# STATS 503: Categorizing San Francisco Crime

Ed Wu (jameswu), Zack Keller (zkeller), John Keane (keanery)

April 13, 2016

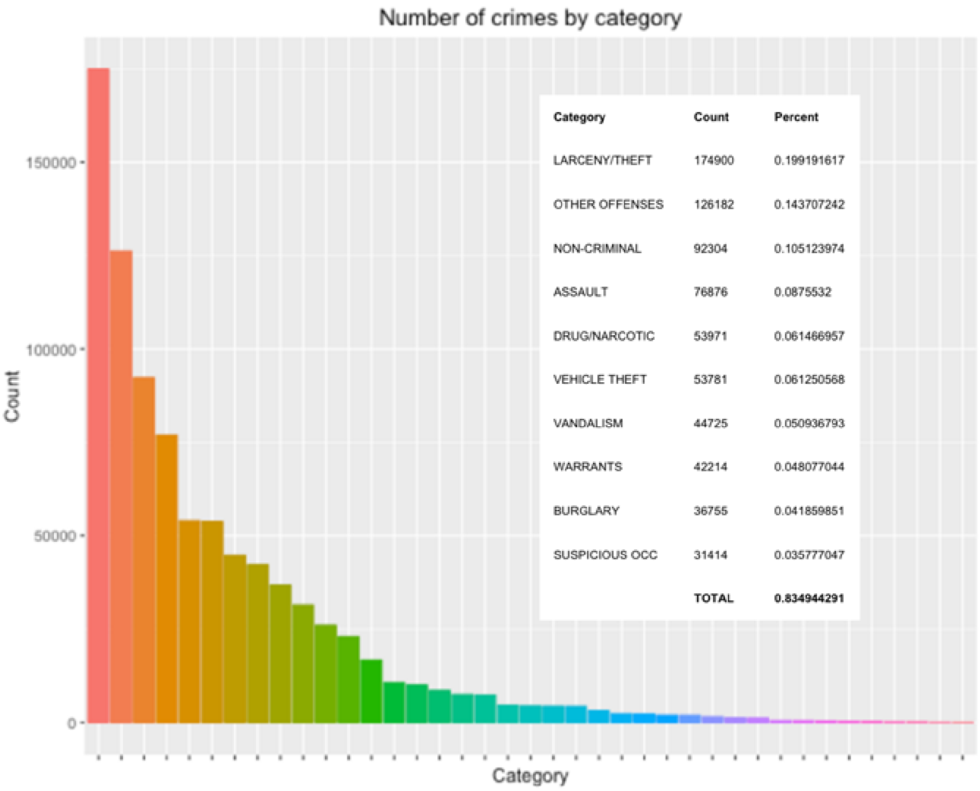
## 1 Introduction

In 2009, San Francisco became among the first American cities to publish and maintain government records for public download and consumption through the `data.sfgov.org` portal. The site makes data available on many municipal aspects, including transportation usage, city infrastructure, and broader socioeconomic statistics. In the summer of 2015, it released data on crime inside the city proper. The set included records of 878,049 crimes classified into one of 39 different categories across 10 San Francisco police districts. These categories ranged in type from forcible sexual assault to embezzlement to liquor law infractions, among others. Each crime record also included information related to the geographic location of the incident, the time and date it occurred, and its eventual resolution. The city challenged data scientists and statisticians to predict the category of a crime given its time and location. We adopted this goal for our project, choosing to apply three classification methods – random forests, boosting, and $k$-nearest neighbors.

Although many aspects of our exploration were instructional and/or academic in nature, the ramifications of creating a successful classifier are significant. The capacity to correctly classify crimes using only temporal and geographic features could give a significant advantage to law enforcement and city officials in their pursuit of a safer city.

## 2 Initial Analysis

Exploratory data analysis revealed interesting patterns in the data and foreshadowed a difficult classification problem. Counts of each class are presented in the figure below, which shows imbalance between class sizes – LARCENY/THEFT has more than 175,000 observations, whereas the smallest class, TREA, has only six across the 2003 to 2013 data. The ten largest classes accounted for roughly 83.5% of the observations.



| Category | Count | Percent |
|---|---|---|
| LARCENY/THEFT | 174900 | 0.199191617 |
| OTHER OFFENSES | 126182 | 0.143707242 |
| NON-CRIMINAL | 92304 | 0.105123974 |
| ASSAULT | 76876 | 0.0875532 |
| DRUG/NARCOTIC | 53971 | 0.061466957 |
| VEHICLE THEFT | 53781 | 0.061250568 |
| VANDALISM | 44725 | 0.050936793 |
| WARRANTS | 42214 | 0.048077044 |
| BURGLARY | 36755 | 0.041859851 |
| SUSPICIOUS OCC | 31414 | 0.035777047 |
|  | **TOTAL** | **0.834944291** |

Figure 1

Classes also displayed lack of separation along time and geographic features. The plots below show simple random samples of LARCENY/THEFT, ASSAULT, and VANDALISM occurrences. Crimes across nearly all categories displayed the same spikes in frequency in the Tenderloin district.

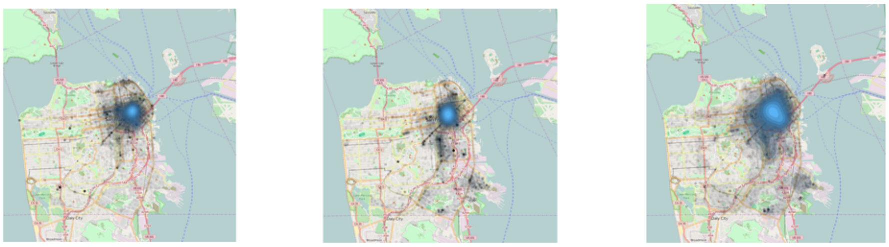


Figure 2: Geographic Density of Crimes

The DATE feature, originally coded to include the date and time of a crimes occurrence in the format “YYYY-MM-DD HH:MM:SS” was split into seven new variables: Year, Month, DayOfMonth, Hour, YearsMo, DayOfWeek, and weekday. This allowed us to explore the temporal trends in crime. Notably, classes displayed similar distributions across most time variables, as exemplified in the plot below.

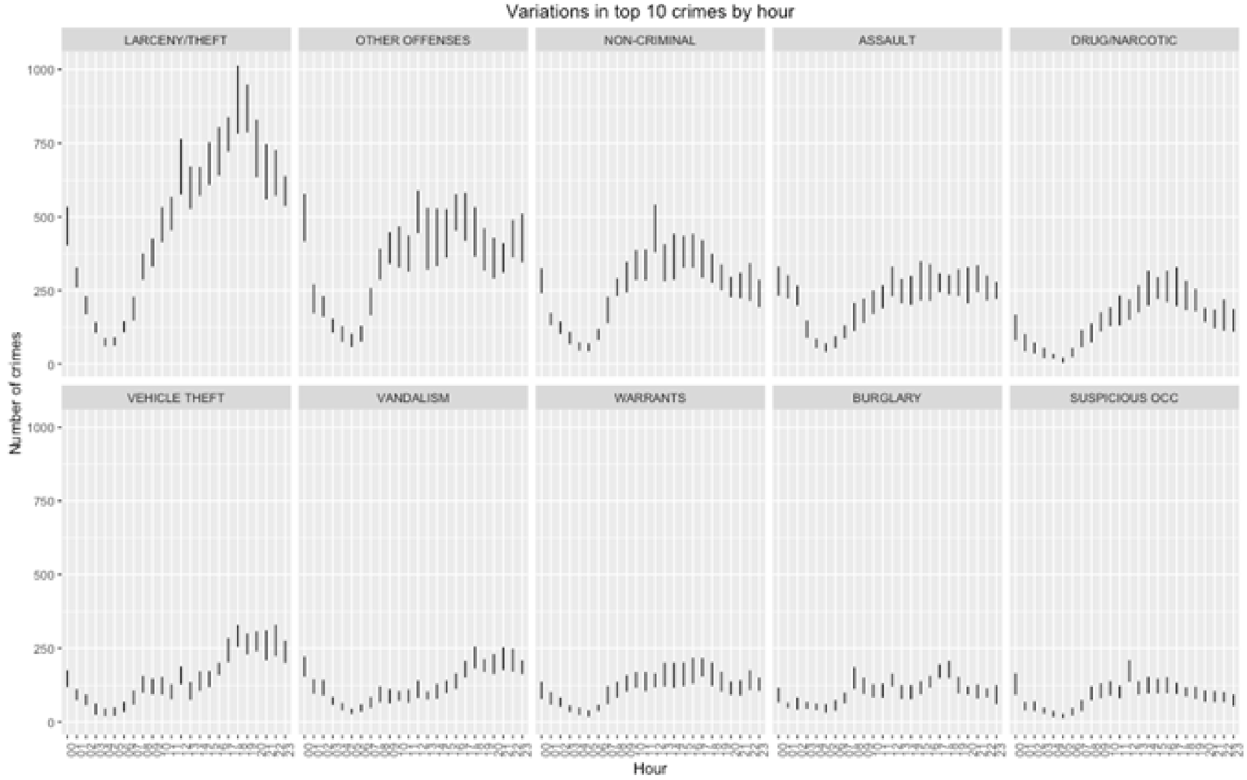


Figure 3

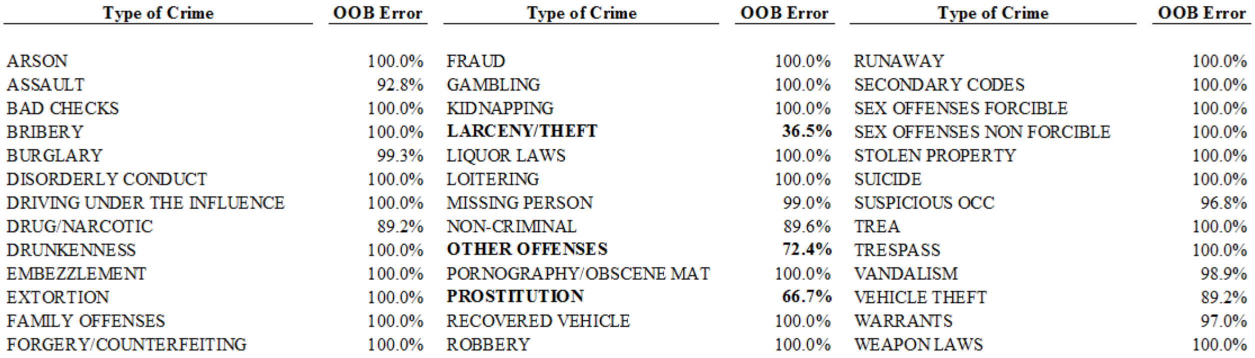
Uniformly, San Francisco is at its safest between the hours of 3:00 and 5:00, and is most prone to common crimes between the hours of 14:00 and 21:00. Although the variance of a particular class across hours is unique, the trend across each one is similar, which raises questions regarding the feature's utility in classification.

# 3 Benchmarks

In this section, we benchmark each classifier. As discussed above, we separated the "Dates" variable into the year, month, day of month, and hour prior to creating the initial model; however, we made no other changes to the variables and generally used the default settings for each model.

## 3.1 Random Forest

For the random forest benchmark, we ran the random forest algorithm on a proportionate stratified sample of the training set. Using the day of week, the police district, the X and Y coordinates, along with the separated time variables yielded an out of bag ("OOB") error rate of 79.78%. We also examine the class OOB error rates for our benchmark fit below:



| Type of Crime | OOB Error | Type of Crime | OOB Error | Type of Crime | OOB Error |
|---|---|---|---|---|---|
| ARSON | 100.0% | FRAUD | 100.0% | RUNAWAY | 100.0% |
| ASSAULT | 92.8% | GAMBLING | 100.0% | SECONDARY CODES | 100.0% |
| BAD CHECKS | 100.0% | KIDNAPPING | 100.0% | SEX OFFENSES FORCIBLE | 100.0% |
| BRIBERY | 100.0% | **LARCENY/THEFT** | **36.5%** | SEX OFFENSES NON FORCIBLE | 100.0% |
| BURGLARY | 99.3% | LIQUOR LAWS | 100.0% | STOLEN PROPERTY | 100.0% |
| DISORDERLY CONDUCT | 100.0% | LOITERING | 100.0% | SUICIDE | 100.0% |
| DRIVING UNDER THE INFLUENCE | 100.0% | MISSING PERSON | 99.0% | SUSPICIOUS OCC | 96.8% |
| DRUG/NARCOTIC | 89.2% | NON-CRIMINAL | 89.6% | TREA | 100.0% |
| DRUNKENNESS | 100.0% | **OTHER OFFENSES** | **72.4%** | TRESPASS | 100.0% |
| EMBEZZLEMENT | 100.0% | PORNOGRAPHY/OBSCENE MAT | 100.0% | VANDALISM | 98.9% |
| EXTORTION | 100.0% | **PROSTITUTION** | **66.7%** | VEHICLE THEFT | 89.2% |
| FAMILY OFFENSES | 100.0% | RECOVERED VEHICLE | 100.0% | WARRANTS | 97.0% |
| FORGERY/COUNTERFEITING | 100.0% | ROBBERY | 100.0% | WEAPON LAWS | 100.0% |

Figure 4: OOB Class Error Rates

As we can see, the majority of the error rates are at or near 100%. For the most common category, LARCENY/THEFT, the OOB error rate is 36.5%. Furthermore, there are two categories with roughly a two-thirds error rate, and the rest are close to or above 90%.

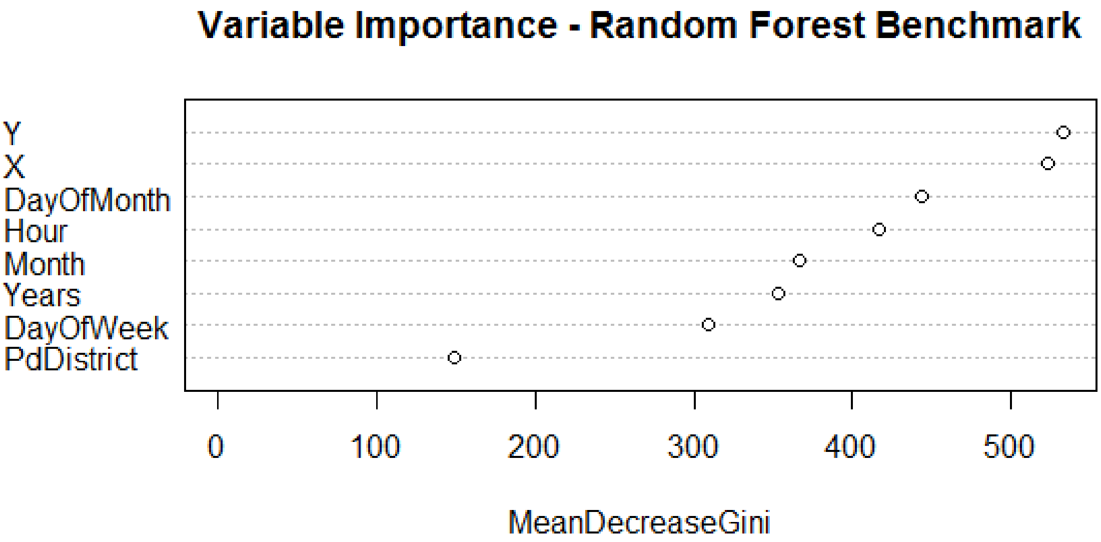


Figure 5: Variable Importance for Random Forest

In the figure above, we consider the variable importance plot for the benchmark random forest. As we can see, the most important variables are the X and Y coordinates, while police district is the least important. This indicates that random forest prefers the increased granularity of X and Y coordinates relative to police district for classification. We can also see how the random forest seems to view the time variables: longer time periods (such as month or year) are viewed as less important than shorter time periods. This seems to indicate that crime fluctuates more with the day or hour than the season or year.

## 3.2 Boosting

Initial results attempting to classify crime with boosting fared similarly to the other classifiers. Our initial fit was performed using the same variables as in the random forest benchmark (X and Y coordinates, police district, day of week, and the separated time variables). After selecting a proportionate sample relative to crime categories from our testing dataset, cross validation was run on this sample for multiple different numbers of iterations for the adaboost algorithm. The below figure shows our cross-validation error rate for these classifiers:

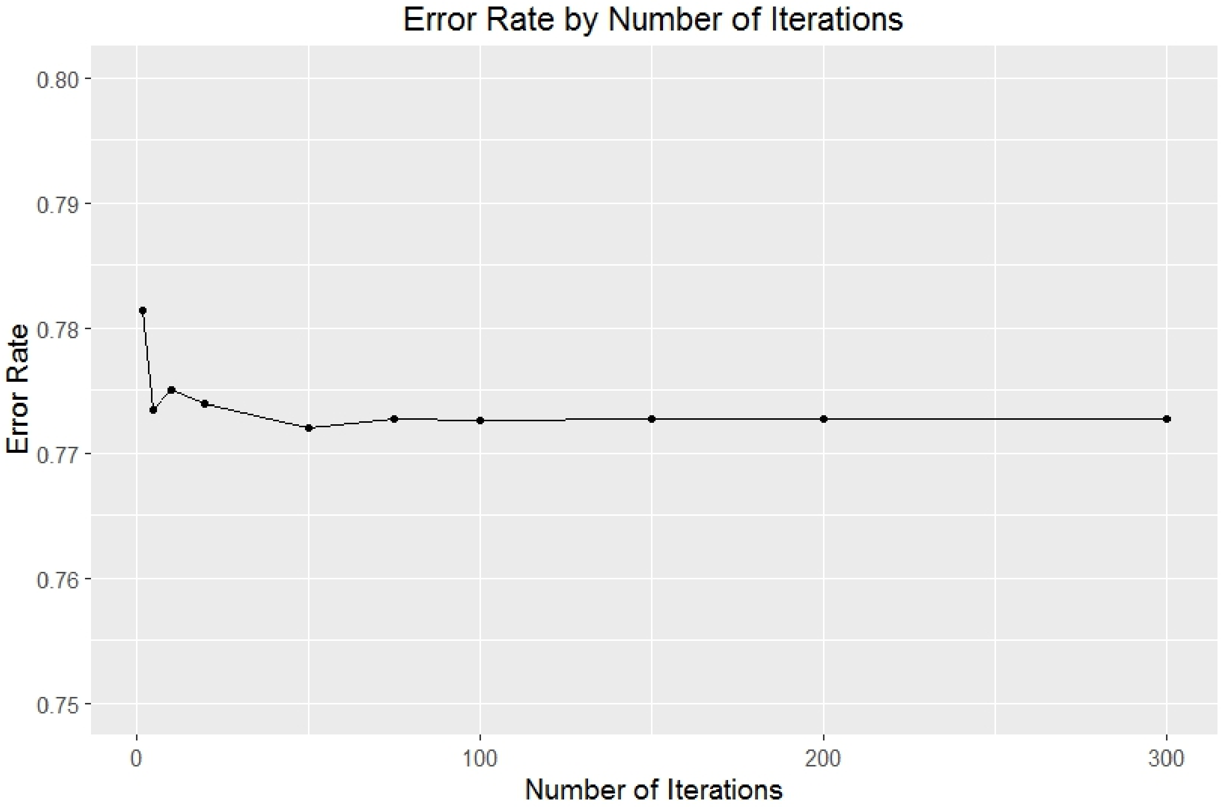
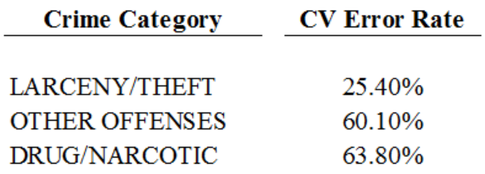


Figure 6

We see that the error rate begins to converge around 77.5% as the iterations increase. Our lowest error rate was seen at 50 iterations with an error rate of 77.2%. Of the 39 categories, only three classes had a CV error less than 100%. These are shown in the figure below. It is worth noting that all three of these categories were in the top 5 categories in the data set.



| Crime Category | CV Error Rate |
|---|---|
| LARCENY/THEFT | 25.40% |
| OTHER OFFENSES | 60.10% |
| DRUG/NARCOTIC | 63.80% |

Figure 7: Boosting Class Error Rates

Next, we examine the variable importances. While the random forest did not consider police district to be important, adaboost considered it the most important by far. The Y coordinate was considered slightly important, and the only other variable above zero was the X coordinate. This was the case across all levels of iterations, as show in the figure below:

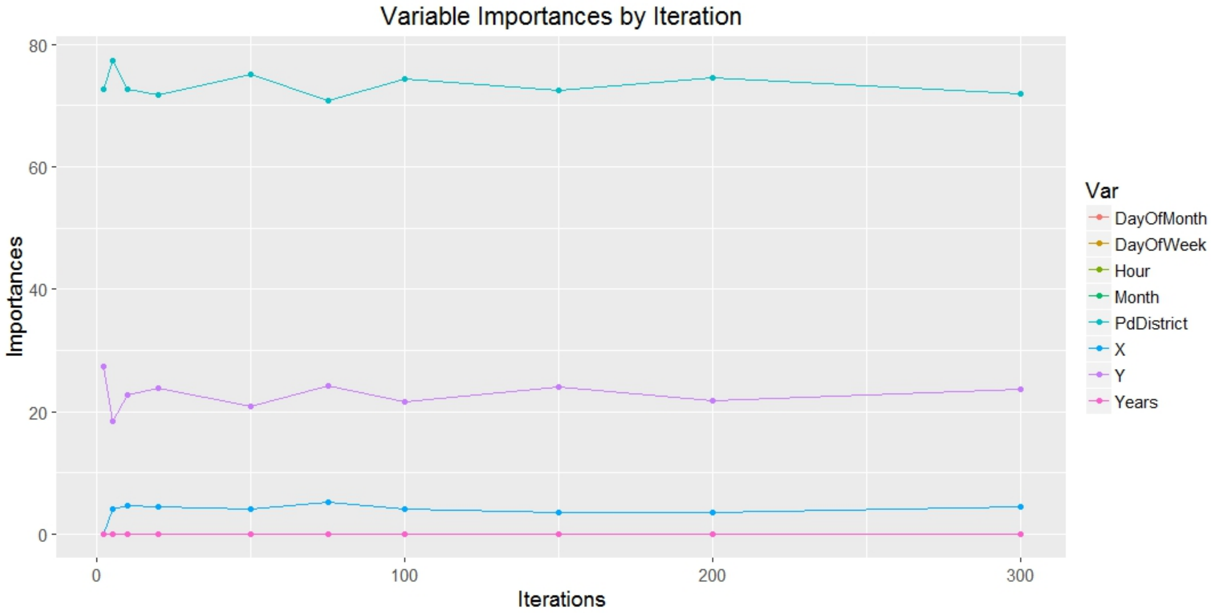
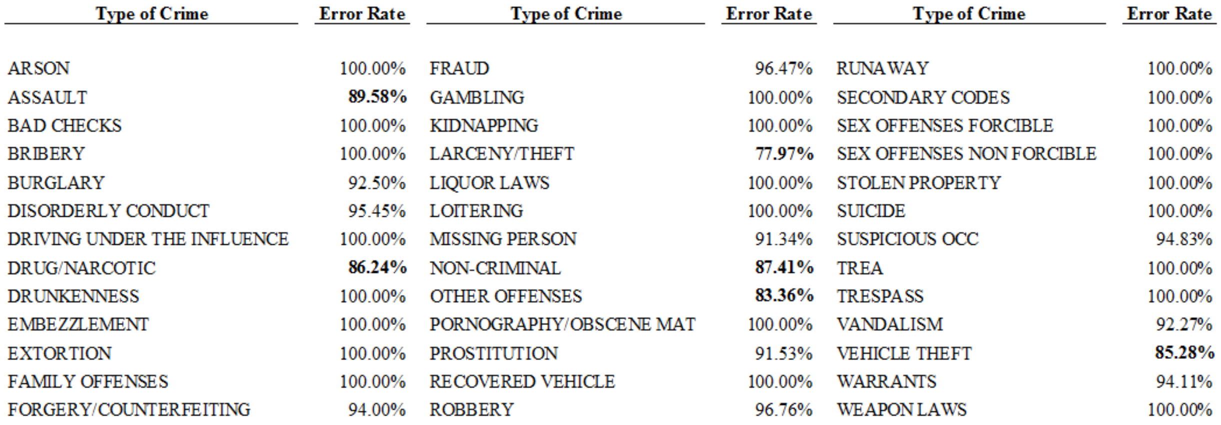


Figure 8

It seems that both adaboost and random forest consider location to be important; however, random forest uses the more granular X and Y coordinate, while adaboost uses the police district.

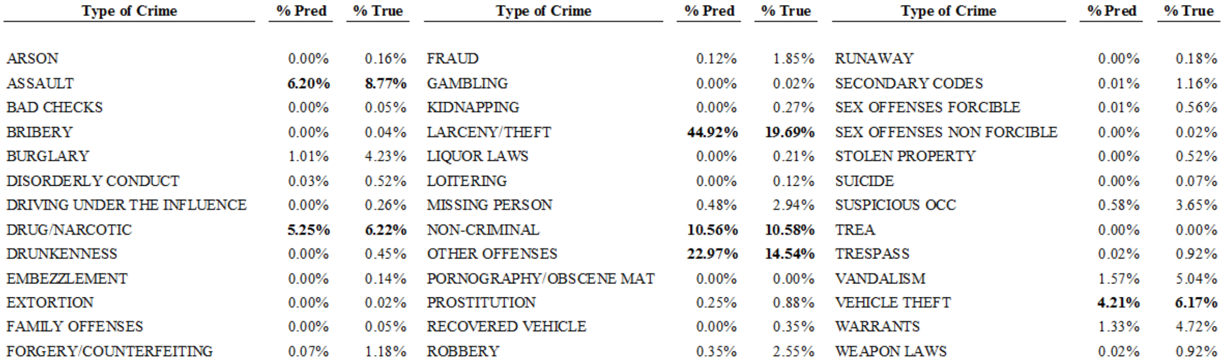
## 3.3 k-Nearest Neighbors

The kNN algorithm was explored as a method of classification and displayed poor performance when applied naively to the data set. The algorithm was applied to a $n = 70,000$ sample of the training data including each of the 39 classes of crime. The function knnEval() from the chemometrics package evaluated the algorithm for a range of $k$ values, and recorded $k$-fold cross-validated errors, allowing us to tune $k$ to an optimal $k = 28$. This initial classifier produced an overall error rate of 81.89%, with class errors displayed below.



| Type of Crime | Error Rate | Type of Crime | Error Rate | Type of Crime | Error Rate |
|---|---|---|---|---|---|
| ARSON | 100.00% | FRAUD | 96.47% | RUNAWAY | 100.00% |
| ASSAULT | **89.58%** | GAMBLING | 100.00% | SECONDARY CODES | 100.00% |
| BAD CHECKS | 100.00% | KIDNAPPING | 100.00% | SEX OFFENSES FORCIBLE | 100.00% |
| BRIBERY | 100.00% | LARCENY/THEFT | **77.97%** | SEX OFFENSES NON FORCIBLE | 100.00% |
| BURGLARY | 92.50% | LIQUOR LAWS | 100.00% | STOLEN PROPERTY | 100.00% |
| DISORDERLY CONDUCT | 95.45% | LOITERING | 100.00% | SUICIDE | 100.00% |
| DRIVING UNDER THE INFLUENCE | 100.00% | MISSING PERSON | 91.34% | SUSPICIOUS OCC | 94.83% |
| DRUG/NARCOTIC | **86.24%** | NON-CRIMINAL | **87.41%** | TREA | 100.00% |
| DRUNKENNESS | 100.00% | OTHER OFFENSES | **83.36%** | TRESPASS | 100.00% |
| EMBEZZLEMENT | 100.00% | PORNOGRAPHY/OBSCENE MAT | 100.00% | VANDALISM | 92.27% |
| EXTORTION | 100.00% | PROSTITUTION | 91.53% | VEHICLE THEFT | **85.28%** |
| FAMILY OFFENSES | 100.00% | RECOVERED VEHICLE | 100.00% | WARRANTS | 94.11% |
| FORGERY/COUNTERFEITING | 94.00% | ROBBERY | 96.76% | WEAPON LAWS | 100.00% |

Figure 9: kNN Class Error Rates



| Type of Crime | % Pred | % True | Type of Crime | % Pred | % True | Type of Crime | % Pred | % True |
|---|---|---|---|---|---|---|---|---|
| ARSON | 0.00% | 0.16% | FRAUD | 0.12% | 1.85% | RUNAWAY | 0.00% | 0.18% |
| ASSAULT | **6.20%** | **8.77%** | GAMBLING | 0.00% | 0.02% | SECONDARY CODES | 0.01% | 1.16% |
| BAD CHECKS | 0.00% | 0.05% | KIDNAPPING | 0.00% | 0.27% | SEX OFFENSES FORCIBLE | 0.01% | 0.56% |
| BRIBERY | 0.00% | 0.04% | LARCENY/THEFT | **44.92%** | **19.69%** | SEX OFFENSES NON FORCIBLE | 0.00% | 0.02% |
| BURGLARY | 1.01% | 4.23% | LIQUOR LAWS | 0.00% | 0.21% | STOLEN PROPERTY | 0.00% | 0.52% |
| DISORDERLY CONDUCT | 0.03% | 0.52% | LOITERING | 0.00% | 0.12% | SUICIDE | 0.00% | 0.07% |
| DRIVING UNDER THE INFLUENCE | 0.00% | 0.26% | MISSING PERSON | 0.48% | 2.94% | SUSPICIOUS OCC | 0.58% | 3.65% |
| DRUG/NARCOTIC | **5.25%** | **6.22%** | NON-CRIMINAL | **10.56%** | **10.58%** | TREA | 0.00% | 0.00% |
| DRUNKENNESS | 0.00% | 0.45% | OTHER OFFENSES | **22.97%** | **14.54%** | TRESPASS | 0.02% | 0.92% |
| EMBEZZLEMENT | 0.00% | 0.14% | PORNOGRAPHY/OBSCENE MAT | 0.00% | 0.00% | VANDALISM | 1.57% | 5.04% |
| EXTORTION | 0.00% | 0.02% | PROSTITUTION | 0.25% | 0.88% | VEHICLE THEFT | **4.21%** | **6.17%** |
| FAMILY OFFENSES | 0.00% | 0.05% | RECOVERED VEHICLE | 0.00% | 0.35% | WARRANTS | 1.33% | 4.72% |
| FORGERY/COUNTERFEITING | 0.07% | 1.18% | ROBBERY | 0.35% | 2.55% | WEAPON LAWS | 0.02% | 0.92% |

Figure 10: kNN Predicted vs. True Class Proportions

It is likely kNN displayed such high class and overall error rates due to the difficulty of the classification problem – qualities of the data that also troubled random forests and adaboost methods. However, kNN performs especially poorly in instances where the "neighborhood" surrounding a candidate for classification displays high heterogeneity – which is exactly the quality of the neighborhood surrounding a crime of an unknown class. Per our exploratory data analysis, crime categories that dominated the data set were not unique or separable by geographic or temporal neighborhood (*i.e.*, times and locations with frequent assaults are also likely to have frequent car thefts). As a consequence, neighborhoods without "strong majorities" create many misclassification errors. Analysis of the figure above shows that kNN identifies large classes within the original sample (LARCENY/THEFT and OTHER OFFENSES, for instance) and over-predicts by these classes. Only the largest classes have classification errors below 90%. Rarer crimes, such as PORNOGRAPHY/OBSCENE MAT, are most likely never in a neighborhood that would produce a majority vote in favor of its correct classification, and as such are misplaced by the algorithm into larger classes.

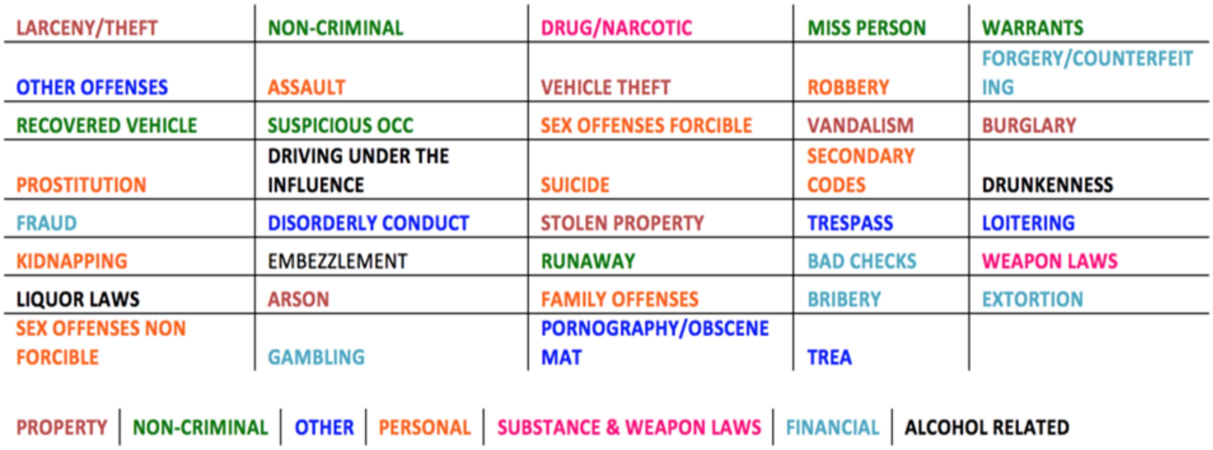
# 4 Classification Issues

## 4.1 Numerous and Similar Classes

The primary issue in accurately classifying the data was the number of classes, along with the similarity between the class descriptions. As mentioned above, there were a total of 39 classes. This made it difficult to accurately predict which class an observation belonged to. Furthermore, many classes were similar. For example, there are several classes related to theft: BURGLARY, LARCENY/THEFT, ROBBERY, and VEHICLE THEFT. Similarly, there are classes for trespassing and loitering, along with "TREA," which is described in the data set as "trespassing or loitering near posted industrial property."

### 4.1.1 Combining Classes

The number of classes and the lack of meaningful differences among them influenced classifier performance. The figure below shows one solution to this problem. Initial analysis of the classes revealed many fell into one of five state- and federally-defined categories: property, personal, financial, non-criminal, and other crimes. After consideration of those classes that fell into the other category, we decided to condense the category into 7 larger groups in hopes of improving class and overall error rates



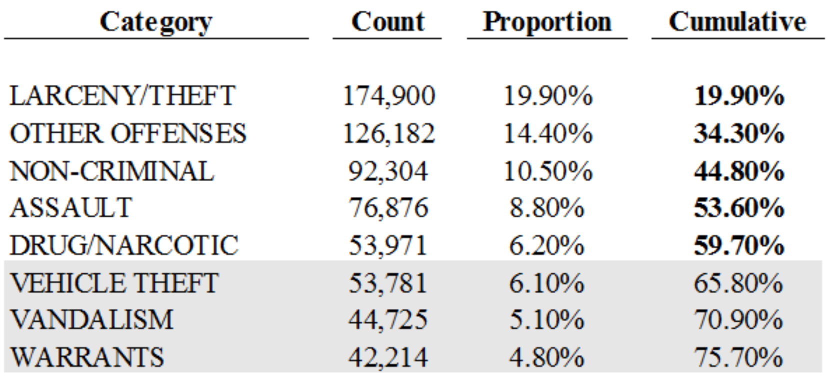
| | | | | |
|---|---|---|---|---|
| LARCENY/THEFT | NON-CRIMINAL | DRUG/NARCOTIC | MISS PERSON | WARRANTS |
| OTHER OFFENSES | ASSAULT | VEHICLE THEFT | ROBBERY | FORGERY/COUNTERFEIT ING |
| RECOVERED VEHICLE | SUSPICIOUS OCC | SEX OFFENSES FORCIBLE | VANDALISM | BURGLARY |
| PROSTITUTION | DRIVING UNDER THE INFLUENCE | SUICIDE | SECONDARY CODES | DRUNKENNESS |
| FRAUD | DISORDERLY CONDUCT | STOLEN PROPERTY | TRESPASS | LOITERING |
| KIDNAPPING | EMBEZZLEMENT | RUNAWAY | BAD CHECKS | WEAPON LAWS |
| LIQUOR LAWS | ARSON | FAMILY OFFENSES | BRIBERY | EXTORTION |
| SEX OFFENSES NON FORCIBLE | GAMBLING | PORNOGRAPHY/OBSCENE MAT | TREA | |

PROPERTY | NON-CRIMINAL | OTHER | PERSONAL | SUBSTANCE & WEAPON LAWS | FINANCIAL | ALCOHOL RELATED

Figure 11: Condensed Classes

### 4.1.2 Focus on High Frequency Crime

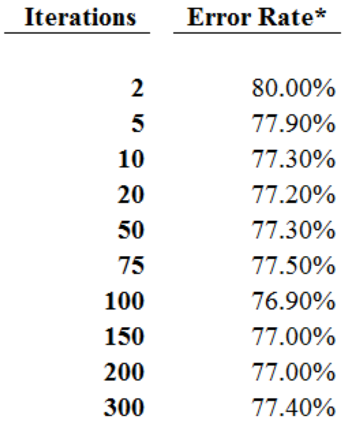
We also tried training our classifier on only the high frequency crimes. The table below shows that, out of 39 types of crimes, the top five crimes accounted for nearly 60% of our data, with the top eight accounting for over 75%. Given this information, it seems reasonable to think that training on just 5 specific types of crime may improve classification overall, as it will hone in on and emphasize those crimes dominating the data set.



| Category | Count | Proportion | Cumulative |
|---|---|---|---|
| LARCENY/THEFT | 174,900 | 19.90% | **19.90%** |
| OTHER OFFENSES | 126,182 | 14.40% | **34.30%** |
| NON-CRIMINAL | 92,304 | 10.50% | **44.80%** |
| ASSAULT | 76,876 | 8.80% | **53.60%** |
| DRUG/NARCOTIC | 53,971 | 6.20% | **59.70%** |
| VEHICLE THEFT | 53,781 | 6.10% | 65.80% |
| VANDALISM | 44,725 | 5.10% | 70.90% |
| WARRANTS | 42,214 | 4.80% | 75.70% |

Figure 12: Cumulative Proportion of Crime

To illustrate, we show the results of this approach using the adaboost method. We trained multiple classifiers (each for different numbers of iterations) on a proportionate sub-sample from the training data of observations with just the top five crimes. We then used cross validation on the rest of the training data and report the error rates below. This method yielded almost the exact same cross validated error rate as before (a low of 76.9% vs. 77.2% previously). It appears that focusing on only the high-frequency categories of crime does not help us.



| Iterations | Error Rate* |
|---|---|
| **2** | 80.00% |
| **5** | 77.90% |
| **10** | 77.30% |
| **20** | 77.20% |
| **50** | 77.30% |
| **75** | 77.50% |
| **100** | 76.90% |
| **150** | 77.00% |
| **200** | 77.00% |
| **300** | 77.40% |

Figure 13: CV Error Rates using Top 5 Crimes

## 4.2 Unbalanced Classes

As mentioned above, the classes were very unbalanced. The largest class, larceny and theft, made up 20% of the data set, while the smallest class, trea, had only 6 observations. As such, if we were to construct a classifier that assigned each observation to larceny and theft, our results would be very similar to those of the benchmarks described above. For example, the random forest classifier had an OOB error rate of 79.78%, with a class error rate of 36.5%

for larceny and theft. Our hypothetical classifier would have an overall error rate of 80% with a class error rate of 0% on larceny and theft.

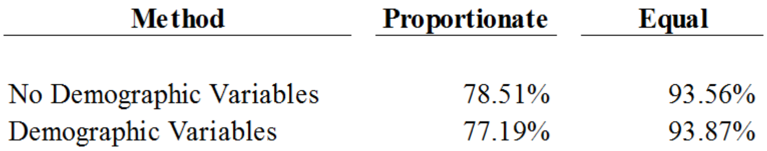
### 4.2.1 Equal Sampling

To overcome this issue, we attempted to use an equal sampling method. Rather than sampling a proportionate amount from each of the 39 classes (or 7 classes in our reduced data set), we sampled an even number from each class, up-sampling where necessary. As we will see below, using an equal sampling method resulted in a higher overall error rate, as it reduces the error rate on the less common classes and increases the error rate on more common classes.

## 4.3 Few Features, Lack of Separation

Not only are there few features relative to the number of classes, there is a lack of separation among classes for the given features. Furthermore, we are given two types of features: time (day of week and date/time) and location (police district and X and Y coordinates). As we noted earlier, most crime happens close together geographically.

One way to deal with the lack of features relative to the categories is to add features. To do this, we found demographic data by police district[1]. This included population, as well as breakdown by race and gender, for each district. While this data was only for a single point in time, we added these variables hoping that they would help to distinguish between similar police districts. For example, when only the name of the district is included in the model, there is no way to tell whether the Northern district is more similar to the Mission or Southern districts. However, looking at demographic data, we can see that both the Northern and Mission district have a population of roughly 80,000, while the Southern district has 24,157 people.



| Method | Proportionate | Equal |
|---|---|---|
| No Demographic Variables | 78.51% | 93.56% |
| Demographic Variables | 77.19% | 93.87% |

Figure 14: Error Rates with and without Demographic Variables

In the figure above, we compare error rates for random forest with and without demographic variables. As we can see, adding in the demographic variables resulted in a slight decrease in the error rates for the proportionate sampling method, and basically no change in error rate for the equal sampling method.

[1] Data are from page 25: `http://sanfranciscopolice.org/sites/default/files/FileCenter/Documents/14683-SFPD_DSBAfinal_trnsmtl.pdf`

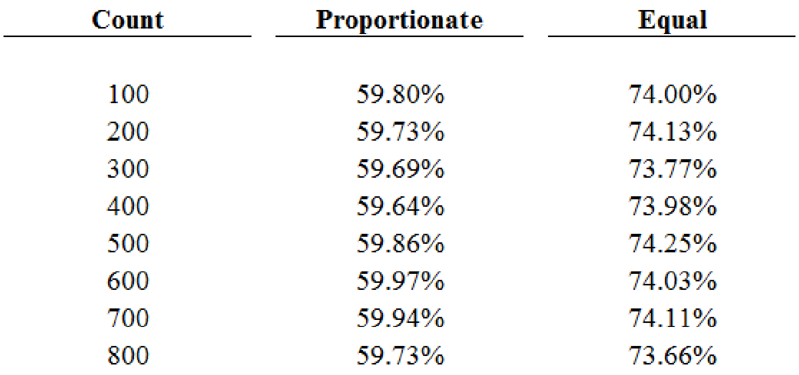
# 5 Random Forests

## 5.1 Sample Size

As mentioned above, the initial fit for the random forest model was created using a sample of the training set due to the number of observations ($n = 702,439$). To deal with the computational issues associated with this large data set, we considered three options. The first was to use a sample from the training set as we did in the initial fit. While this method was computationally expedient, we believed that using this method would result in losing valuable information from the rest of the data set. The second solution we considered was to specify the size of the sample within the random forest algorithm itself using the "sampsize" option. While this would have enabled us to sample more of the data set, using this option was very slow. The last option we considered was to create multiple random forests with fewer trees in each forest, using a different sample each time. We then combined the forests into a single forest which we used to make predictions. By using this method, we sampled more of the training set, while avoiding the excessive computational time of the second option. However, combined random forests do not output out of bag error rate. As such, we set aside roughly 10,000 observations (sampled proportionately from the original training set) as a validation set. For the remainder of the random forest section, we will refer to the validation error in assessing the error rates for the random forests.

## 5.2 Tuning Parameters

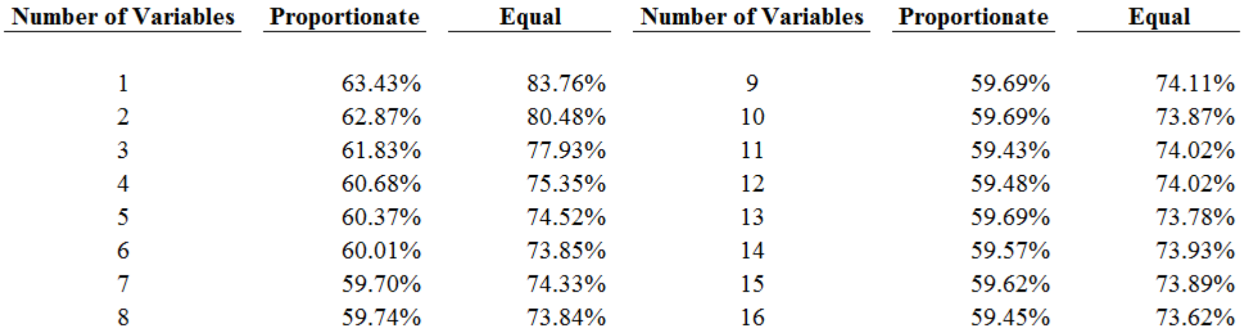
For the random forest algorithm, there were two parameters that we needed to tune: number of trees and the number of variables to try at each split. As mentioned above, we constructed our random forest model by taking multiple samples and creating a forest from each sample. To simplify the search for the optimal number of trees, we set the number of trees in each forest to 10 and then tested the sensitivity of our validation error rate to the number of forests created. The figure below shows the validation error rate compared to the number of trees for both the proportionate and equal sampling methods. Note that the default number of trees for the random forest algorithm is 500, while each of these trials range 1,000 to 8,000 trees. Because there is no discernible difference between using 100 forests of 10 trees and 800 forests of 10 trees for either proportionate or equal sampling, we chose to use 100 forests.



| Count | Proportionate | Equal |
|---|---|---|
| 100 | 59.80% | 74.00% |
| 200 | 59.73% | 74.13% |
| 300 | 59.69% | 73.77% |
| 400 | 59.64% | 73.98% |
| 500 | 59.86% | 74.25% |
| 600 | 59.97% | 74.03% |
| 700 | 59.94% | 74.11% |
| 800 | 59.73% | 73.66% |

Figure 15: Validation Error Rates compared to Number of Forests

The next input to tune was the number of variables to try at each split. Once we included the demographic variables discussed above, we had a total of 16 variables. Because the number of features was relatively limited, we chose to test each value from 1 to 16.



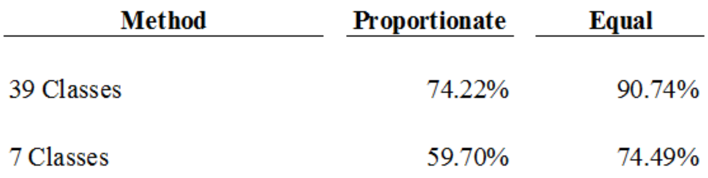
| Number of Variables | Proportionate | Equal | Number of Variables | Proportionate | Equal |
|---|---|---|---|---|---|
| 1 | 63.43% | 83.76% | 9 | 59.69% | 74.11% |
| 2 | 62.87% | 80.48% | 10 | 59.69% | 73.87% |
| 3 | 61.83% | 77.93% | 11 | 59.43% | 74.02% |
| 4 | 60.68% | 75.35% | 12 | 59.48% | 74.02% |
| 5 | 60.37% | 74.52% | 13 | 59.69% | 73.78% |
| 6 | 60.01% | 73.85% | 14 | 59.57% | 73.93% |
| 7 | 59.70% | 74.33% | 15 | 59.62% | 73.89% |
| 8 | 59.74% | 73.84% | 16 | 59.45% | 73.62% |

Figure 16: Validation Error Rates compared to Number of Variables at Each Split

In the figure above, we see that while lower “mtry” resulted in lower validation error rates for both proportionate and equal sampling, there was not much difference once the number of variables reached 7. We chose to use 11 variables at each split, as the validation error was lowest at this value.

## 5.3 Results

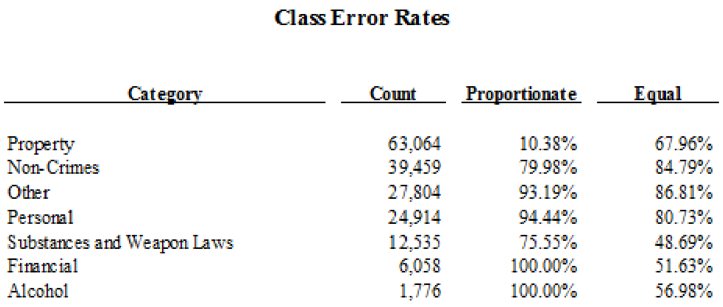
In the figure below, we include the test error rate using our final random forest model (using 100 forests of 10 trees each and 11 variables tried at each split). As expected, the proportionate sample outperforms the equal sample in overall error rate. Similarly, using 7 classes rather than 39 results in lower error rates.



| Method | Proportionate | Equal |
|---|---|---|
| 39 Classes | 74.22% | 90.74% |
| 7 Classes | 59.70% | 74.49% |

Figure 17: Test Error Rates for Random Forest

Next, we examine the class error rates for the 7 classes. As we can see, when we use a proportionate sample, the error rate for property crimes is only 10%; however, the error rates for the other 6 classes are all above 75%. For financial crimes and alcohol-related crimes, the error rates are 100% (likely due to the scarcity of these crimes relative to the 5). When we examine the equal sampling rate, we can see that the error rate on the most common two classes are higher, while the error rates on the other five classes are lower. In other words, the random forest performs better on most classes, yet performs worse overall because its classification error on the common classes has increased.



**Class Error Rates**

| Category | Count | Proportionate | Equal |
|---|---|---|---|
| Property | 63,064 | 10.38% | 67.96% |
| Non-Crimes | 39,459 | 79.98% | 84.79% |
| Other | 27,804 | 93.19% | 86.81% |
| Personal | 24,914 | 94.44% | 80.73% |
| Substances and Weapon Laws | 12,535 | 75.55% | 48.69% |
| Financial | 6,058 | 100.00% | 51.63% |
| Alcohol | 1,776 | 100.00% | 56.98% |

Figure 18: Test Error Rates by Class for Random Forest

Finally, we consider the variable importances for the final model. Below we show the variable importance plot for the 39 category random forest. Note that the order and relative magnitude is the same for the 7 category random forest. The order of the variable importance remain unchanged for the variables from the benchmark random forest. The Y and X coordinates are most important, followed by the time variables, day of week, and then the new demographic variables and police district. The new demographic variables are on par with police district in terms of importance.

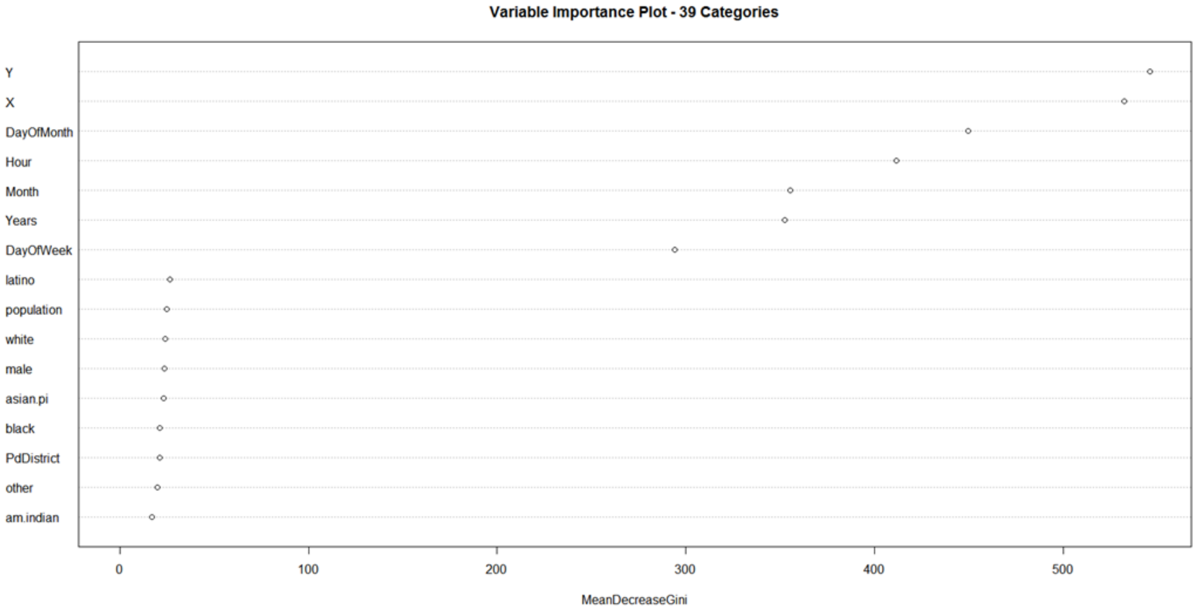
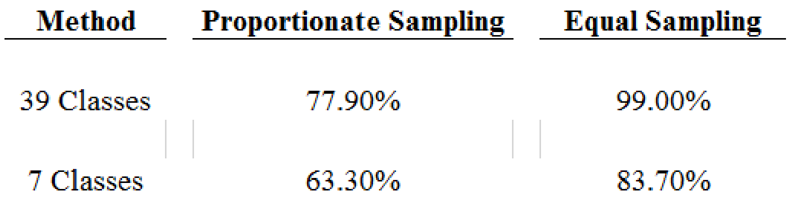


Figure 19: Variable Importance Plot for Random Forest

# 6 Boosting

## 6.1 Equal Sampling and Category Reduction

The error rates for both sampling methods and both set of categories are shown below. These are based on a classifier trained on 50 iterations of the adaboost algorithm:

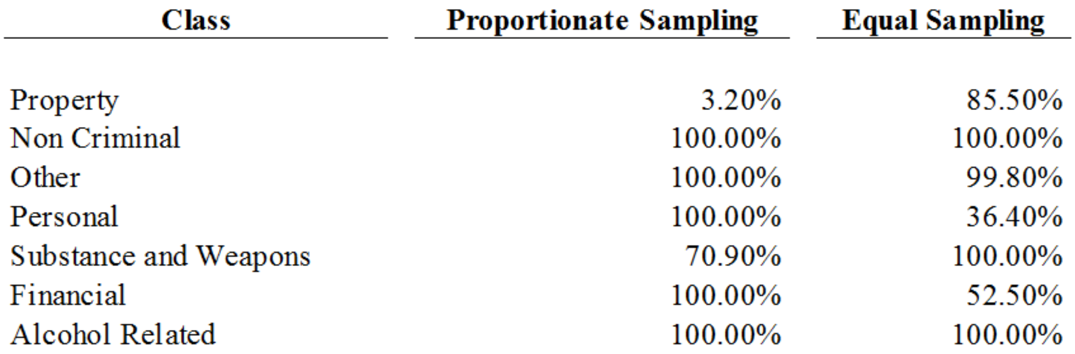


| Method | Proportionate Sampling | Equal Sampling |
|---|---|---|
| 39 Classes | 77.90% | 99.00% |
| 7 Classes | 63.30% | 83.70% |

Figure 20: Boosting Test Error Rates

We see that using equal sampling resulted in a much higher error rate. The error rate was 99%, due to an increased classification of seldom occurring crimes. The category reduction did result in a lower error rate, which is expected as 7 classes are easier to classify than 39. Finally, it is worth noting that while these error rates for proportionate sampling are quite high, this is still an improvement over random guessing, given the large number of classes.

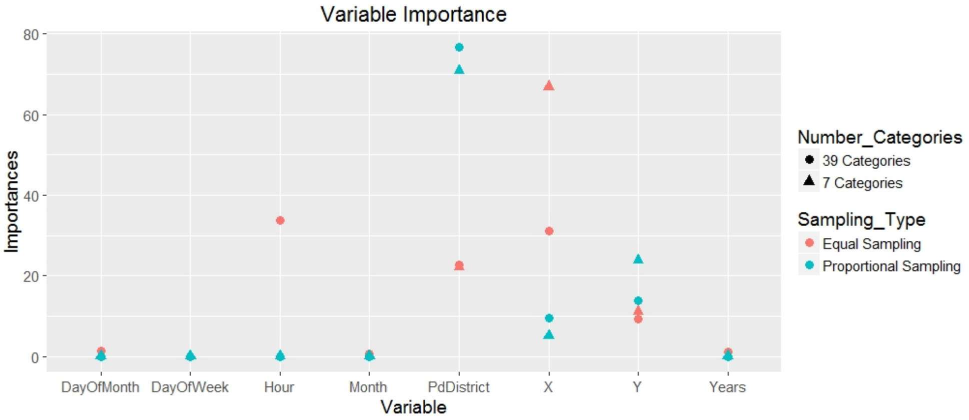
Another helpful table is the error rate for each of the 7 classes:



| Class | Proportionate Sampling | Equal Sampling |
|---|---:|---:|
| Property | 3.20% | 85.50% |
| Non Criminal | 100.00% | 100.00% |
| Other | 100.00% | 99.80% |
| Personal | 100.00% | 36.40% |
| Substance and Weapons | 70.90% | 100.00% |
| Financial | 100.00% | 52.50% |
| Alcohol Related | 100.00% | 100.00% |

Figure 21: Boosting Class Error Rates

As we discussed above, our classifier focuses on the most common type of crime, property crimes, in the case of proportionate sampling. The discrepancy is even more notable in this case, with property crimes being correctly classified almost 97% of the time. On the other hand, the equal sampling method is able to categorize groups which occur less frequently. Our preference of classifiers would depend on the loss function being used (*e.g.*, if classifying an infrequently occurring crime is very important, equal sampling might be the preferred method). Lastly, we can look at what happens to variable importance with these changes to our training data:



As we can see, the importance of the hour and X coordinate variables are higher in the equal sampling case, while they were at or near zero in our benchmark. This indicates that seldom occurring crimes may be more easily classified by variables which are less important for common crimes.

# 7 k-Nearest Neighbors

## 7.1 Weighting distances and representing dissimilarity in categorical features

kNN is distinguished from the other classifiers applied to the data by its assumption that the observations exist in a metric space, such that distances between observations can be calculated to create groups of $k$-nearest neighbors. Data are commonly scaled and centered before the algorithm is trained to ensure features of disparate metrics and variances do not produce disparate weights in the algorithms decisions. Moreover, categorical variables are often coded into numeric columns, so that important information regarding data can be mapped into the dissimilarities calculated between observations. This process proved especially subjective in the case of the San Francisco crime data, which included several categorical features.

First, consider the day of the week on which a particular crime occurred (*i.e.*, Monday through Sunday). We suspected including this information in the dissimilarity matrix might improve classification, insofar as the day of the week might lend structure to the data (for instance, perhaps business districts are less populous on the weekends, or perhaps crimes such as DUIs are more likely to occur on the weekends). Problematically, the simple mapping $(Mon = 1, Tue = 2, \ldots, Sun = 7)$ misrepresents the dissimilarity between days, because it makes the most dissimilar pair $(Mon = 1, Sun = 7)$, which truthfully represents no further distance than $(Wed = 3, Thu = 4)$.

A more equitable method of representing these factors is in representing each day as a unit vector in $\mathbb{R}^7$, such that $Mon = (1, 0, 0, 0, 0, 0, 0)$ and $Thu = (0, 0, 0, 1, 0, 0, 0)$, for instance. This mapping considers each day of the week as equidistant from the others. For example, Monday is no further from Tuesday than it is from Friday. While this mapping is still false, it is considerably better than the previous option, as it merely fails to fully capture the dissimilarity between days of the week, rather than create false and inflated dissimilarities like the prior method did.

Similar uncertainty arose when considering how kNN might handle date and time variables. In a fashion similar to the days of the week in which two crimes occurred, the dissimilarity between a crime that happened in hour 23 and in hour 2 would be artificially inflated. So too is the case for a crime that occurred on the 29th of a given month and the 3rd of the next. Using the same method chosen for the day of the week would have dramatically increased the dimension of the features if applied to the hour and date features and thus diffused the "neighborhood" of $k$ neighbors. Another option for the date variables would be to re-express them as numbers between 1 and 4017, corresponding to the number of days in the domain of the data, which ranged from 1/1/2003 to 12/31/2013. However, such coding would fail to express structure in the data corresponding to any periodic trends across a month (for instance, perhaps robberies were more likely to occur near the end or beginning of months, when rent is due). Moreover, adding in so many integer values increases the risk of ties in determining the $k$-nearest neighbors, further requiring researchers add in very small noise variables to ensure the algorithm will work. Without meaningful justification for these or other speculations, there appeared little reason to choose one numeric mapping over another. As a consequence, these variables were left untreated.

Finally, it is worth noting that the addendum of demographic information to the feature set resulted in a way of creating adequate dissimilarity measures between police districts, which otherwise would have been coded much in the same way as the days of the week feature. Providing racial breakdowns and population sizes for each police district allowed kNN to determine if and how certain police districts should be considered closer to one another than others.

## 7.2 Equal vs. proportionate sampling after category reduction

After creating numeric mappings of categorical data, adding in demographic information by police district, and condensing categories from 39 to 7 classes, kNN was re-applied to samples of both equal- and proportionate-class allocations. The figure below shows plots of training, cross-validation, and test errors on both samples.

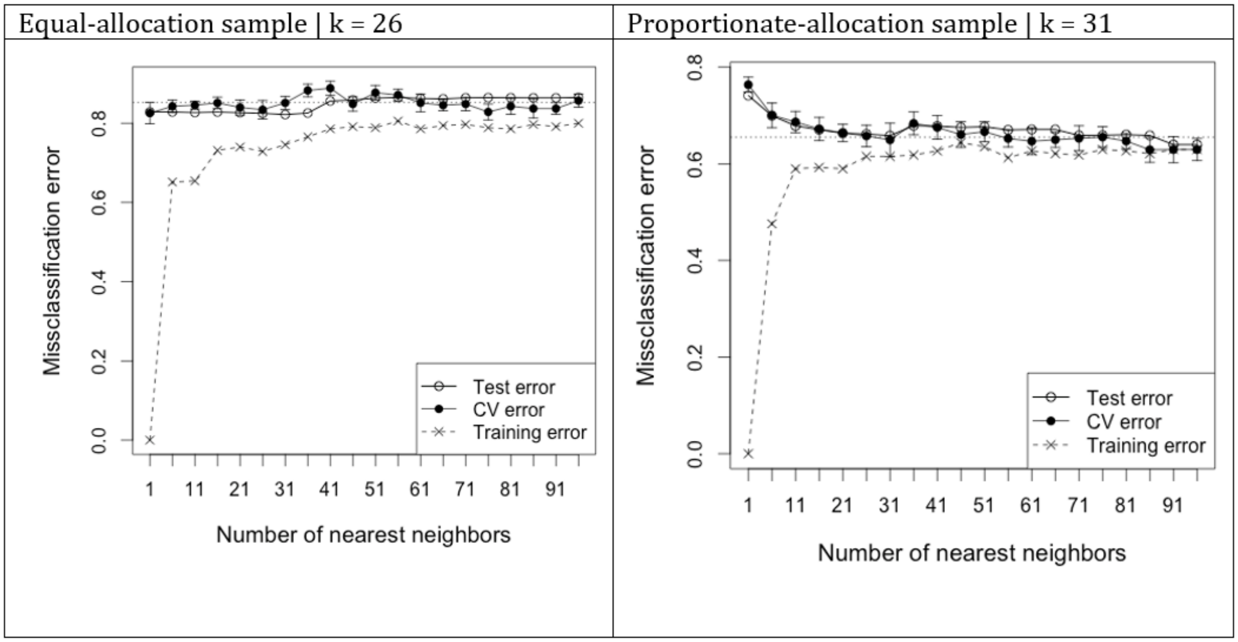
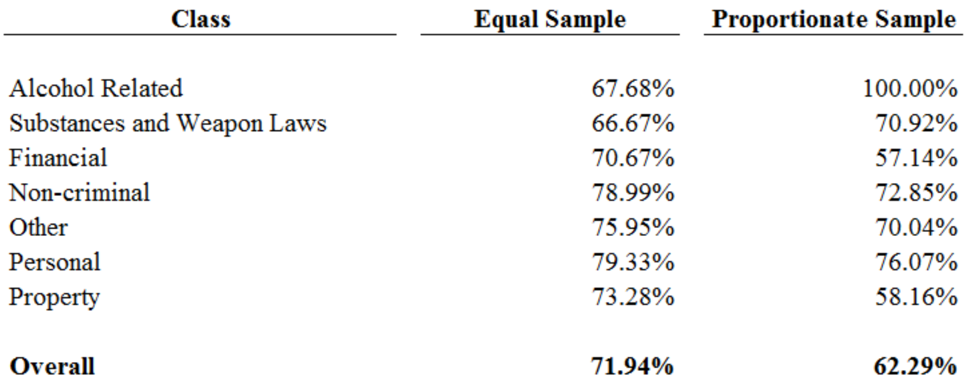


Figure 22: Training, CV and test error for final kNN models



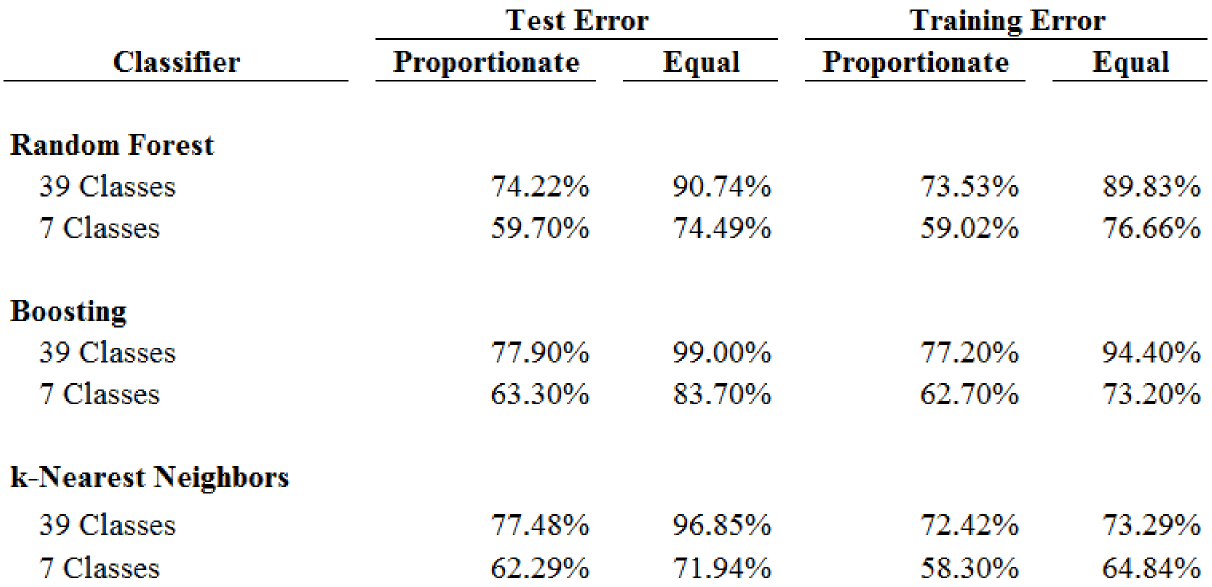
| Class | Equal Sample | Proportionate Sample |
|---|---|---|
| Alcohol Related | 67.68% | 100.00% |
| Substances and Weapon Laws | 66.67% | 70.92% |
| Financial | 70.67% | 57.14% |
| Non-criminal | 78.99% | 72.85% |
| Other | 75.95% | 70.04% |
| Personal | 79.33% | 76.07% |
| Property | 73.28% | 58.16% |
| **Overall** | **71.94%** | **62.29%** |

Figure 23: Class Error Rates for final kNN model

We observed significant improvements over the original 81.89% overall classification error after processing data to better meet kNNs assumptions of a metric space, adding in demographic data, and condensing classes from 39 to 7 categories. We see that kNN behavior is dependent on the type of sample given for training. Equal sampling of each of the strata produces class errors of roughly similar rates, with an overall classification error of 71.94%. However, sampling proportionate to class size causes kNN to – for the most part – ignore smaller classes in favor of larger ones. We observe that the majority of the decrease in overall classification error in the proportionate sample is due to the increased classification of test observations as property crimes.

It is interesting to note that while all three classifiers achieved similar error rates in the proportionate sampling case, the class error rates were quite different. The error rates for both boosting and random forest were driven mostly by the low error rate on property crimes (at or below 10%). Random forest outperformed kNN in terms of overall error rate on proportionate sampling. On the other hand, kNN had a much higher error rate on property crimes, almost 60%, but achieved substantially lower error rates on most other classes.

# 8 Comparison of Results



| Classifier | Test Error | | Training Error | |
|---|---|---|---|---|
| | Proportionate | Equal | Proportionate | Equal |
| **Random Forest** | | | | |
| 39 Classes | 74.22% | 90.74% | 73.53% | 89.83% |
| 7 Classes | 59.70% | 74.49% | 59.02% | 76.66% |
| **Boosting** | | | | |
| 39 Classes | 77.90% | 99.00% | 77.20% | 94.40% |
| 7 Classes | 63.30% | 83.70% | 62.70% | 73.20% |
| **k-Nearest Neighbors** | | | | |
| 39 Classes | 77.48% | 96.85% | 72.42% | 73.29% |
| 7 Classes | 62.29% | 71.94% | 58.30% | 64.84% |

Figure 24: Training and Test Errors for the Final Models

As we can see in the figure above, each classifier performed better on 7 classes than on 39 classes. As we expect, using 7 classes resulted in a much easier classification problem. Furthermore, each classifier achieved lower overall error rates when using a proportionate sample rather than an equal sample. For kNN, these rates were the closest in the 7 class case, as kNN appeared to have the most success balancing classification on more and less common classes. However, it appears that kNN had difficulty for the equal sampling case when using all 39 classes. We believe this is due to the up-sampling of small classes: in the 39 class case, up-sampling was necessary for many of the classes, while up-sampling was not necessary when using the 7 class set. Heavily up-sampled classes contained replications of the same observation, which kNN treated as multiple observations with identical features and consequently, a mutual distance of zero. Finally, boosting had the largest disparity between proportionate and equal sampling error rates, as boosting had the greatest tendency to assigning observations to the most common class.

We can also see that random forest achieved the lowest error rates in the proportionate sampling case overall. However, as we noted before, this was at the cost of misclassifying less common classes. In the equal sampling case for 7 classes, kNN had the best performance.

Finally, we can see that the random forest and boosting had training and test errors that generally coincided, though boosting appears to overfit in the case of equal sampling for 7 classes. On the other hand, kNN has more of a tendency to overfit. This is particularly noticeable in the case of equal sampling for all 39 classes (likely due to the up-sampling issue described above). While the training error was only 73.29%, the test error was 96.85%.