San Francisco Crime Classification

Abstract

The purpose of this project was to develop a classifier for the categories of crimes occurring in San Francisco. The problem was proposed by Kaggle, as one of their machine learning competitions.

Given data relating to previous crimes in the city, a Naive Bayes algorithm was applied to a set of carefully selected features to generate predictions. Time and place proved critical to the success of the crime category prediction, therefore the majority of research involved investigating which combination of spatial and temporal features produced highest increases in performance.

The resulting prediction achieved a logloss value of 2.61253, which was a 28.39% decrease on the baseline value. This result was not impressive in terms of the Kaggle leaderboard, however there are several ways in which it could be improved - such as expanding the types of data used to build the model to incorporate a wider variety of potential influences, or to develop a model which combines several different algorithms to utilise advantages of other methods.

Introduction

San Francisco, like many other large urban cities, is subject to high levels of crime and illegal behaviour. With it's infamous history of criminal activity, and the fact that it has now evolved to become the technological capital of the world, it is no surprise that the city has publicly released hundreds of thousands of records with information relating to the crime that has occurred over the past 12 years, as part of a machine learning competition.

Hosted by Kaggle, the competition provides a dataset of information about each incident, and the aim is to predict the category of crime that occurred. Solving this problem would allow authorities to have a better idea of what types of crime are likely to occur when and where, and would allow for better resource allocation around the city.

Background

To solve this problem, a variety of spatial and temporal features have been extracted and adapted from the given dataset, and have been utilised by a range of machine learning algorithms to generate the highest accuracy of predictions.

The method of evaluation used for this competition is multi-class logarithmic loss. This value is calculated as follows:

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

where N is the number of test instances, M is the number of class labels, y_{ij} equals 1 if instance i belongs to class j, and p_{ij} equals the predicted probability of instance i belonging to class j. The perfect solution would result in a logloss of 0, with decreases in performance leading to increases in logloss value.

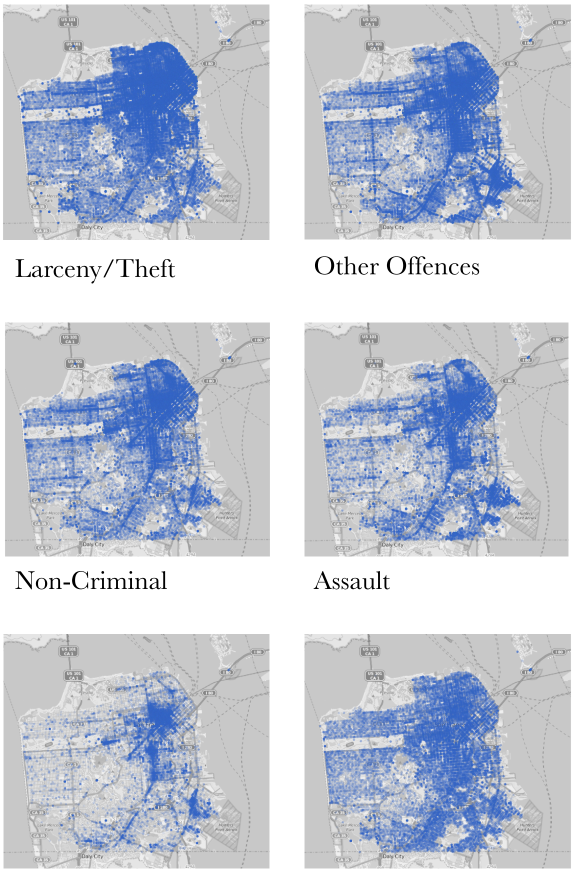
Methods

Upon initial analysis of the training dataset, there seemed to be two key areas which could contribute to the classification of the crime - time and space.

Included in the given features were the latitude and longitude of the location of the crime, along with a rough street address, and the Police Department district. Also provided was a date/time variable, and which day of the week the crime occurred.

When looking at the locational data of each crime type, it is clear to see that there are hotspots around the city - however many of these high-density areas do not have significant levels of differentiation between the categories of crime. The six most frequent crimes are mapped individually in Figure 1, and together in Figure 2 to visualise this.

Figure 1



Larceny/Theft

Other Offences

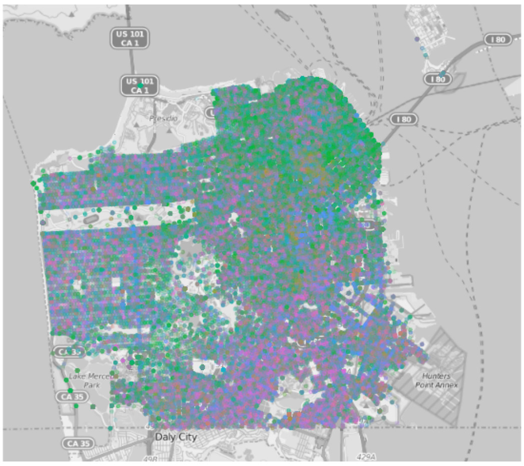
Non-Criminal

Assault

Drug/Narcotic

Vehicle Theft

Figure 2



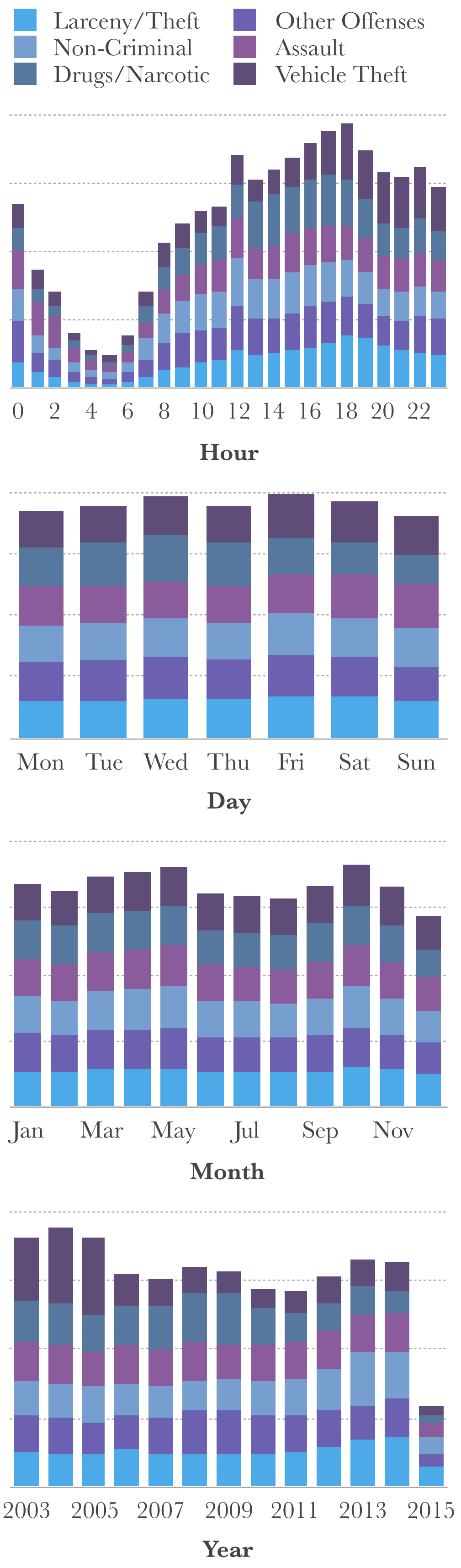
Top 6 Crimes

From these maps it was determined that the latitude and longitude values did not provide enough value to be included as features themselves, but other features could be developed from them in order to better categorise the locations.

In terms of the temporal features provided, the date/time value required some preprocessing before it could be analysed. This involved splitting up the feature and discretising the relevant parts. To begin with, the hour, month, and year of the crime were extracted as individual features. Along with the day of the week, these are represented in Figure 3 which shows the percentage of each type of crime that occurred across these different time periods (hour, day, month, year).

The charts show significant correlation between these features and the category of crime - for example the clear decline in drug/narcotic and vehicle theft in recent

Figure 3



years, or the sharp increase in larceny/theft and non-criminal activities that occur at 12pm.

Before starting the experimental stages of the project, a small amount of data needed to be preprocessed to remove anomalous results. For example, there were a minority of instances in which the longitude was given as 90.0 (which is the North Pole). In cases like this with clear errors or outliers, the instances were removed from the training data, and for the test data these instances had anomalies replaced with the mean values generated in training.

Experiments

To set a baseline for the problem, each probability was set to simply the general probability distribution of that crime type for every instance. This resulted in a logloss score of 3.64827.

The first form of testing used a very simple set of features - one spatial (the Police Department district) and one temporal (the day of the week). The intention behind this was to try a variety of machine learning algorithms and compare their success rates on a basic model, before extending the features to include those that were most relevant to the algorithm chosen. The algorithms considered were Naive Bayes, k-Nearest-Neighbour, Multi-Layer Perceptrons, Random Forests, and Support Vector Machines.

The fact that the goal of this competition was to generate predicted probabilities for each class ruled out the use of SVM's as this algorithm would need to be heavily adapted to provide this.

k-NN seemed like it could be a good algorithm due to the spatial features provided, however because of the size of the datasets it would require very large values of k, and this resulted in unfeasible running times given the scale of the project. Similarly, the Multi-Layer Perceptron algorithm took too long when training models to be considered applicable. k-NN would have also incurred problems caused by the skewed dataset. In the training data, the most common category represents almost 20% of the set, whereas the least common category appears in less than 0.00001% of the set - this would cause larger classes to dominate the classification.

When comparing Naive Bayes and Random Forests on the basic model, they performed equally well in terms of accuracy, but Naive Bayes had some improvement on logloss, and hugely outperformed Random Forests in terms of the time taken to build the model. The Random Forests was run with 100 trees, and both models used five-fold cross validation. The results are shown below in Figure 4.

Figure 4

	Naive Bayes	Random Forests
LogLoss	2.65261	2.66028
Accuracy	22.07%	22.07%
Build Time	0.16s	46.9s

After the decision to use Naive Bayes as the algorithm going forward with this project, the next step was to add more features. The first additions to the feature set were the year, month, and hour extracted from the date/time variable. From the hour value, a further feature was created which split the 24 hours into 4 time-slots - morning, afternoon, evening, and night. Similarly, another feature was created from the day of the week value, which was a binary feature representing whether the crime occurred on a weekday or not. This new model of 7 features (hour, time-slot, month, year, day of week, weekday, police department district) was trained with ten-fold cross validation using Naive Bayes, and improved the accuracy to 22.33%, and reduced the logloss to 2.62901.

However, the Naive Bayes algorithm operates on the assumption that all features are conditionally independent, and this feature set included two pairs of features which could be seen as conditionally dependant - the day of week and weekday, and the hour and time-slot. For this reason further testing was performed to see if the removal of one of each pair would improve the predictions. The best combination turned out to be using just the time-slot and the binary weekday features, and removing the hour and day of week features. This improved the accuracy to 22.55% but logloss remained almost the same.

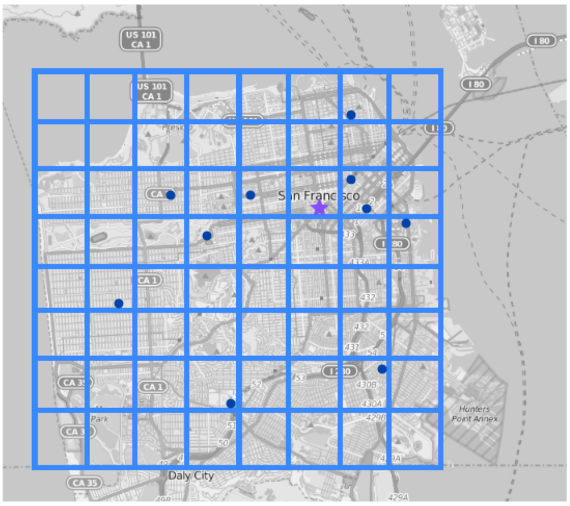
An additional comparison was made regarding the type of attribute used in the model for certain features. With month and hour, there were two options for how they were treated as attributes - either as numeric values or as nominal. In the previous test they had been to set to numeric, because the Naive Bayes model treats numeric values in such a way that closeness in value represents closeness with the instances, which seems intuitive in the context of time (with the exception of January not seeming close to December etc).

When switching the type to nominal attributes with a range of {01, 02, 03, … } etc., the result was a decline in performance as expected, so the attributes remained numeric in the model.

The next experiment was to attempt to improve upon the location data. So far, only the Police Department district was used, but three additional features were considered.

Firstly, a feature which represents the distance of the crime from the city centre of San Francisco. Secondly, a feature which represents the distance of the crime from the Police Department station of the district. Finally, a feature which split the city of San Francisco into an 8x8 grid, and assigned the crime location to one of 64 cells. Figure 5 visualises these concepts.

Figure 5



New location features

After incorporating each feature in turn to the feature set, the resulting predictions did not improve upon the previous values, as shown in Figure 6.

Figure 6

	Accuracy	LogLoss
City centre	21.93%	2.69312
PD station	9.54%	3.86642
Grid cell	22.71%	2.66186

Reasoning for this poor performance was determined to be the fact that these features each violate the Naive Bayes' assumption of conditional independence in some way. For example, the distance to the PD station varies greatly, but is related to the existing PD district feature because the size of the district also varies, so a small distance to the station often correlates with a small district. This would affect the city centre feature in a similar way.

The best of the three new features was the grid cell (which actually improved accuracy, but not logloss), so to check whether the conditional independence was the cause of a negative performance, an additional test was implemented where the Police Department district was replaced with the grid cell feature, so that only one locational feature existed in the feature set. This resulted in an improvement, with accuracy at 22.52% and a new low for the logloss value, at 2.61253.

One final test was performed to check whether binarising the other temporal features (year and month in addition to weekday which was already a binary variable) would yield a better result, however it was marginally worse than the current prediction with 22.59% accuracy but 2.61775 for the logloss.

The final feature set of the classifier used was as follows:

- Year
- Month
- Time-slot
- Weekday
- Grid cell
- Category

Conclusion

The final score of the classifier built for this competition was a logloss of 2.61253. This is a significant improvement on the baseline, but still reasonably far away from the current leading score of 2.05079. The progress of the classifier is plotted below in Figure 7, showing the changes in result as the feature set was extended, reduced, and adapted.

Figure 7

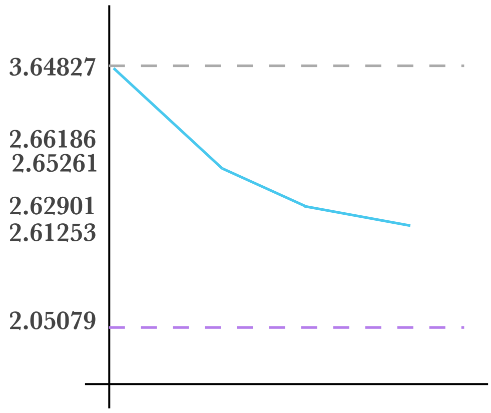


Figure 8 shows the ablative analysis of the final feature set. From this it can be seen that the grid and time-slot features accounted most for the decrease in logloss. But it is worth noting that the order of deconstruction would have an affect on how much each component contributes.

Figure 8

Component	Accuracy	LogLoss
All	22.52%	2.61253
-Month	22.50%	2.61998
-Year	22.14%	2.62385
-Timeslot	21.79%%	2.64001
-Grid	19.91%	3.64827

One way in which this project could be expanded and likely improved is with the addition of more data. Firstly, the test dataset could have included some of the extra features which appeared only in the training set. In particular, experimentation with using resolution as a feature for the training set demonstrated improvements on the validation sets. Using the final model as an example, inclusion of resolution as a feature when performing ten-fold cross validation increased accuracy to 33.91% and brought logloss down to 2.2415.

Additional data that would have been interesting to examine would be data relevant to the city of San Francisco - for example demographic data such as the average ages, incomes, unemployment rates and poverty levels for specific areas. Another area of relevance could be the weather, with temperatures and rainfall etc. perhaps having an influence, or using a calendar of events that occurred around the city to see whether that has any correlation to crime levels.

A further way to improve upon the predictions and therefore the logloss value would be to try a combination of machine learning algorithms - some form of ensemble method, using bagging, or boosting.