San Francisco Crime Classification

2015 Fall CSE 255 Assignment 2 Report

Shen Ting Ang
A53095324
s3ang@eng.ucsd.edu

Weichen Wang
A53089102
wew129@eng.ucsd.edu

Silvia Chyou
A53101184
schyou@ucsd.edu

ABSTRACT

We aim to classify the type of crimes committed within San Francisco, given the time and location of a criminal occurrence. This study is important and beneficial. Using data mining approaches, we can predict the location, type and time of criminal occurrences in the city. We also explore some interesting questions, for example, if more crimes occur on certain days of the week or certain times of the day.

1. 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 [3]. 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 [12].

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.

2. EXPLORATORY ANALYSIS

Our dataset is the San Francisco Crime Data from 2003 to 2015, from [7]. This dataset was originally from SF OpenData [11], San Francisco Government's Open Data platform.

2.1 Summary of the Dataset

The dataset includes data from 6 Jan 2003 to 13 May 2015 inclusive, with a total of 878,049 data points. This works out to an average of 195 incidents per day over 4510 days. The dataset appears to have been into alternating weeks, i.e. the training set from Kaggle contains odd weeks while the unseen test set contains even weeks. The data is in a CSV file, each data point represented as a row with the following 9 columns:

1. Date - timestamp of the crime incident
2. Category - category of the crime incident (what we will predict)
3. Descript - description of the incident
4. DayOfWeek - day of the week of the incident
5. PdDistrict - Police Department District which the incident occured
6. Resolution - how the incident was resolved
7. Address - approximate street address of the incident
8. X - Longitude
9. Y - Latitude

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
Total	**878,050**

Table 1: Number of Crimes for Each Police Department District

The data set is ordered by timestamp, with the most recent entries (i.e. 13 May 2015) at the top of the CSV file. As a guideline, the official population of San Francisco was 776,733 in 2000 and 805,235 in 2010, which represented a 4% increase.

While the dataset is generally clean, an issue was discovered with the Latitude and Longitude coordinates - there were a few hundred entries with Longitude and Latitude given as -120.5 and 90 respectively. As the street addresses were insufficient for us to correct these entries and the number of these entries were small (representing less than 0.5% of the dataset), we decided to remove them from the dataset.

For the purposes of our analysis and prediction, the description of the incident and the resolution are both not useful - the description is merely a more verbose description of the incident, while the resolution gives the outcome of the incident. The street address of the incident is better described by the longitude and latitude, and is also not very useful.

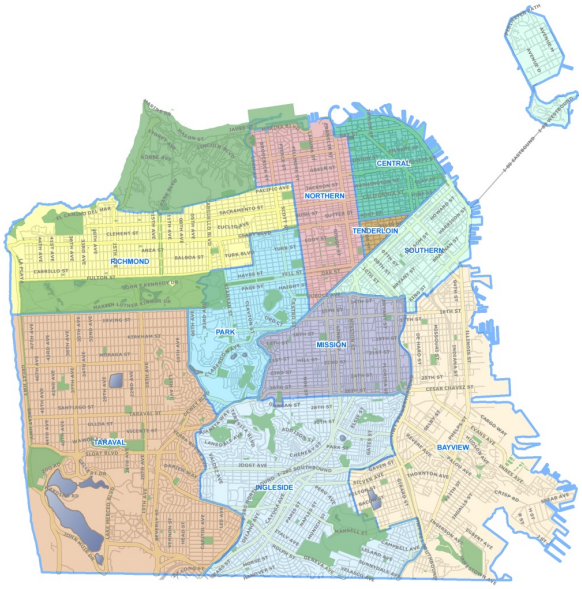

Figure 1: Map of the Police Districts

Resolution	Number of Crimes
NONE	526,790
ARREST, BOOKED	206,403
ARREST, CITED	77,004
LOCATED	17,101
PSYCHOPATHIC CASE	14,534
UNFOUNDED	9,585
JUVENILE BOOKED	5,564
COMPLAINANT REFUSES TO PROSECUTE	3,976
DISTRICT ATTORNEY REFUSES TO PROSECUTE	3,934
NOT PROSECUTED	3,714
JUVENILE CITED	3,332
PROSECUTED BY OUTSIDE AGENCY	2,504
EXCEPTIONAL CLEARANCE	1,530
JUVENILE ADMONISHED	1,455
JUVENILE DIVERTED	355
CLEARED-CONTACT JUVENILE FOR MORE INFO	217
PROSECUTED FOR LESSER OFFENSE	51

Table 2: Number of Crimes for Each Resolution

A more coarse categorization of the incident location is given in the Police Department District - there are 10 of these and the breakdown of the number of crimes is given in **Table 1**:

A map of the police districts is shown in **Figure 1**.

While the resolution of the incidents is not part of our main analysis, it is interesting to see how the cases were dealt with **Table 2**:

Note that in the majority of the cases, there was no action taken.

2.2 Characteristics of the Dataset

Figure 2 is a pie chart that demontrates the total number of crime of each categories. There are 39 categories of crime in total, and we only displayed the top ten of them. The most common seen crime is LARCENY/THEFT.

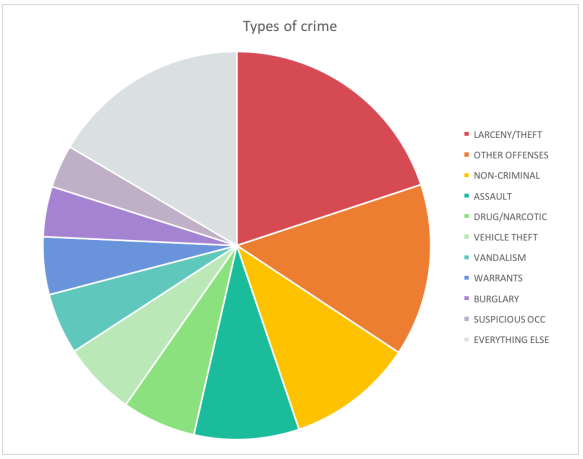

Figure 2: Types of Crime

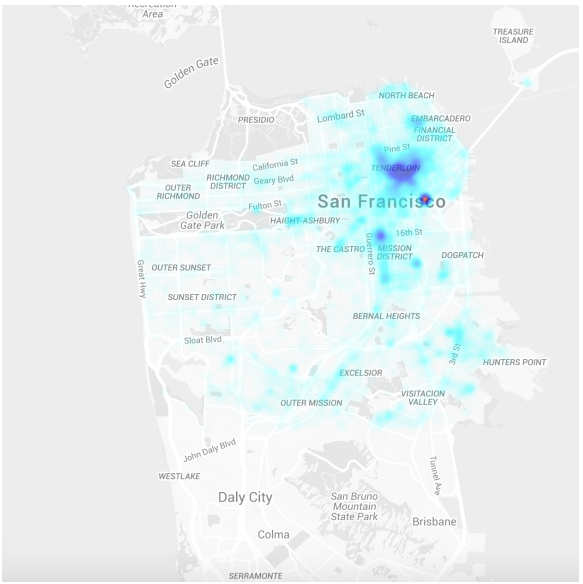

Figure 3: Heat Map

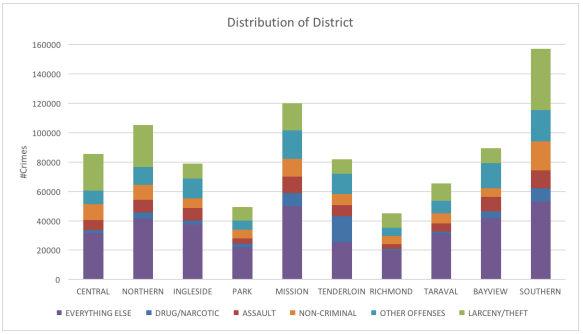

Figure 4: Distribution of District

In addtion, **Figure 3** is a heat map of the criminal occurences in San Francisco. We parsed the data and utilized Google Map API, to gain a better insight about how the crime are distributed. With the heatmap result, we understand that the crimal occurences are highly related to the

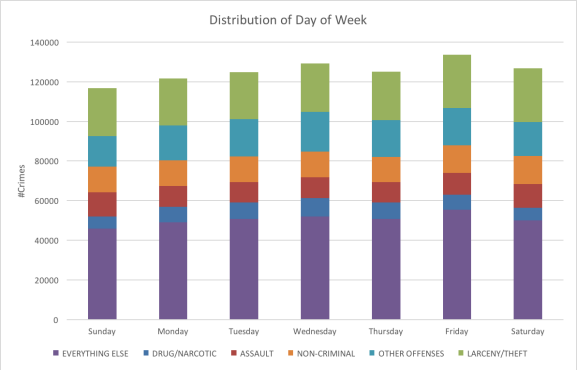

Figure 5: Distribution of Day of Week

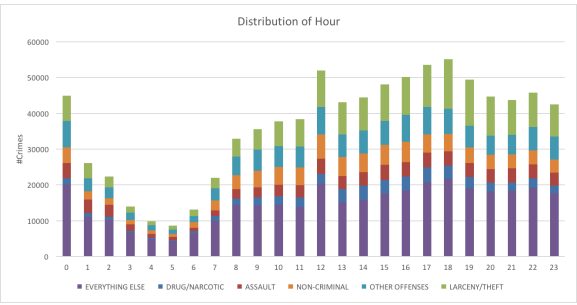

Figure 6: Distribution of Hour

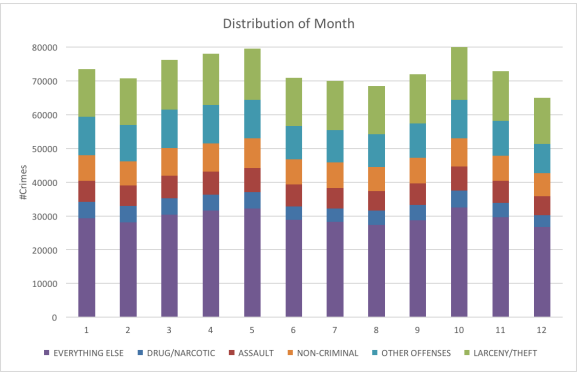

Figure 7: Distribution of Month

location, and that we should make good use of all the locational features we have.

Moreover, **Figure 4** is a stacked bar chart of the number of crime each pd district. Different color in a bar represents different category. Most of the criminal incidents took place in SOUTHERN and least in RICHMOND.

We would like to further explore other columns of our dataset to help us extract useful features. What are the distributions for day of week, hour, month, and even year for the crimes record? In **Figure 5**, we can see the distribution of day of week, the highest criminal occurrence was on Fridays and lowest was on Sundays. The result is not too surprising as what we believe is that since Friday is the day before the weekend, people tend to go out for dinner or do something special and have fun. As a result, since everyone is going out, there should be a higher chance to encounter

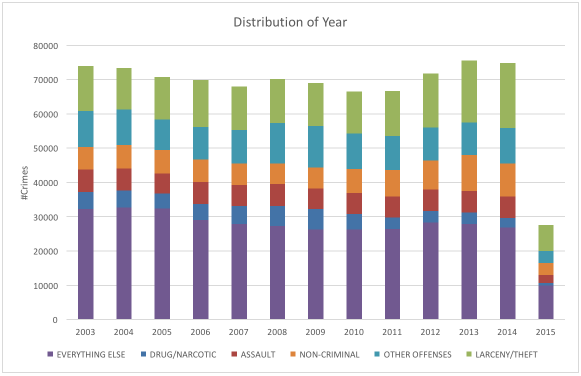

Figure 8: Distribution of Year

a criminal event. Whereas for Sundays, since it is the last day of the weekend, people tend to stay at home and thus the criminal occurrences should be lower.

What about hour distribution? In **Figure 6**, we can see that the highest criminal occurrence was at 18 o'clock and the lowest was at 5 am. This result is not too surprising either. Since people usually get out of work at around 5 to 6 pm, 6 pm seems like a reasonable and likely time to have the highest criminal occurrence. The more people outside, the higher the chance to have a criminal event. It is worth to notice that 12 pm is also another time that has high crime occurrence because it is the lunch time. Moreover, since 4 to 5 am is the sleep time, thus it is reasonable to observe a lowest criminal occurrence at 5 am.

In addition, let us explore the month distribution from **Figure 7**. We observe that there is not much difference among 12 months, the variation is low. The highest criminal occurrence was in October whereas the lowest criminal occurrence was in December.

For year distribution in **Figure 8**, since the dataset only covered till May 2015, the total number of crimes for 2015 is not the completed result. The variation among years is low as well. We can see that the number of crimes does increase for 2013 and 2014.

3. PREDICTING CRIMINAL OCCURRENCES

A predictive task using this data set is to predict the category of crime given the day and location. This is the predictive task given in the Kaggle competition, and we have decided to attempt this task.

3.1 Preprocessing

For the purposes of our analysis, we split the dataset given by Kaggle into three parts for training, validation and testing with proportion 60%, 20% and 20% respectively. We used Scikit-learn's `train_test_split` function for this, setting a specific random seed to ensure reproducibility across different runs.

To avoid errors, we first convert the separators of the CSV file in LibreOffice to be the carat symbol (^) instead of commas to avoid issues with commas appearing in the description or address columns. The use of the carat symbol as a separator is common practice. especially in text mining applications.

The dataset was stored as a Pandas dataframe, and the first preprocessing step was to remove entries with erro-

neous Latitude and Longitude coordinates as described in the previous section. The category of crime was encoded using Scikit-learn's LabelEncoder function, and the categorical features such as hour of the day, day of the week, district, month of the year were encoded using Pandas' `get_dummies` function.

3.2 Features

From the dataset, the columns that are of interest are the timestamp, the day of the week, the police district and the latitude and longitude coordinates. From the timestamp, we can extract features such as the month of the year and the hour of the day. Along with the day of the week, these should be treated as categorical features. Similarly, the police district is also treated as a categorical feature. These are represented as binary variables corresponding to the different values each feature can take.

An interesting question arises as to how the Latitude and Longitude features should be used. Arguably, the police district is a function of these features, but with just 10 police districts, the exact coordinates are a much richer source of location data. We decided to break the entire San Francisco area into a grid and encode the location information from the Latitude and Longitude into which particular grid square where a crime occurred. We choose to initialise this as 64 squares by dividing the range of Latitude and Longitude into 8 respectively.

3.3 Possible Approaches

This is a classification problem which can be addressed using supervised learning methods such as Logistic Regression, Naive Bayes or Support Vector Machines. Ensemble methods such as Random Forest or Gradient Boosting are also possible algorithms which can be used for this problem.

3.4 Baseline

A baseline model would be to use Logistic Regression, using the day of the week and police district as features. This uses information directly from the dataset with an easy to understand algorithm.

3.5 Evaluation

The evaluation metric used by Kaggle is the Multi-class Log Loss. "The metric is negative the log likelihood of the model that says each test observation is chosen independently from a distribution that places the submitted probability mass on the corresponding class, for each observation" [9].

A simple metric would be to measure the classification accuracy by comparing the most probable class to the actual class. This is a metric used often in evaluation of classification algorithms [16, 15].

Another metric used in classification tasks is the multi-class confusion matrix. This would allow to see common misclassifications between specific classes, and to calculate precision and recall.

4. MODELING AND RESULTS

Our model is shown in **Table 3**.

4.1 Baseline Model

We first consider a baseline model, using just the Police District and day of the week as model features. The baseline algorithm as described earlier will be logistic regression. As a basis for comparison, we also use Naive Bayes and Random Forest (with 150 trees and a maximum depth of 20). We use the scikit-learn implementation of the algorithms, using the default settings for Naive Bayes, and Logistic Regression with C=0.01. The result is shown in **Table 4**.

As a reference, a log-loss score of 2.62 would be slightly below the median score on the Kaggle leaderboard.

Feature	Type	Size
District	Categorical	10
Day of the Week	Categorical	7
Hour of the Day	Categorical	24
Month of the Year	Categorical	12
Lat/Long	Categorical	16/24/32

Table 3: Model Features

Features	Classifier	Log-loss		Accuracy	
		Valid	Test	Valid	Test
District+Day	Logistic Regression	2.62120	2.62123	0.22130	0.22031
District+Day	Naive Bayes	2.61369	2.61435	0.22120	0.22006
District+Day	Random Forest (150,20)	2.61887	2.61971	0.22130	0.22031

Table 4: Log-loss and accuracy of baseline model with Logistic Regression, Naive Bayes and Random Forest

Features	Classifier	Log-loss		Accuracy	
		Valid	Test	Valid	Test
District+Day+Hour	Naive Bayes	2.58148	2.58253	0.22452	0.22241
District+Day+Hour	Logistic Regression	2.59149	2.59157	0.22433	0.22208
District+Day+Hour	Random Forest (150,20)	2.58382	2.58410	0.22581	0.22468
District+Day+Month	Naive Bayes	2.61366	2.61391	0.22157	0.22011
District+Day+Month	Logistic Regression	2.62024	2.61999	0.22149	0.22040
District+Day+Month	Random Forest (150,20)	2.63293	2.63254	0.22072	0.22040
District+Day+Hour+Month	Naive Bayes	2.58149	2.58211	0.22499	0.22256
District+Day+Hour+Month	Logistic Regression	2.59058	2.59038	0.22460	0.22253
District+Day+Hour+Month	Random Forest (150,20)	2.58756	2.58864	0.22478	0.22307

Table 5: Log-loss and accuracy of Naive Bayes, Logistic Regression and Random Forest on various combination of time features

4.2 Additional Time Features

From our initial data exploration, there seems to be differences in crime occurrences during certain months or certain hours of the day. It thus makes sense to add in features encoding the hour of the day and the month of the year. The result is shown in **Table 5**.

The month features on their own actually results in worse results in both the validation and testing set, while the hour features improves the log-loss by 0.02. It is interesting to note that adding in the month features only improves the log loss slightly, but not very surprising as there is a more significant difference in various hours of the day. From the validation results, it makes sense to use both the hour and month features together.

4.3 Location Features

On top of the Police District, we can use the Longitude and Latitude features. A simple method that we chose to use is to break the area into a square grid of 8 squares in each direction, forming 64 squares. We encode this into 16

Features	Classifier	Valid Log-loss	Valid Accuracy
District+Day+Hour+Month+Grid(8)	Naive Bayes	2.67998	0.21067
District+Day+Hour+Month+Grid(8)	Logistic Regression	2.56860	0.22824
District+Day+Hour+Month+Grid(8)	Random Forest (150,20)	2.53794	0.23558

Table 6: Log-loss and accuracy of Naive Bayes, Logistic Regression and Random Forest with the addition of location grid features (size 8) on the validation set

Features	Classifier	Valid Log-loss	Valid Accuracy
Day+Hour+Month+Grid(8)	Naive Bayes	2.60558	0.20922
Day+Hour+Month+Grid(8)	Logistic Regression	2.59161	0.21681
Day+Hour+Month+Grid(8)	Random Forest (150,20)	2.56035	0.22374

Table 7: Log-loss and accuracy of Naive Bayes, Logistic Regression and Random Forest with the location grid features (size 8) instead of Police District on the validation set

Features	Classifier	Valid Log-loss	Valid Accuracy
District+Day+Hour+Month+Grid(12)	Naive Bayes	2.66196	0.21024
District+Day+Hour+Month+Grid(12)	Logistic Regression	2.56334	0.22840
District+Day+Hour+Month+Grid(12)	Random Forest (150,20)	2.51824	0.23860
District+Day+Hour+Month+Grid(16)	Naive Bayes	2.65437	0.21680
District+Day+Hour+Month+Grid(16)	Logistic Regression	2.55166	0.23367
District+Day+Hour+Month+Grid(16)	Random Forest (150,20)	2.50024	0.24814
District+Day+Hour+Month+Grid(20)	Naive Bayes	2.66695	0.21763
District+Day+Hour+Month+Grid(20)	Logistic Regression	2.55197	0.23573
District+Day+Hour+Month+Grid(20)	Random Forest (150,20)	2.49647	0.24962

Table 8: Validation Log-loss and Accuracy for various sizes of location grid features on Naive Bayes, Logistic Regression and Random Forest

Features	Classifier	Valid Log-loss	Valid Accuracy
District+Day+Hour+Month+Grid(20)	Random Forest (150,10)	2.55806	0.23521
District+Day+Hour+Month+Grid(20)	Random Forest (150,15)	2.51978	0.24120
District+Day+Hour+Month+Grid(20)	Random Forest (150,20)	2.49647	0.24962
District+Day+Hour+Month+Grid(20)	Random Forest (150,25)	2.50689	0.24907
District+Day+Hour+Month+Grid(20)	Random Forest (200,25)	2.49546	0.24957
District+Day+Hour+Month+Grid(20)	Random Forest (250,25)	2.49606	0.24934

Table 9: Validation Log-loss and Accuracy for various settings of n_estimators and max_depth for Random Forest

categorical features, 8 each representing the X and Y grid of a particular criminal occurrence. The result is shown in **Table 6**.

Naive Bayes performed poorly, likely due to the fact that the grid features are not independent of the district. Removing the district features improves the log-loss for Naive Bayes, but the log-loss and accuracy both are still worse than the model using the district. The result is shown in **Table 7**.

Random Forest seems to outperform the other two methods using the models with these additional features. There are two different sets of parameters we can tune, the first would be the grid size of our location features, while we cover the parameter tuning of Random Forest in the next subsection. Also, we include the district features in our model using Random Forest. The result is shown in **Table 8**.

There is some evidence of overfitting with a grid of size 20 with regards to logistic regression. The improvement for Random Forest is also slowing down.

4.4 Optimization

We try various values for the parameters of the Random Forest classifier. There is a huge effect on varying the maximum tree depth, with evidence of slight overfitting when this exceeds 20. The number of estimators used was also varied, with 200 estimators giving the best results both in terms of log-loss and classification accuracy. The result is shown in **Table 9**.

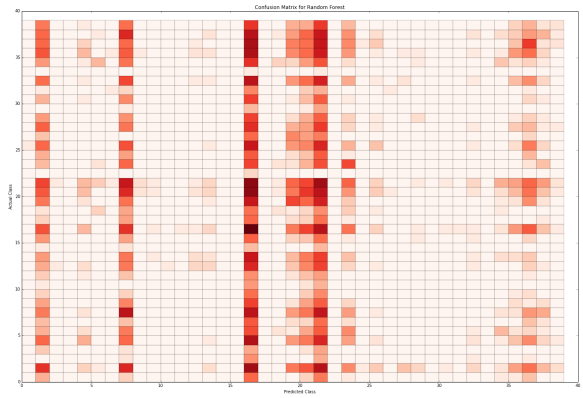

Figure 9: Confusion Matrix (Log-normalized) of our proposed model on the test set

4.5 Other models

We considered using Support Vector Machines, but decided not to use them due to two reasons: (i) the data is non-linear and performs badly using a linear Support Vector Classifier, and (ii) the Scikit-learn implementation of SVC is documented as performing badly when the number of data points exceeds 10,000. In practice, this was the case as training was not completed even after half an hour. In comparison, the biggest Random Forest model used above finished training in approximately 10 minutes or less.

Gradient Boosting is considered "state-of-the-art" for classification problems. However, it is more computationally expensive compared to Random Forest and some effort is also required to tune the parameters - which is difficult to fit in the limited timeframe of this project. We will consider using this in future submissions for the actual Kaggle competition.

4.6 Proposed Model

We propose using all the features - District, Day, Hour, Month and Grid (size 20) with a Random Forest with 200 estimators and maximum depth 20. Log-loss and Accuracy on the test dataset was 2.49745 and 0.24863 respectively.

The log-loss on the Kaggle test set, using the model on the entire dataset was 2.52142.

4.7 Discussion

While it is possible to interpret Random Forest models [4], in practice, most of the time, Random Forests are treated as a "black-box" model. We did not attempt to interpret the Random Forest model in our analysis.

The advantage of the Random Forest model is its ability to make use of the additional location features, whereas the Logistic Regression model started overfitting at a smaller number of features. The Naive Bayes model ran into issues when we used both the district and actual coordinates due to issues with these features not being independent of each other.

It is worth noting that in all the models, the classification accuracy was low, less than 25% even in our best model. This is likely due to the fact that the dataset is skewed, and the effects are visible in the confusion matrix (ref to confu-

sion matrix). There are several vertical bands in the confusion matrix, e.g. class 16 corresponds to Larceny/Theft, which is the most common type of crime. This is caused by the models "overfitting" to more commonly seen labels.

5. RELATED WORKS

5.1 Data Augmentation

The dataset does not include any other information pertaining to the geography or demographics of the city. It is possible to use information from other datasets that would be useful for predicting crime, e.g. using power consumption, public transit travel data, business information as proxies for population density during various times of the day.

Another approach would be to use weather data to augment the dataset for prediction. In [1], the author Lam used San Francisco Police Report data from Kaggle and used WeatherData libraary to retrieve San Francisco weather from December 13, 2014 to May 13, 2015. Lam merged weather data with crime data to find the correlation between weather condition and the type of crime. Lam concluded that criminals are most active at cooler temperature than hot temperature. The next step of Lam's research is to create a predictive model to predict the type of crime based on the temperature and weather condition of the day.

5.2 Neural Networks

Another algorithm which could be use for this prediction task would be neural networks. Neural networks have gained popularity in the past decade and have been used for various classification tasks such as handwritten digit recognition [17], image recognition [18] and speech recognition [16]. In many applications, neural networks have displaced the previous state-of-the-art methods as they deliver better classification accuracy.

There are several drawbacks to using neural networks though: (i) neural networks are in general a black box; there is no easy way to interpret the model weights, (ii) the hyperparameter space for neural networks is large, from the number of nodes to the type of activation units and the learning rate of the backpropagation steps (iii) neural networks are computationally expensive, perhaps even more so than gradient boosting trees, and large neural networks often require the use of optimized code and GPUs to train.

One of the published solutions on Kaggle [2] uses the Keras [8] package in Python to implement a neural network classifier. It is worth noting that Keras, which is built on top of Theano [13], has greatly reduced the complexity of coding a neural network model, but work is still needed to select an optimal set of hyperparameters. The performance using a simple neural network is listed as being around 2.55 in terms of log-loss in the comments, which is slightly worse than our final model. However, the published model did not make use of optimizations such as Contrastive-Divergence pre-training nor did the model use as rich a set of features as we did; neural networks tend to perform better with a larger feature space.

5.3 K-Nearest-Neighbour

K-nearest neighbor (KNN) can be used as an algorithm on classification. An object is classified by a majority vote of its k nearest neighbors [5]. KNN algorithm can be used in this task to classify the data, with the distance of neighbors being determined by the categorical features as described above.

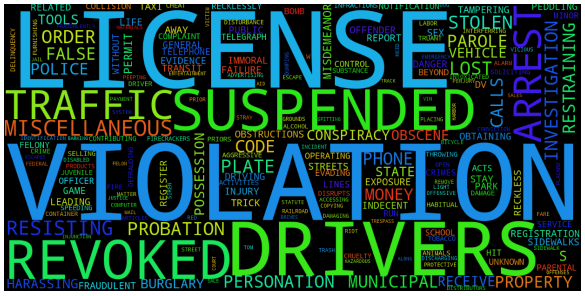

Figure 10: Word Cloud of Other Offenses

A participant shared his approach [6] on Kaggle using `KNeighborsClassifier` from sklearn library in Python, with a high log-loss of above 25. A possible reason is that KNN suffers from poor performance whenever the class distribution (in this case, the category of crime) is skewed [14]. The "majority voting" method will raise problems when there are huge classes who dominate the predictions, and there will be a propensity for new data to be voted into more popular classes. Unfortunately, the dataset is highly skewed - the most popular class occurs in one-fifth of the dataset, while the least popular class only appears 6 times, which is less than 0.001% of the entire dataset . Therefore it seems inappropriate to use KNN in this task. Another possible reason of the poor performance of this participant's result may be the features he chose as only the X, Y (latitude, longitude) coordinates were used as model features.

5.4 Interesting Findings

An interesting observation presented by another participant on Kaggle [10] is that the data in the category "other offences" are mostly about traffic violations. He parsed the data whose category is "other offences" and utilized their description to build a word cloud. The cloud showed several traffic-related words ('license', 'drivers', 'traffic'), indicated that the crime falls in the category of other offenses are mainly traffic violations. The figure is shown in **Figure 10**.

6. CONCLUSIONS

The problem of predicting crime seems like a simple application of classification algorithms, however the exploration of the data unearthed some interesting trends such as the relationship between crime and the hour of the day. As crime is a social issue, there exists opportunities to augment the data using other existing data sets such as weather data, as we have found in our literature review. While the dataset gives us the police districts, we have attempted to improve on this by using the X and Y coordinates as a grid to create finer location features. Our final model represents our efforts to create a model which makes use of both the time and location features present in the original dataset, as well as using the Random Forest which is easy to understand and yet "complex" enough to make use of the larger feature set in our model. This model also runs relatively quickly on the large dataset.

Our group has enjoyed working on this dataset and will be exploring some of the ideas discussed for our future Kaggle submissions. It will be interesting to see how the Kaggle

prize-winning models will influence future police work, and whether this will have the effect of helping to lower the crime rate in San Francisco.

7. REFERENCES

[1] Correlation between weather condition and the type of crime. https://nycdatascience.com/correlation-between-weather-condition-and-the-type-of-crime/.

[2] Fighting crime with keras. https://www.kaggle.com/smerity/sf-crime/fighting-crime-with-keras.

[3] History of san francisco. https://en.wikipedia.org/wiki/History_of_San_Francisco.

[4] Interpreting random forests. http://blog.datadive.net/interpreting-random-forests/.

[5] k-nearest neighbors algorithm. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.

[6] K-nearest-neighbour. https://www.kaggle.com/wawanco/sf-crime/k-nearest-neighbour.

[7] Kaggle: San francisco crime classification. https://www.kaggle.com/c/sf-crime/data.

[8] Keras documentation. http://keras.io/.

[9] Multi class log loss. https://www.kaggle.com/wiki/MultiClassLogLoss.

[10] "other offenses" = traffic violations. https://www.kaggle.com/gantell/sf-crime/other-offenses-traffic-violationss.

[11] Sf opendata. https://data.sfgov.org/.

[12] The tech industry is stripping san francisco of its culture, and your city cloud be next. http://www.newsweek.com/san-francisco-tech-industry-gentrification-documentary-378628.

[13] Theano. http://deeplearning.net/software/theano/.

[14] D. Coomans and D. L. Massart. Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta*, 136:15–27, 1982.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[16] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[17] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.