Group Info

Name	CS logins
Qian Mei	qmei
Huan Lin	hl18
Han Sha	hsha
Xiaocheng Wang	xwang28

Abstract

From 1934 to 1963, San Francisco was infamous for housing some of the world's most notorious criminals. Despite the fact that nowadays the city by the bay is known more for its tech scene than its criminal past, there is still no scarcity of crime due to the increasing wealth inequality and housing shortages.

Therefore, based on nearly 12 years of crime reports from across all of San Francisco's neighborhoods, providing time and location, our project is aimed to explore the dataset visually so as to discover the overall trends and distribution using d3 and MATLAB. Furthermore, based on the insights gained from data visualization, we build a predictive model for crime classification by applying machine learning techniques on various combinations of attributes. As for real-world application purpose, our prediction results may be referred by local police department in San Francisco so that they can assign workload of police main power accordingly, quickly and more efficiently given the crime data.

Data

The data we are using in our project comes from Kaggle Data Science Competition Platform(https://www.kaggle.com/c/sf-crime/data). This dataset contains criminal incidents derived from San Francisco Police Department Crime Incident Reporting system, ranging from 1/1/2003 to 5/13/2015 with the following dimensions of attributes:

- *Dates* – timestamp of the crime incident, format YYYY-MM-DD HH:MM:SS
- *Category* – category of the crime incident,totally 39 categories
- *Descript* – detailed description of the crime incident
- *DayOfWeek* – the day of the week, 7 values
- *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 of crime scene
- *Y* – Latitude of crime scene

Since the data is formatted well, we didn't perform the further data cleaning because it may result in missing useful information. However, due to the fact that the label attribute "Category" is excluded in test data, it's not applicable since we are evaluating our machine learning model based on accuracy. Thus in our exploration, we focused on information in training data, which contains totally **878050** crime data entries represented as (Dates, Descript, DayofWeek, PdDistrict, Resolution, Address, X, Y). Therefore, preparing for feature extraction and machine learning, we have to split the provided "train.csv" into new training data and test data. Our strategy is such that half for training, and half for testing.

Hypothesis

- We assume the total crime counts are decreasing from 2003 to 2015.
- We assume police Department District(PdDistrict) is most indicative and correlated to the crime classification.
- We assume more crimes would take place in weekdays compared to weekends.
- We assume southern is the most dangerous district in San Francisco.

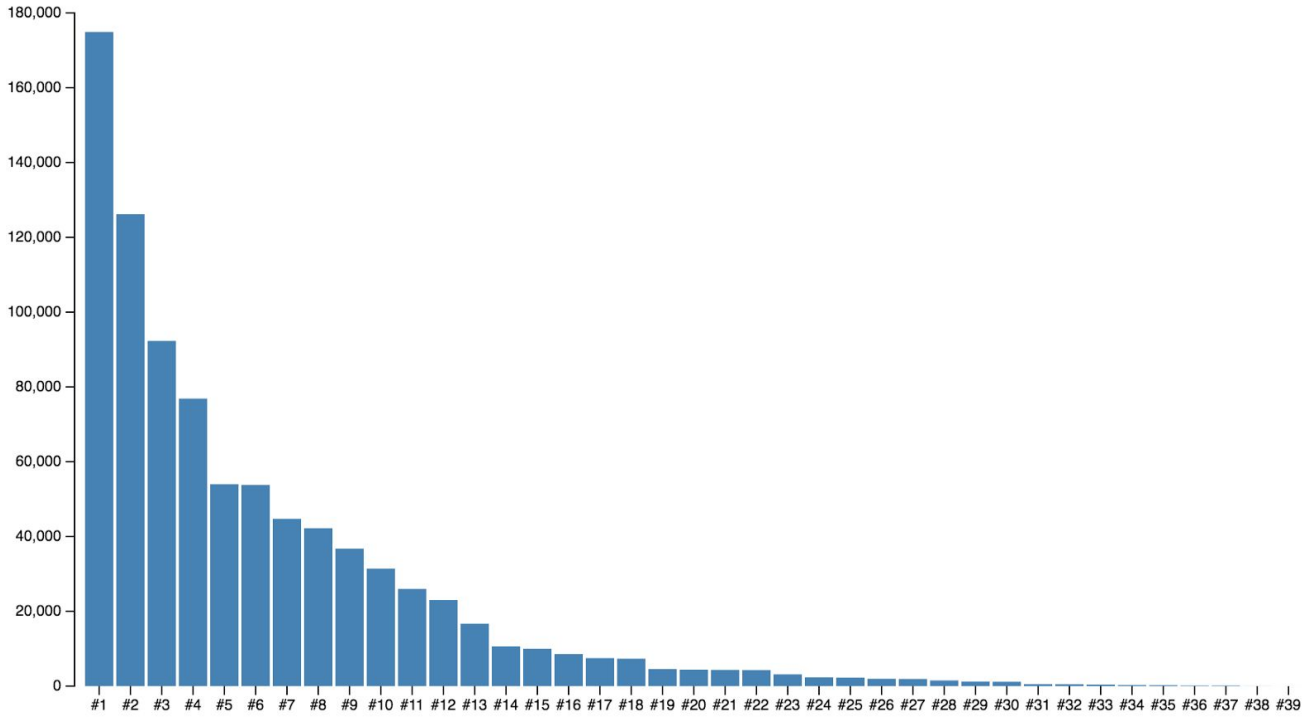
Methodology

- **Data Visualization:**
  In order to solve problems such as, ***what is the crime category distribution and pattern in the city by the bay, what are the most relevant features for predicting crime categories***, we start with exploring the data visually to see if any hints and insights can be obtained. Our visualization are mainly delivered by D3 and MATLAB.
- **Machine Learning:**
  Besides, we also do some statistics analysis for the purpose of better understanding of the overall dataset. Then, based on the hypothesis listed above, we take the following steps to build and select predictive model:
    1. Split data into train set and test set. Learn a model on training data and test on unseen test data.
    2. Extract attributes to be used as input for machine learning algorithm.
    3. Use one-hot encoding to prepare real training features so as to avoid that any learner based on standard distance metrics (such as k-nearest neighbors) between samples will get confused.
    4. Try different classification techniques, tune parameters, use cross validation to avoid overfitting problem.
    5. Compare and analyze performance on different classifiers.

Data Visualization, Trends & Patterns

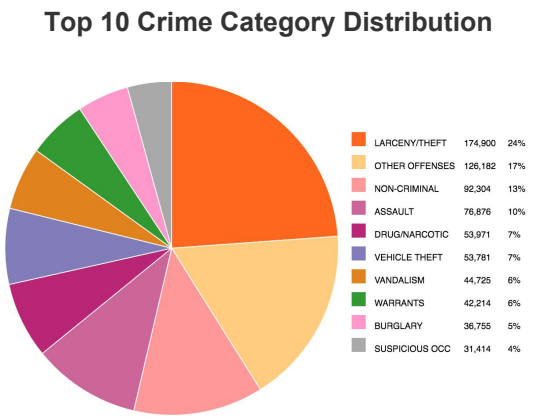
To begin with, we'd like to know the distribution of the crime category and answer the question: **What's the most common crimes**? There are totally 39 crime categories, and they are listed as followed(*ordered by the total crime counts from highest to lowest*)

#1 Larceny/Theft
#2 Other Offenses
#3 Non-Criminal
#4 Assault
#5 Drug/Narcotic
#6 Vehicle Theft
#7 Vandalism
#8 Warrants
#9 Burglary
#10 Suspicious Occurrences
#11 Missing Person
#12 Robbery
#13 Fraud
#14 Forgery/Counterfeiting
#15 Secondary Codes
16 Weapon Laws
#17 Prostitution
#18 Trespass
#19 Stolen Property
#20 Sex Offenses Forcible
#21 Disorderly Conduct
#22 Drunkenness
#23 Recovered Vehicle
#24 Kidnapping
#25 Driving under the influence
#26 Runaway
#27 Liquor Laws
#28 Arson
#29 Loitering
#30 Embezzlement
#31 Suicide
#32 Family Offense
#33 Bad checks
#34 Bribery
#35 Extortion
#36 Sex Offense non forcible
#37 Gambling
#38 Pornorgraphy/Obscene Mat
#39 Trea



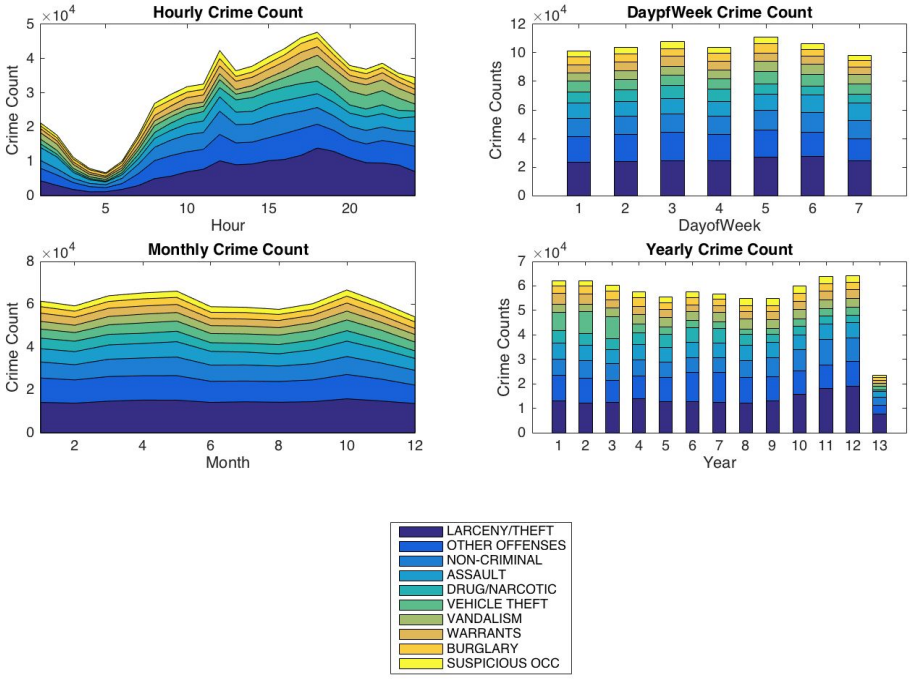
Total Crime Counts of 39 Crime Categories from 2003-2015

We can see that that total crime count throughout the 12 years of each crime type differs significantly, suggesting categories are unevenly distributed.



Additionally, statistics show that nearly ¼ of the total crimes belong to “Larceny/Theft”. Moreover, top five crime types accounts for 60 percentage, and top 20 classes cover 97% of the entire dataset. In the light of this statement, we realize that training instances for #21-#39 crime categories are too few. Therefore, the predictions of this part of data might affect the overall performance of the model negatively considering the fact that we have to build a 39-class classifier. Besides, we believe theft is the most dominant crime type, which thus deserves further exploration.

After mastering the overall distribution of the target label “Category”, we move on to probe into the hidden timely pattern and geographical patterns regarding the proportion of the data belong to top 10 crime categories, for the simple reason that the data provides information regarding two aspects: time and location.



In order to test the assumption that crimes have timely pattern, we derived information about hour, dayofweek, month and year from two attributes “Dates” and “DayofWeek” and deliver the viz above, which turns out that there are indeed some interesting patterns:

Weekly Crime Count

- The distribution of “dayofweek” vs “crime counts” is comparatively even. No extreme value appears in a specific day, with the highest crime counts 133734 in Friday and the lowest counts 116707 on Sunday. And it seems that our third hypothesis is incorrect.

Hourly Crime Count:

- 5am is the most peaceful time period, with the lowest crime incidents.
- Crimes are more likely to take place after 12pm than before 12pm.
- There are three peek-hour for committing crimes: mid-night at around 12pm, noon at around 12pm and afternoon period “17:00-18:00”.

Monthly Crime Count:

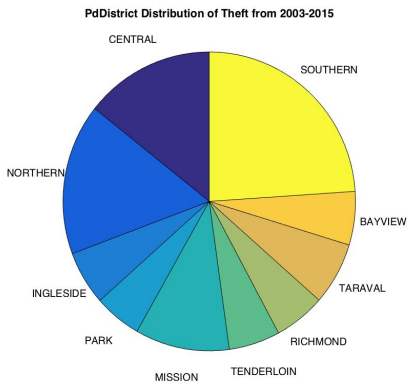
- We can see from the stacked area chart, there are 2 peaks of crimes through the year: May and October.

Yearly Crime Count:

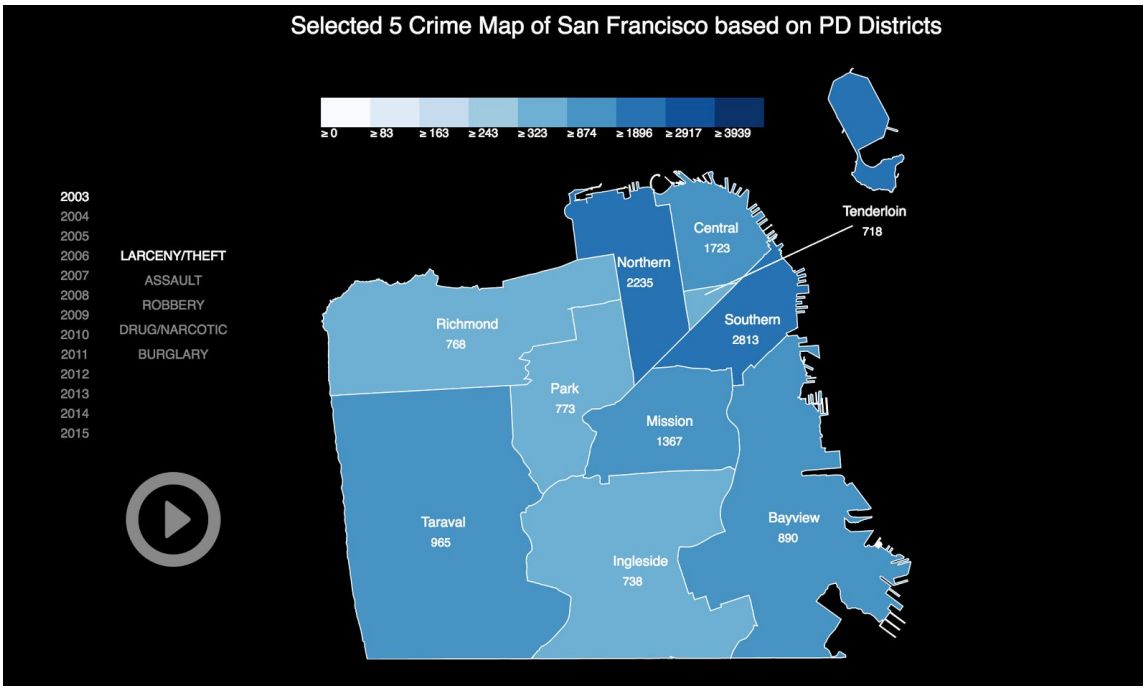
- The 13th year, which corresponds to 2015, has far less crime than the years before. It’s not because the police department has reinforced intensive fights against criminal so that the crime rate has reduced significantly. But, the underlying reason is that we only have the crime data before 05/13/2015.

- More importantly, San Francisco witnessed increasing crime rate since 2010. Compared to 2010, the crime rate increased **16.5%** until 2014. And when we investigate into **theft which tops among all the 39 categories**, it’s obvious that theft increases sharply since 2009, giving rise to the total high crime rate over the year. And the vast majority of theft belongs to property crimes: **automobile break-ins, pickpocket/pursesnatch and shoplifting**.

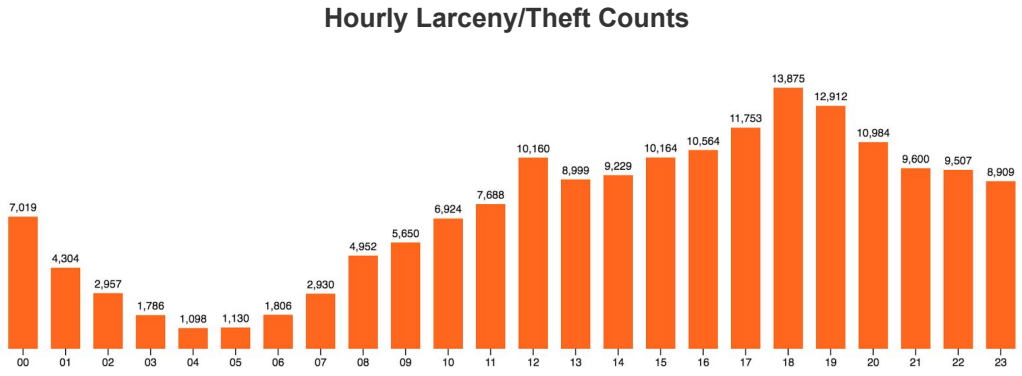
- After thorough analysis, we consider two possible causes of great significance:
  (1) Electronic devices like smartphones, tablets are increasingly prevalent, and these items seem to be the easiest and most lucrative target for thieves.
  (2) Another possible reason may be that, the victims or witnesses of crimes are more willing to report happened crime to police, meaning that the crime wave is actually an increase in crimes being reported instead of crimes happened.



Statistics also show that, **southern has the highest property theft crime**, reaching 41845 over the 12 years' period of time. And this picture shows the yearly theft counts in 2003(The full animation can be accessed by this link: https://embed.plnkr.co/GrPNfG7qQuwZwYKLr1D0/)



And also from the pie chart above, we discovered the fact is such that ***nearly a quarter 's crime occurs in the densely populated, transit-rich Southern Police Station district***, which runs from The Embarcadero to south of Market Street. Besides, we can also find some evidence if we look back to the hourly theft count(refer to the bar chart below). The peek is 18:00 ~ 19:00, turning out to be the rush hours in the daytime when people are busy commuting.



Machine Learning

Our project endures a touch experience when we attempt to build effective and comparatively “accurate” predivtive model. We mainly go through the four phases below and both negative and postive results we’ve gained from our trials are recorded as below.

Negative Results

- **”Roberry/Non-Roberry” Classifier**
  Our first trial is doing binary classification instead of multi-classifications. Our assumption is quite simple, we are using use (DayOfWeek, PDDistrict) pair as features to predict the crime category. And We choose the category ‘ROBBERY’, use the classification labels are 0/1 array indicating whether a crime belongs to this category.

  Applying selected two features, we’ve tried three algorithms: Naive Bayes, Logistic Regression and SVM. We use first half of training data as training, and second half of train set as testing for the simple reason that the test.csv does not have a label for category, thus we can’t use it to compute the accuracy. All of these three algorithms outputs the same training accuracy and validation accuracy, with training accuracy 0.974156 and validation accuracy 0.973455

  Why would this happen? let’s take a look at confusion matrix: [[427371, 0] [11654, 0]] As we can see, the classifier simply labels everything as non-ROBBERY. The reason behind is that, we only have 2.69% (23000/855049=0.026899) of all crimes that belong to ROBBERY. Thus we can conclude that (DayOfWeek, PDDistrict) pair is not a good predictor for crime categories.
- **Triple-classification**
  In the next phase, we decided to adjust the learning goal of our machine learning algorithm. Instead of simple binary classifier’s outputting whether a crime is ROBBERY or NON_ROBBERY, we adopt multi-class classification, which classifies the crime as LARCENY/THEFT, NON-CRIMINAL or ASSAULT. In this attempt, we are still using the same (DayOfWeek, PDDistrict) as the feature set. The resulting accuracy is around 0.22 for all Naive Bayes, Logistic Regression and SVM.

  This is clear not an accurate classifier. We doubt the problem may be that we are using too few features giving rise to underfitting. Thus possibly we can try more attributes as learning features or try other categories as the learning goal such as regional data, which can be useful for the police department to focus more on a certain region than the other, as crimes are committed more in some places.

Positive Results

- **“LARCENY/THEFT / NON-LARCENY/THEFT” Classifier**
  Based on the category distribution we’ve visualized earlier, we targeted crime category of LARCENY/THEFT, attempting to build a LARCENY/THEFT / NON-LARCENY/THEFT classifier because such category has totally 174900 incidents in training data, thus we probably can get a more accurate classifier due to more information hidden in more data.

  Also, we made some adjustment of the features we fed into the model. When performing feature extraction, we found that the geographical coordinates are not useful as all the crimes occurred in San Francisco. In other words, the longitude and the latitude are nearly the same. In the light of the above statement, it’s really hard to tell the difference based on the longitude and the latitude. However, it seems to make more sense if we are using PdDistrict to tell more information of geographical location of crime zones.

  In addition, we added a new feature indicating whether it’s daytime or night. From the visualization, we found that the crime is to some extent related to time. Specifically, 05:00:00 – 05:59:59 is the most peaceful time period. So we converted “Date” field into 4 discrete values indicating the time period when the crime occurred. Thus, we had three features, Time Period, Day of Week and PDDistrict.

  Moreover, in order to balance the data from both classes which aims to prevent the classifier from overfitting, we only used part of NON LARCENY/THEFT data. By adding the above intuitions into the model-building process, the resulting accuracy of predicting whether a crime belongs to “LARCENY/THEFT” is around 60%. And the performance statistics is as followed,

Binary Classification Performance on LARCENY/THEFT

	Precision	Recall	F1-score
NON LARCENY/THEFT	0.62	0.67	0.64
LARCENY/THEFT	0.58	0.52	0.55

  From the statistical data above, we can see that it is much better than previous attempts, 22% accuracy. As type I errors and type II errors for both classes are of no big difference, the classifier predicts well for both classes. Considering that there exist a certain amount of data which belong to different classes with same features, the result is quite good. For the next week, we plan to optimize our features to improve the performance of our classifier, otherwise we can’t make progress only with current features.

- **Ultimate Solution**

  Since the dataset is relatively large (over 800,000 data entries), before applying any learning algorithms, we preprocess and filter the input file first. The basic idea is to generate a series of subsets of training and testing pairs. Furthermore, as we discussed earlier, since crime distributions are uneven across different categories, especially the unpopular ones such as 'GAMBLING' and 'BRIBERY', which may introduce outliers and decrease the accuracy, we decide to filter out those unpopular instances. The output of this preprocess is a series of training and testing pairs for top 20, top 10, top 5, top 3 categories, and of course, the original unfiltered training and testing pairs.

  After preprocessing, the next step is feature extraction. The raw input such as the date (2003-01-06) is split into 3 separate features (year, month and day). Resolution feature is left out, since there are fair amount of entries with NULL value. The address of individual crime is left out, since there is generally one crime per address. The coordinate (x, y) is also left out for the same reason. However, we do keep PdDistrict as an important geo-related feature, as it is much easier to represent. The resulting feature space we are going to use is [year, month, day, hour, DayOfWeek, PdDistrict].

  Another important optimization we utilize is '***One-Hot-Encoding***'. The reason behind is that, for the features, such as DaysOfWeek, are ordinal labels. It makes no sense to represent, for example, Fridays (labeled as 4) as twice as large as Wednesdays (labeled as 2). The better representation would be a 7-bit length vector that represents Fridays as [0000100] (4th bit on), and Wednesdays as [0010000] (2nd bit on), and so on.

  Finally the dataset is ready to use. Thanks to sklearn python packages, we are able to implement various machine learning algorithms easily, from the basic ones like Logistic Regression Classifier, to the complicated ones like Random Forest. Here is a list of classifiers we have tried.

	LARCENY/THEFT/NON-LARCENY/THEFT
SVM (Linear SVC)	0.7973
Logistic Regression	0.7974
BernouliNB	0.7973
Adaboost	0.7728
Random Forest	0.7972
Bagging	0.7974
Gradient Boosting	0.7972

Last but not least, we revisit our failed binary classification we discussed earlier on, using the models we constructed above. This time we try to classify the most popular categories. Here is the result.

	TOP 3 CLASSES	TOP 5 CLASSES	TOP 10 CLASSES	TOP 20 CLASSES	All 39 CLASSES
SVM	0.2907	0.2907	0.1969	0.1415	0.1477
LogReg	0.3329	0.3329	0.2389	0.2043	0.1998
BernouliNB	0.3476	0.3476	0.2487	0.2117	0.2079
Adaboost	0.3767	0.3767	0.2662	0.2138	0.1825
Random Forest	0.2877	0.2877	0.1727	0.1559	0.1309
Bagging	0.4690	0.3412	0.2368	0.2026	0.1931
Gradient Boosting	0.4880	0.3760	0.2701	0.2316	0.2233

Analysis and Future Directions

Although it's difficult to classify most of examples correctly when doing multi-class classification, our result is better than that of random guessing as we extract several useful features like year, month, day and PdDistrict, and ignore meaningless features like X, Y. We employed several machine learning algorithms discussed in class such as SVM, Naive Bayes, Logistic Regression and so on. The best result that we can reach when classifying top 3 classes is around 0.48 using gradient boosting. On average, we can just reach around 0.35.

We also tried to optimize on feature extractions. But the accuracy seemed to encounter bottlenecks. The reason why the accuracy of multi-class classification can't be improved further is that, the classifiers have high biases because algorithms just learn simple models from relatively insufficient features.

In addition to that, from the datasets, we also found that there are quite a few examples which have the same feature combinations but belong to different categories, which leads to the bottleneck of the accuracy of multi-class classification. Also, looking into the weight factor of the classifiers, the weight of each feature is not big enough and this indicates that our classifiers may underfit on data. One of the solutions to underfitting problem is to add more features more related to categories such as the criminal environment, weather or temperature.

Moreover, with the number of categories increasing, the accuracy of classifiers decrease a lot. We deem there are two main reasons:

1) features are not enough.
2) numbers of examples of each category are of big difference.

The first part has been been covered and discussed above. As for the second part, we contribute it to the categories distribution. As discussed earlier, training instances for #21~#39 crime types are too few compared to other types, which exerts negative effect on the accuracy due to the noise added to the data. Lack of data is the most severe problem when classifying multi classes. This leads to the result that the classifier will classify an unseen example as the category of more examples rather than that of less examples. In order to classify better, the data should be evenly distributed among all categories.