255 Assignment 2, Fall 2015

Crime Prediction in San Fransisco

Patrick Phaneuf
pphaneuf@eng.ucsd.edu

Dorothy Yen
doyen@eng.ucsd.edu

Sean Grady
sgrady@eng.ucsd.edu

1. IDENTIFYING A DATASET

In June 2015, Kaggle began a competition named "San Francisco Crime Classification"[8], ending in June 2016. The competition's dataset caught our attention due the subject being very tangible, with crime being at the forefront of modern media and to San Francisco being culturally significant due to its current tech industry. The dataset is also described by geographic and temporal features, therefore enabling potentially interesting visualizations. After our initial investigation of this dataset and the Kaggle competition, we realized that there was a large amount of accessible information on different ways of analyzing this very data set through blog posts and scripts published for this competition on Kaggle and Kaggle's forums. Through the nature of Kaggle competitions, a means of evaluation is also provided for by the competition rules. We chose this dataset for these reasons.

The "San Francisco Crime Classification" competition and its accompanying dataset, provided by SF OpenData, consists of 878,049 samples of crime reports from all neighborhoods of San Francisco spanning from January 2003 to May 2015. The data is initially split by Kaggle into two sets: the training and testing set. Odd numbered weeks (1, 3, 5, 7, ...) are put in the training set, and even numbered weeks (2, 4, 6, 8, ...) are put in the test set. The fields in each sample point are given in table 1.

Table 1: Training and testing data fields.

Training	Testing
Timestamp	Timestamp
Day of the Week	Day of the Week
Category	
Description	
Police Department District	Police Department District
Resolution	
Approximate Street Address	Approximate Street Address
Latitude/Longitude	Latitude/Longitude

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Our Initial exploratory efforts were focused on finding and visualizing the basic statistics for the dataset. The obvious categories to investigate first were the crime counts per police department district. We therefore created pie charts for total crime count per district (figure 1) and total crime count per category (figure 2).

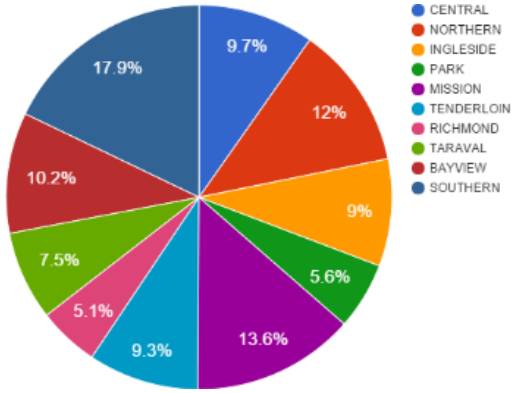

Figure 1: Percentage of total crimes per police district.

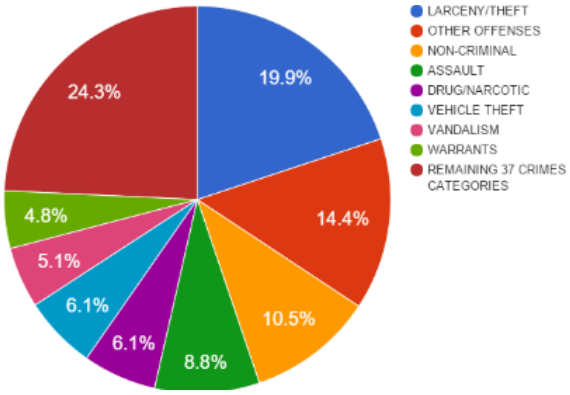

Figure 2: Percentage of total crimes per crime category.

We then leveraged the time features and explored crime total counts for police districts over time (figure 3) and total crime counts for crime categories over time, with some examples given in figure 4. Beyond the crime count rankings categories, this type of graph reveals some interesting temporal

trends. For example, Larceny/Theft have increased substantially since 2011, Drugs/Narcotics have decreased since 2009, Prostitution has decreased since 2007 and Secondary Codes have increased since 2005. The Recovered Vehicle and Vehicle Theft categories have large changes in their magnitude due to a shift in identifying a percentage of Vehicle Theft as Recovered Vehicle entries in 2006[15].

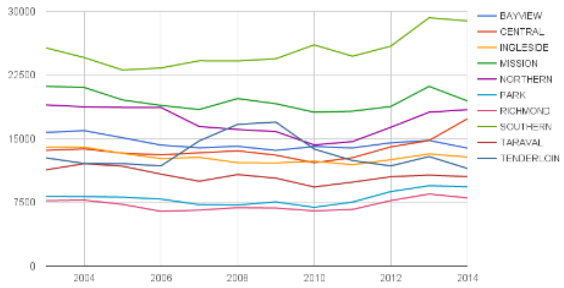

Figure 3: Crime counts for each police district over time.

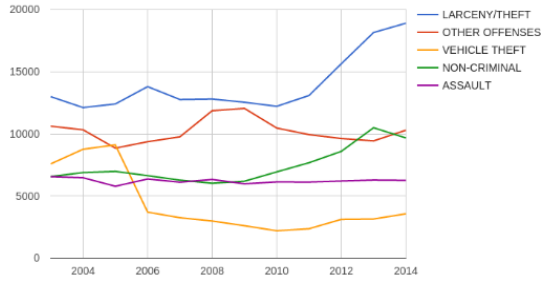

Figure 4: Crime counts for several of the most common categories over time.

These investigations enabled us to understand which crimes had the highest frequency, which districts had the most crimes and if there were any significant changes in crime categories over time.

We then sought to leverage the crime?s geographic information. After some initial efforts in plotting crimes on maps through Python we discovered and began using the data plotting web platform CartoDB[1]. With the ability to plot our data with CartoDB, we aimed to find interesting geographic trends with crime. CartoDB's density plots, which operate on what seems to be a logarithmic scale, immediately enabled us to identify crime hotspots and also revealed that many crimes share hotspots. These shared hotspots could reveal features describing how certain crimes categories often appeared near other crime categories. Table 2 shows the hotspots that a few of the most common crimes occur in. Figures 5 and 6 show examples of the density plots.

Due to the sparsity of certain crimes, time lapse visualizations were more telling. Figure 7 is a time lapse snapshots of the Prostitution crime category, which revealed hotspots that the density map didn't.

The time lapse functionality also had revealed the fact that crime hotspots had temporal trends. Certain crime hotspots existed for a few years and then no longer manifested and could possible migrate. This had been observed

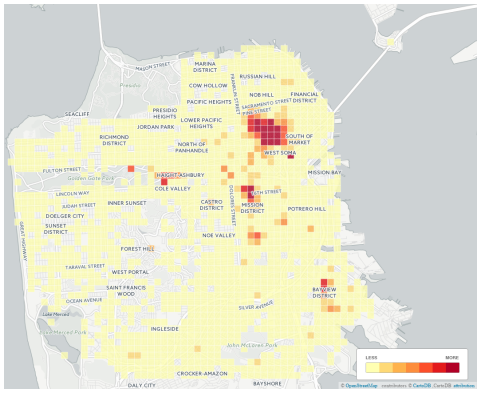

Figure 5: Density map of Drugs/Narcotics offenses.

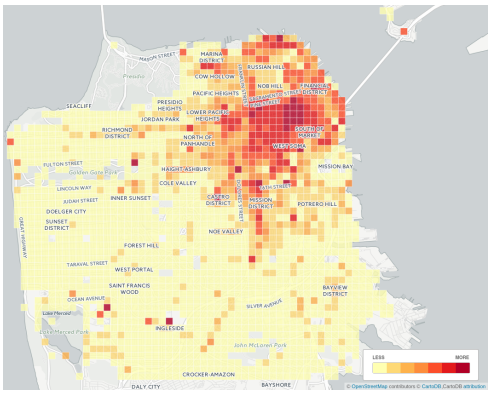

Figure 6: Density map of Larceny/Theft offences.

for the Suicide category on the Golden Gate bridge and Non Criminal category in Haight-Ashbury.

Using the map visualizations, we had also discovered that the Hall of Justice, located in West South of Market, was a hotspot for almost all crimes. There were 26,354 records reported at the Hall of Justice?s latitude/longitude coordinates. Most latitude/longitude coordinates only had one crime reported. The second most frequent coordinates after the Hall of Justice had less than 5,000 reports. We believe the significant difference in crimes reported at this location is an artifact of how crimes are reported and could therefore be avoided in training our model.

Through our data exploration we were able to describe some of its basic trends and additionally had discovered many interesting properties that we could leverage as features in our model. Exploring these trends and properties therefore lent us the intuition and inspiration for our feature design and model selection described in the following sections.

2. IDENTIFY A PREDICTIVE TASK

Kaggle provided a test dataset of >800,000 samples consisting of an ID, timestamp, day of the week, district, approximate street address, and the longitude/latitude of the crime. Each incident is labeled with exactly one category. The predictive task is to assign a probability to each incident for each class (category of crime). In order to evaluate our models, we use the multi-class logarithmic loss func-

Table 2: A collection of the most common crimes, with marks in the hotspots each crime shares.

	Assault	Drugs/Narcotics	Larceny/Theft	Robbery	Prostitution
Tenderloin/Union Square	x	x	x	x	x
Telegraph Hill	x				
Mission	x	x	x	x	x
Southeast McLaren Park	x				
Bayview	x	x			
Haight-Ashbury		x			
Financial			x		
SOMA			x		
Lower Pacific Heights			x		
Castro District			x		
Ingleside			x		
Embarcadero			x		
North Beach			x		
China Town			x		
Marina			x		

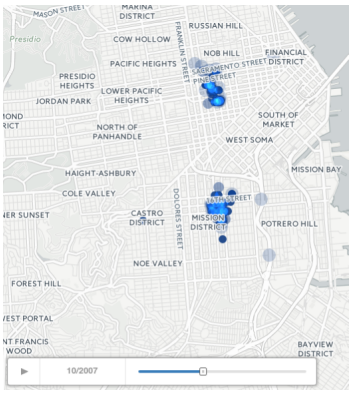

Figure 7: Snapshot from a time lapse of Prostitution offenses.

tion, given by equation 1. The submitted probabilities for a given incident are not required to sum to one because they are rescaled prior to being scored (each row is divided by the row sum). In order to avoid the extremes of the log function, predicted probabilities are replaced with max(min(p, $1-10^{15}$, 10^{-15})[16]

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

Where N is the number of cases in the test set, M is the number of class labels, log is the natural logarithm, y_{ij} is 1 if the observation is in class j and 0 otherwise and p_{ij} is the predicted probability that observation i belongs to class j.

Our intention with this project's predictive task is to therefore produce a model and a set of features that perform well according to the rules described in the Kaggle competition[8]. We additionally intend on participating in the competition by submitting our predictions. The username used in submitting to the competition is "dorothyy".

In order to establish a baseline, we predicted the probability for each category as the fraction of occurrences in the training set. This resulted in a score of 2.68016 on the testset.

We tested different models and features, comparing the accuracy and log loss performance to understand which rendered better results. We found that performance results varied with the random splits of training/validation sets. Therefore, we had to consider these measures of evaluation on the same train/validation set or use an average of several randomized splits.

The models and features we explored and how the data was preprocessed is discussed in the next section.

3. MODELS

We explored Bernoulli Naive Bayes, Gaussian Naive Bayes, Multinomial Naive Bayes, Linear SVM, K-Means and K-Neighbors Classifier. We found that Naives Bayes was critical due to its output being a probability. Additionally, Naive Bayes is one of the best choices for our data set because the conditional independence assumption (that the features are conditionally independent given the label) holds well for the chosen features. For example if the crime was categorized as assault, knowing that it happened at night gives me no additional information about whether it happened on a street corner or in the middle of a block, or on what day of the week it occurred.

Our final choice of classifier was a Bernoulli naive bayes model, where the probability of observing a particular feature vector $\mathbf{x}$ is given by

$$p(\mathbf{x}) = \frac{(\sum_i x_i)!}{\Pi_i x_i!}\Pi_i p_{ki}^{x_i} \quad (2)$$

and a probability is predicted for each label.

We optimized our model by choosing features which resulted in the best performance on a validation set we constructed at random from 25% of the training data. The Bernoulli Model and Multinomial Model shared most features but ultimately, the Bernoulli outperformed the Multinomial Model. Although the Multinomial Model allowed for non binary features, we found a significant improvement when treating such features as binary. For example, instead of one feature "Hour" with values [0-23], 24 separate features for [00:00, 01:00, . . . , 23:00] improved our predictions. The shared features between the two models include police district, whether the date is a holiday, day of the week, year,

hour, whether it's day or night, and the type of address (street corner or the middle of a block). An "isWeekend" feature and non-binary Month feature improved the Multinomial Model but did not improve and was not available to the Bernoulli Model.

The Multinomial Model utilized Laplace smoothing to handle the unlikely case in which a particular feature-label pair did not appear in the training set, for example, if 'assault' was never reported at 3pm in the test set but did appear in the training set at that time.

Because we used a variant of the naive Bayes (NB) model, we did not run into any scalability issues or overfitting. It was, however, necessary to choose features carefully to avoid correlated features and the 'double counting' problem. For example, using districts as well as X/Y location (two location measures) resulted in noticeably worse performance. Additionally, using X and Y as separate features (instead of some binary feature like districts) resulted in a less accurate model.

There were some features that improved our prediction earlier in the feature development stage but once we had stronger features, they actually hindered our predictions. An example of this is a feature for time of day (Midnight, Morning, Afternoon, Evening) or simply day or night. A naive attempt to incorporate crime category hotspots by district also did not improve performance. This is likely due to being too similar to the district feature that was already in the feature set.

In addition to considering each day of the week as an individual feature, we investigated the effect of weekends versus weekdays on crime categories. We did not find a significant improvement in our model when replacing individual day of week features with a weekend/weekday feature as well as when we added the weekend/weekday feature to the individual day of week features. This was contrary to what we had expected. On the other hand, crimes rates showed some correlation with holidays.3 The holidays that we explored included New Year's Day, Valentine's Day, Memorial Day, Independence Day, Halloween, Thanksgiving and the day after Thanksgiving, Christmas Eve/Day and New Year's Eve. The reason we expected to see this correlation is with increased home robberies as families take vacations, increased drunk driving incidents and domestic abuse with increased alcohol consumption, increased counts for identify theft, etc.

In addition to the selection of smart features, addressing two outliers also helped improve our model. The year proved to be a very useful feature, but because the dataset only included incidents up to May 2015, the feature "2015" was not as informative. Therefore, we treated all incidents in 2015 as a 2014 crime to benefit from the 2014 trends. The second outlier was the >20,000 incidents reported at the Hall of Justice. The way we optimized our model with consideration to this outlier is we removed the entries from the training set and when encountering the same coordinates in the test set, instead we predicted the average for each category that we found in the training set at those specific coordinates.

We also considered several other kinds of classifiers, including k-means clustering (k-means), k nearest neighbors (kNN), and a variety of support vector machines (SVM).

We explored using k-means because we felt that different types of crimes might be clustered in distinct "hotspots", but we found that while there are certainly clusters, most common crimes share the same hotspots, so most of the time cluster membership did not adequately differentiate between crimes.

We found that kNN worked very well, but only for extremely large (1000) values of k. For a dataset as large as ours, this makes the algorithm slow to run and not particularly scalable. We also found that NB methods outperform kNN when the features are chosen appropriately anyway.

All SVM classifiers are non-probabilistic, and given that perfect (or even near-perfect) classification accuracy is likely impossible (due to the number of crime categories in which different samples often share identical features), it seems useful to be able to give a probability distribution over the possible categories, rather than just a prediction and/or confidence. Law enforcement agencies would find it more helpful to have a list of the crimes likely to occur in a given area and some information regarding their relative probabilities than a simple classifier. In addition, the Kaggle competition necessitates the reporting of probabilities by model. These details lead us to ultimately avoid using an SVM as our classifier.

4. LITERATURE

The origin of the data, as mentioned by the Kaggle competition page[8], is from SF OpenData, an online resource for publishing the public data of the City and County of San Francisco. In doing a deeper search for the origins of the data, we found what seemed to be the most relevant publication of the data, named "SFPD Incidents"[10], which presents the data used in the Kaggle competition in a very similar format and includes graphics of crime density and frequency.

The "San Francisco Crime Classification" Kaggle competition page[8] serving as our starting point for this project, contained a wealth of knowledge on the given crime data set. Through the nature of Kaggle competitions, the open discussions and posting of scripts used to generate results recorded in the competition became the most inspiring aspect of the literature search. For this competition, Kaggle hosts a scripts page[11] that describes participant?s solution scripts and any other aspect of the data which the participants found interesting. Some interesting scripts page entries that overlapped the work we executed were top crimes[12], crime densities[13] and crime history[14]. Kaggle competition script pages could possibly be good resources for practical starting points and pattern investigation in future data mining tasks.

In our search on this competitions subject, we had come across a blog post[7] on this particular competition separate from the Kaggle website. This post provided a study with functional Python code on using a Bernoulli NB model from crime classification and served us as a boilerplate for preprocessing the data using Python Pandas[17] into appropriate classifier features and generating the necessary Kaggle submission file. The post also revealed the fact that the Python scikit-learn[9] package implemented many of the models we wanted to use and the log-loss function used in evaluating submissions. We therefore leverage scikit-learn for many of the models used in our results and with the log-loss function in our model and feature validation and leverage Pandas for building our feature vectors.

Crime analytics has been the focus for a substantial amount of academic articles. There exists much accessible academic literature on the subject, covering a large number of perspec-

tives and approaches for clarifying patterns in crime data. In our research, we identified an article titled "A Study on Classification Learning Algorithms to Predict Crime Status"[2], published in 2013, that provided a significant review of a similar crime classification task. The focus of this study was to categorize a crime as either being "critical" or "non-critical", describing the relative potential for violence of a crime. The models considered in this paper were:

1. Naive Bayes
2. Decision Trees
3. Support Vector Machines
4. Neural Network
5. k-Nearest Neighbor

While we had already considered and tested many of the models described in the paper, we took notice of the conclusion of the paper that described the best performing model as being the k-Nearest Neighbor and subsequently included this model in our development. This article also detailed the Chi-squared feature selection method that greatly benefited their model?s performance.

This article concludes that kNN in conjunction with Chi-squared feature selection rendered the best results for classifying crimes as "critical" or "non-critical". Their findings also described NB, in addition to kNN, had performed better than an SVM. We believe we had experienced similar results, though our model evaluation focused on the log loss, which is unobtainable from SVMs due to their inability to return probabilities for classification. We can however compare models according to their accuracy, simply described as the following:

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{total predictions}}$$

In comparing the accuracy results of models, we noticed that models in which had better Log Loss results didn't necessarily have better accuracy. Since our intention is to mostly operate within the bounds of the Kaggle competition's rules, we therefore chose to rely primarily on Log Loss for model evaluation and discontinued exploring solutions with SVM or leveraging Chi-squared for feature selection.

5. RESULTS

In conclusion, we explored Bernoulli Naive Bayes, Gaussian Naive Bayes, Multinomial Naive Bayes, Linear SVM, K-Means and K-Neighbors Classifier. Bernoulli and Multinomial Naive Bayes were very close in reducing the Log Loss and both were significantly better than the non-probabilistic models explored since the goal is to minimize multi-class Log Loss. Naive Bayes success is primarily due to the conditional independence assumption. Gaussian Naive Bayes performed poorly because the majority of the data is not gaussian distributed. K-Means did not perform well because most crimes were clustered in the northeastern corner of San Francisco and therefore were not differentiated well enough to be practically leveraged. kNN performed as well as our average/worst-case NB models but required large values of K which were extremely slow to run. SVM did not perform well because it is non-probabilistic and it is very difficult to obtain a perfect classification due to the number of crime categories that share identical features. Ultimately, using the validation set to obtain the best features, Bernoulli outperformed Multinomial Naive Bayes.

Tables 3 and 4 show the effects of individual features on each model's performance. It was helpful to include features based on PD District, day of week, hour of day, year, holidays, weekends, and address type. We were careful not to include features that conveyed the same information to avoid the double counting problem that arises with Naive Bayes. Examples of features that did not improve the model include PD District defined hotspots, day or night, and a looser definition of holiday. It was interesting to find that although Multinomial Bayes allowed for non-binary features, it performed better by creating such features such as year as a binary feature.

All in all, this dataset was extremely interesting, and provided a large number of avenues for exploration and classification. Additionally, the Kaggle competition aspect added extra motivation and fun to the task. With only ten days to create a model, we did not expect to place high in the leaderboard. As of December 1, 2015, our best score (Log Loss) on the test set is 2.51916 giving us a position of 197/884.

With more time, the following ideas would be worth experimenting with. While the police department districts were an extremely helpful feature in most classification methods, it seems likely that a different number of divisions with different boundaries could provide even better classification accuracy. Exploring the data further with an eye to obtaining an optimal set of "districts" could provide a substantial improvement.

Similarly, while we noted the existence of hotspots in our initial exploration of the data, we were not able to incorporate them as features in our models. With more time, adding membership in (or possibly proximity to) all, or a subset of, hotspots as features might be something worth attempting.

Finally, while we were able to test out many different models and classifiers, there are of course many more which could be tried. In particular, boosting and bagging as well as random forests, neural networks, and gradient boosting were all models that were reported to perform well on this data set.

6. REFERENCES

1. Map Your World's Data - CartoDB. (n.d.). Retrieved November 26, 2015, from https://cartodb.com/
2. Somayeh Shojaee, Aida Mustapha, Fatimah Sidi, Marzanah A. Jabar (May 2013). A Study on Classification Learning Algorithms to Predict Crime Status. International Journal of Digital Content Technology and its Applications(JDCTA), Volume7, Number9. Retrieved from http://www.researchgate.net/publication/266971832_A_Study_on_Classification_Learning_Algorithms_to_Predict_Crime_Status
3. The 10 Most Common Holiday Crimes. (n.d.). Retrieved November 28, 2015, from http://www.leelofland.com/wordpress/the-10-most-common-holiday-crimes/
4. http://scholarlycommons.law.northwestern.edu/cgi/ viewcontent.cgi?article=4004&context=jclc
5. http://cs.brown.edu/courses/csci2950-t/crime.pdf

6. http://www.sciencedirect.com/science/article /pii/S187705091200720X

7. Machine learning to predict San Francisco crime. (n.d.). Retrieved November 27, 2015, from http://efavdb.com/ predicting-san-francisco-crimes/

8. San Francisco Crime Classification. (2015, June 2). Retrieved November 23, 2015, from https://www.kaggle.com/c/sf-crime

9. Scikit-learn. (n.d.). Retrieved November 25, 2015, from http://scikit-learn.org/stable/

10. SFPD Incidents - from 1 January 2003. (n.d.). Retrieved November 30, 2015, from https://data.sfgov.org/view/vsk2-um2x

11. San Francisco Crime Classification. (n.d.). Retrieved November 26, 2015, from https://www.kaggle.com/c/sf-crime/scripts

12. Dashboard. (n.d.). Retrieved November 28, 2015, from https://www.kaggle.com/sanghan/sf-crime/top-crimes-zones

13. Dashboard. (n.d.). Retrieved November 28, 2015, from https://www.kaggle.com/dbennett/sf-crime/test-map

14. Dashboard. (n.d.). Retrieved November 26, 2015, from https://www.kaggle.com/swbevan/sf-crime/a-history-of-crime-python

15. Dashboard. (n.d.). Retrieved November 26, 2015, from https://www.kaggle.com/eyecjay/sf-crime/vehicle-thefts-or-jerry-rice-jubilation

16. Dashboard. (n.d.). Retrieved December 1, 2015, from https://www.kaggle.com/c/sf-crime/details/evaluation

17. Dashboard. (n.d.). Retrieved December 1, 2015, from http://pandas.pydata.org/

Table 3: The effect of different feature sets on the final model (Bernoulli naive Bayes).

Feature Set	Log Loss	Accuracy	significance
Bernoulli Naive Bayes			
PD Districts	2.61543823076	0.220689072157	
PD Districts, *Address Type*	2.57023087076	0.224629445248	Address Type helps
PD Districts, Address Type, *years*	2.55220918076	0.228320903233	Years help
PD Districts, Address Type, years *DayOfWeek*	2.55186114348	0.229161578779	
PD Districts, Address Type, years DayOfWeek, *Month*	2.54861233646	0.229462155511	Month does not help
PD Districts, Address Type, years DayOfWeek, *Darkness*	2.53737434045	0.229650015968	Darkness does not help
PD Districts, Address Type, years DayOfWeek, *HourOfDay*	2.52321378222	0.233487065808	
PD Districts, Address Type, years DayOfWeek HourOfDay, *isHoliday*	2.52321378222	0.233487065808	
PD Districts, Address Type, years DayOfWeek, HourOfDay, isHoliday *isWeekend*	2.52301611859	0.233585692548	
Final Model PD Districts, Address Type Years, DayOfWeek HourOfDay, isHoliday Multinmomial Additional Features: Month (Numerical), isWeekend	2.51816347275	0.235342187823	This is our final solution, which we uploaded to Kaggle
Final Model with separate 2015	2.52282772808	0.234083522759	
Multinomial Naive Bayes			
PD Districts	2.61404419075	0.220689072157	
PD Districts, *Address Type*	2.57619988016	0.223286242979	Address Type helps
PD Districts, Address Type, *years*	2.55508999474	0.227738535816	Years help
PD Districts, Address Type, years *DayOfWeek*	2.55423131592	0.228710713682	
PD Districts, Address Type, years DayOfWeek, *Month*	2.55008251841	0.228405440439	Month does not help
PD Districts, Address Type, years DayOfWeek, *Darkness*	2.54355289996	0.229180364825	Darkness does not help
PD Districts, Address Type, years DayOfWeek, *HourOfDay*	2.54716089449	0.229715767128	
PD Districts, Address Type, years DayOfWeek HourOfDay, *isHoliday*	2.5239484368	0.232369296087	
PD Districts, Address Type, years DayOfWeek, HourOfDay, isHoliday *isWeekend*	2.52231815745	0.23310195187	
Final Model PD Districts, Address Type Years, DayOfWeek HourOfDay, isHoliday Multinmomial Additional Features: Month (Numerical), isWeekend	2.51939401606	0.234482726231	
Final Model with separate 2015	2.52377678124	0.234186846011	

Table 4: Results for Gaussian naive Bayes, Linear SVM and k Nearest Neighbors

Model	**Feature(s)**	**LogLoss**	**accuracy**
GaussianNB	Districts	20.0957881845325	0.0117874836285
	X, Y	3.45880769879072	0.0879135584534
	Districts, X, Y	23.9067918128639	0.000612151927567
Linear SVM	Districts	-	0.2213370537
	X, Y	-	0.1399550139
kNN	X, Y, neighbors = 40	5.76697011605	0.27207448323
	X, Y, neighbors = 50	5.10655117779	0.273489550709
	X, Y, neighbors = 100	3.82699577849v	0.272424691077
	X, Y, neighbors = 1000	2.59118415	-
	Districts, neighbors = 40	6.41508192101	0.214668868515
	Districts, neighbors = 50	5.918119433	-