomly choose the features, those features which are strong predictors will be selected in most trees, which result in a high correlation between trees.

6. EXPERIMENT RESULTS

We trained all the three models discussed in the previous section with different parameter settings and feature combinations. Although time related features and geographic features are both important as we can see in the data exploration section, the one-hot encodings of time-related features are relatively more straight-forward than that of geographic ones, so we mainly focused on evaluating geographic features and include all the time-related features. For reference, the best log loss of the original test data on the leaderboard is about 1.96 and the median is about 2.57.

6.1 Naive Bayes

Firstly, we trained a Naive Bayes model, which is the simplest model among the three. As explained above, we use numerical representation of features instead of one-hot encoding for this model, because intuitively a one-hot represented feature is strongly correlated between the binary values. Considering that Naïve Bayes model assumes features are independent but our geographic features: PdDistrict, Cellid and Address are somewhat more correlated with one another than with other features, we evaluated this model with the same time related features discussed above, but different geographic feature sets:

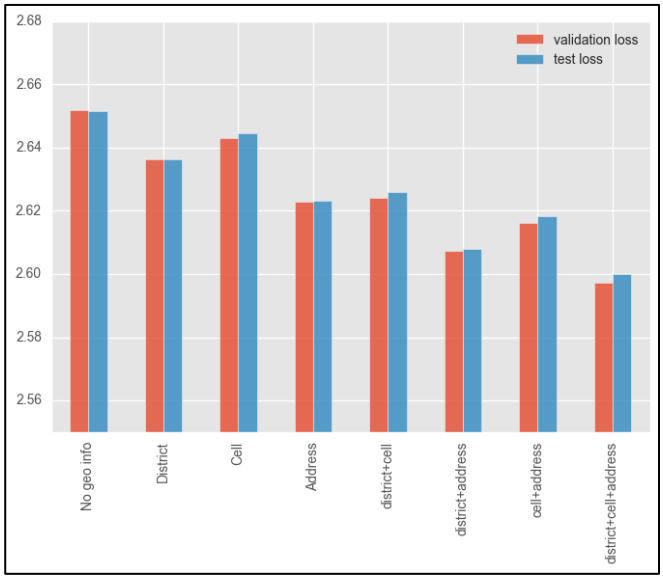

Figure 9 Log loss of different feature sets

The result from Figure 9 gives us some hints that maybe we should include all the three geographic features in the following models since they all contribute to better performance to some extent although they obviously have stronger correlation.

6.2 Logistic Regression Classifier

For logistic regression classifier, we started to use one-hot representation for all the features. We trained and evaluated this model over several different feature sets, with regulation parameter C set to 1.0.

Logistic Regression Classifier	Validation loss	Test loss
Numerical representation of all features	2.637831	2.637972
All time features	2.561897	2.562949
All time features + district	2.561306	2.562913
All time features + district + cell id	2.520856	2.523257
All time features + district + cell id + address	2.492194	2.493280

Table 1 Log loss of different feature sets

As shown in Table 1, one-hot representation is much better than numerical representation, and as we add more geographic features, the performance get better.

6.3 Random Forest Classifier

Random Forest Classifier generates the best results for our classification task with on-hot representation of features.

Random Forest Classifier has two important hyper parameters: number of trees and max depth of trees. We first use cross validation on max depth and found that 25 is good enough. Then we mainly focused on tuning the number trees. Generally, as we can see in Figure X, both validation and test performance are getting better as the number of trees increases, but gets very slightly after 300.

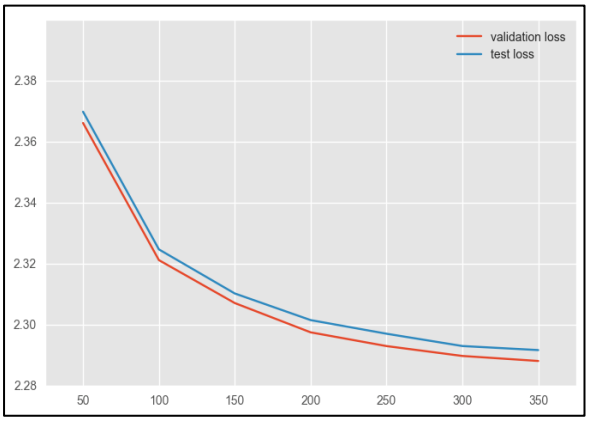

Figure 10 Validation and test loss against #trees

We also evaluated this model on different feature sets, as we can see in Figure X. To our surprise, The Address feature which is encoded by the frequency of crimes at an address acts as a much more important role in reducing the log loss in this model than in the other two. We think the relatively small granularity of address compared to PdDistrict and Cell id makes it capable of providing more precise geographic information in classification. And the categorized encoding of crime frequency by address extracts some prior information between location and crimes.

Random forest Classifier	Validation loss	Test loss
Numerical representation of all features	2.561307	2.562913
All time features	2.487060	2.488371
All time features + district	2.478570	2.480188
All time features + district + cell id	2.464559	2.473776
All time features + district + cell id + address	2.289851	2.293105

Table 2 Log loss of different feature sets

From the confusion matrix heatmap of Figure 11, we see that the predictor predicts LARCENY/THEFT (no. 16) most often, which corresponds to what we discovered in the data exploration section

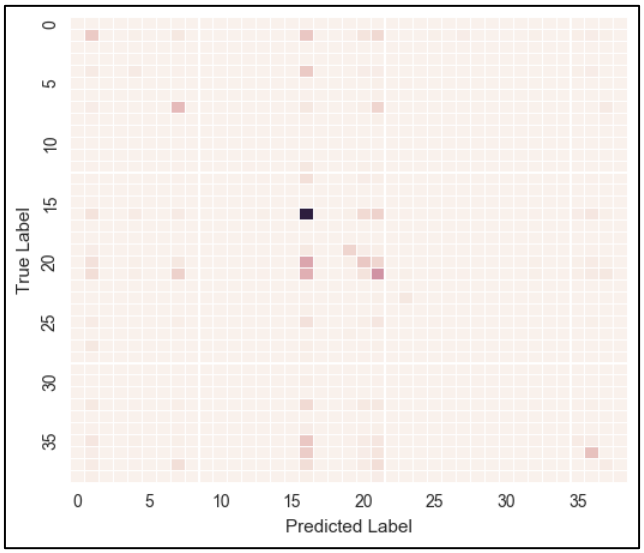

Figure 11 confusion matrix

7. RELATED WORK

7.1 Predict Hotspot

Many people focus on predicting the hotspots of crime. An approach for detecting the crime hot spots by structured classification method is presented in VijayaKuma's paper[1]. It differs from traditional spatial classification as it focused on preprocessing the crime events before mapping. Tahani implements the apriori algorithm on location and time features and excluded the crime type feature to come up with a list of all crime hotspots along with its related frequency[2].

7.2 Comparison of Models

There are also work on comparison of performance between models on crime classification problems.

Iqbal has made a comparison between Naive Bayesian and Decision Tree algorithms performed over the crime dataset[3]. He achieved 70.81% and 83.95% accuracy for Naive Bayesian and Decision Tree method respectively, which indicates Decision Tree performs much well on crime classification problem. They have made experiments on three different models Naive Bayes, KNN and Gradient Tree Boosting[4]. They found Gradient Tree Boosting achieves the best performance compared to the other two models.

7.3 Data Augmentation

The original dataset only includes limited information about geography and time. There's a useful method to improve the accuracy of predicting crime, which is data augmentation. It is nice to take into account

information from other datasets like traffic flow, population density, weather information, and so on to help predict crime.

Cohn points out that there's a significant positive relationship between the number of days with a maximum temperature equal to or less than 32°F ('Days Cold') and robbery rates[5] ,which suggested robbery crimes are more possible to happen in cold weather. Heller and Markland found that the number of hours of daylight was a significant predictor of calls for police service in three United States cities (Chicago, Detroit, and St Louis) [6]. Feldman and Jarmon (1979) found a negative correlation between precipitation and assaults[7].

7.4 Ensemble Learning

Ensemble learning is a very powerful technique to increase accuracy on a variety of ML tasks. It combines many classifiers by averaging or voting. It is said that a lower correlation between ensemble model members tends to result in an increase in the error-correcting capability[8].

8. CONCLUSION

We thought this is a simple classification task, but in practice, the size of dataset has been a challenge, not to mention finding feature from limited information and fine-tuning classifiers.

Dataset exploration and feature selection are largely in parallel. Time related information are easy to extract; more attention has been paid to geographical information. We tried a series of method to get a more meaningful representation than original latitude and longitude, and finally used Google's s2 and split whole city into 70+ cells. At first we guess information in PdDistrict and address may overlap with geolocation, but they improves performance greatly. A possible explanation is that they reflect some "social" adjacency.

Model training and evaluation goes more smoothly. Random forest outperforms other classifiers just as we expected. Due to machine capacity, we were unable to try some time or memory consuming models, like SVM and gradient boosting tree.

Although there are many unexpected problem, our group enjoyed working on it. It's a pity that the Kaggle competition has already end and we can't know our model's behavior on private leader board.

9. REFERENCES

[1] M. VijayaKumar, P.Balamurugan, Basim Alhadidi, and Hanumanthappa. "Crime Classification Algorithm for Mining Crime Hot Spot and Cold Spot". International Journal of Computing Academic Research (IJCAR). ISSN 2305-9184 Volume 3, Number 2(April 2014), pp. 58-69.

[2] Almanie, Tahani, Rsha Mirza, and Elizabeth Lor. "Crime Prediction Based On Crime Types And Using Spatial And Temporal Criminal Hotspots." arXiv preprint arXiv:1508.02050 (2015).

[3] Iqbal, Rizwan, et al. "An experimental study of classification algorithms for crime prediction." Indian Journal of Science and Technology 6.3 (2013): 4219-4225.

[4] Junbo Ke, Xinyue Li, and Jiajia Chen. "San Francisco Crime Classification." University of California San Diego (2015).

[5] Cohn, Ellen G. "Weather and crime." *British journal of criminology* 30.1 (1990): 51-64.

[6] HELLER, N. B., and MARKLAND, R. E. (1970), 'A Climatological Model for Forecasting the Demand for Police Service', Journal of Research in Crime and Delinquency, 7: 16.

[7] FELDMAN, H. S., and JARMON, R. G. (1979), 'Factors Influencing Criminal Behavior in Newark:

A Local Study in Forensic Psychiatry', Journal of Forensic Science, 24: 234—9.

[8] http://mlwave.com/kaggle-ensembling-guide/