Crime Prediction and Classification in San Francisco City

Addarsh Chandrasekar, Abhilash Sunder Raj and Poorna Kumar

***Abstract*— To be better prepared to respond to criminal activity, it is important to understand patterns in crime. In our project, we analyze crime data from the city of San Francisco, drawn from a publicly available dataset. At the outset, the task is to predict which category of crime is most likely to occur given a time and place in San Francisco. To overcome the limitations imposed by our limited set of features, we enrich our data by adding information from the United States Census to it. We also attempt to make our classification task more meaningful by merging multiple classes into larger classes. Finally, we report and reflect on our results with different classifiers, and dwell on avenues for future work.**

I. Introduction

Many important questions in public safety and protection relate to crime, and a better understanding of crime is beneficial in multiple ways: it can lead to targeted and sensitive practices by law enforcement authorities to mitigate crime, and more concerted efforts by citizens and authorities to create healthy neighborhood environments. With the advent of the Big Data era and the availability of fast, efficient algorithms for data analysis, understanding patterns in crime from data is an active and growing field of research.

In our project, we use spatio-temporal and demographic data to predict which category of crime is most likely to have occurred, given a time, place and the demographics of the place. The inputs to our algorithms are time (hour, day, month, year), place (latitude, longitude, and police district), and demographic data (population, median income, minority population, and number of families, which we get from the United States Census). The output is the category of crime that is likely to have occurred. We try out multiple classification algorithms, such as Naive Bayes, Support Vector Machines, Gradient Boosted Decision Trees, and Random Forests. We also perform multiple classification tasks – we first try to predict which of 39 classes of crimes are likely to have occurred, and later try to differentiate between blue- and white-collar crimes, as well as violent and non-violent crimes.

II. Related Work

Much of the current work is focused in two major directions: (i) predicting surges and hotspots of crime, and (ii) understanding patterns of criminal behavior that could help in solving criminal investigations.

Important contributions towards the former include [1] by Bogomolov et al, who try to predict whether any particular area in London will be a crime hotspot or not, using anonymized behavioural data from mobile networks as well as demographic data. In [2], Chung-Hsien Yu et al use classification techniques to classify neighbourhoods in a city as hotspots of residential burglary, using a variety of classification algorithms such as Support Vector Machines, Naive Bayes, and Neural Networks. (More work on the usefulness of Support Vector Machines for hotspot detection can be found in [3]). Toole et al demonstrated in [4], by analyzing crime records for the city of Philadelphia, that significant spatio-temporal correlations exist in crime data, and they were able to identify clusters of neighbourhoods whose crime rates were affected simultaneously by external forces. They also noted significant correlations in crime across weekly time scales.

Towards the second objective of understanding patterns of criminal behaviour, significant contributions have been made by Tong Wang et al in [5], in finding patterns in criminal activity and identifying individuals or groups of individuals who might have committed particular crimes. Their approach was to identify a common modus operandi across crimes, which could then be linked to groups or individuals who might commit the crime. For this, the authors proposed a new machine learning method called Series Finder, which was trained to recognize patterns in housebreak incidents in Cambridge, Massachusetts.

Our approach shares certain similarities with some of the work described above, in that we use spatio-temporal and demographic information to discover which types of crimes are likely to have occurred. However, we are notably different in that, given the data, we seek to predict which category of crime is most likely to occur, and we are hence concerned principally with understanding the differences between different types of crime, which is relatively unexplored territory.

III. Our Dataset

Our dataset is a publicly available dataset that we obtained from Kaggle, which has information about 878,049 crimes that took place in San Francisco city over a span of nearly twelve years. Each crime is labeled as belonging to one of 39 categories.

A. Features

Every entry in our training data set is about a particular crime, and contains the following information:

- **Date** and **timestamp** of the incident.
- **Day of the week** that the crime occurred.
- Name of the **Police Department District**.
- **Address**: the approximate street address of the crime incident.
- **Latitude**.
- **Longitude**.
- **Category**: category of the crime incident. This is the target variable.
- **Description**: a brief note describing any pertinent details of the crime. (This was not used as a feature in our classifiers.)
- **Resolution**: whether the crime was resolved (with the perpetrator being, say, arrested or booked) or not. (This was also not used as a feature in our classifiers.)

TABLE I: Some sample rows from our dataset

Dates	Category	Descript	DayOfWeek	PdDistrict
13-05-2015 23:53	WARRANTS	WARRANTS	Wednesday	NORTHERN
13-05-2015 23:53	OTHER OFFENSES	TRAFFIC VIOLATION	Wednesday	NORTHERN
13-05-2015 23:33	OTHER OFFENSES	TRAFFIC VIOLATION	Wednesday	NORTHERN
13-05-2015 23:30	LARCENY	GRAND THEFT	Wednesday	NORTHERN

Resolution	Address	X	Y
ARREST, BOOKED	OAK ST / LAGUNA ST	-122.4258917	37.7745986
ARREST, BOOKED	OAK ST / LAGUNA ST	-122.4258917	37.7745986
ARREST, BOOKED	VANNESS AV / GREENWICH ST	-122.424363	37.80041432
NONE	1500 Block of LOMBARD ST	-122.4269953	37.80087263

B. Preprocessing

Before implementing machine learning algorithms on our data, we went through a series of preprocessing steps with our classification task in mind. These included:

- Dropping features such as Resolution, Description and Address: The resolution and description of a crime are only known once the crime has occurred, and have limited significance in a practical, real-world scenario where one is trying to predict what kind of crime has occurred, and so, these were omitted. The address was dropped because we had information about the latitude and longitude, and, in that context, the address did not add much marginal value.
- The days of the week, police categories and crime categories were indexed and replaced by numbers.
- The timestamp contained the year, date and time of occurrence of each crime. This was decomposed into five features: Year (2003-2015), Month (1-12), Date (1-31), Hour (0-23) and Minute (0-59).

Following these preprocessing steps, we ran some out-of-the-box learning algorithms as a part of our initial exploratory steps. Our new feature set consisted of 7 features, all of which were now numeric in nature.

C. Feature Enrichment

As we plunged into solving our classification problem, we felt that our feature set was not adequate enough in terms of the information it contained to predict crime. In order to improve our feature set, we augmented our dataset with additional features that we scraped from the United States Census data. This included demographic data such as mean income level of a neighbourhood, racial diversity and so on. We felt that the addition of such information could improve our performance at the task of crime prediction. The census dataset was matched with our dataset using the location coordinates of the crime in our original dataset, and increased our number of features to 19.

D. Collapsing Crime Categories

We also felt that the number of output classification labels, i.e. 39, in the original dataset was too high for accurate prediction. The labels were too fine-grained, and we realized that several of these crime categories were similar to one another, and could therefore be collapsed into smaller classes for better prediction. Further, such collapsing could be done in several ways. Two specific ways we went ahead with this were:

- Blue Collar Crimes vs White Collar Crimes: Blue Collar Crimes included crimes such as Larceny, Arson and Burglary while White Collar crimes included crimes such as Fraud, Forgery and Extortion.
- Violent vs Non-Violent Crimes: Violent crimes included crimes such as Assault, Arson and Prostitution while Non-Violent crimes included crimes such as Traffic Violations and Trespassing.

IV. Methods

After the preprocessing described in the previous sections, we had three different classifications problems to solve, which we proceeded to attack with an assortment of classification algorithms. The following sections explain the models we used in detail.

TABLE II: Sample Rows From The Census Data.
Note: The column headers are as follows. TC: Tract Code, TIL: Tract Income Level, TMFI: Tract Median Family Income, MFI: Median Family Income, OOU: Owner Occupied Units, FU: Family Units, Min.: Minority

TC	TIL	Distressed	TMFI %	MFI	2015 TMFI
101.00	Moderate	No	69.03	$96,900	$66,890
102.00	Upper	No	154.01	$96,900	$1,49,236
103.00	Upper	No	136.32	$96,900	$1,32,094

2010 MFI	Tract Pop.	Tract Min. %	Min. Pop.	OOU	1- to 4- FU
$64,886	3739	53.92	2016	260	424
$1,44,750	4143	18.95	785	943	918
$1,28,125	3852	39.25	1512	505	1282

A. Naive Bayes

As part of our initial exploratory analysis, we implemented a Naive Bayes classifier based on a **multi-variate event model** with **Laplace smoothing**. This is a multi-class classification problem: the target variable Y (crime category) can be one of 39 classes, represented by numbers from 1 to 39. Therefore, ϕ_y was modeled as a multinomial distribution.

$$Y \in \{1, 2, \ldots, 39\} \tag{1}$$

$$Y \sim \phi_y(\text{Multinomial}) \tag{2}$$

The latitude and longitude data were not used for classification, and all the remaining features are categorical variables. Thus, our feature vector, X, is a 7-dimensional vector. Each of the features takes a range of values: concretely, Month $\in \{1, 2, \ldots, 12\}$, Day of Week $\in \{1, 2, \ldots, 7\}$, and so on. Therefore, each feature is modeled by a multinomial distribution:

$$X_i \in \{1, 2, \ldots, k_i\} \tag{3}$$

$$X_i | \{Y = j\} \sim \phi_{i|y=j}(\text{Multinomial}) \tag{4}$$

Assuming that there are m training examples, the parameters $\{\phi_y, \phi_{i|y=j}\}$ are estimated using the following (Laplace-smoothed) equations:

$$\phi_y(j) = P\{Y = j\} = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = j\} + 1}{m + 39} \tag{5}$$

$$\phi_{j|y=l}(k) = P\{X_j = k | Y = l\} \tag{6}$$

$$= \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = k \wedge y^{(i)} = l\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = l\} + k_j} \tag{7}$$

B. Random Forests

Random Forests is a very popular ensemble learning method which builds a number of classifiers on the training data and combines all their outputs to make the best predictions on the test data. Thus, the Random Forests algorithm is a variance minimizing algorithm that uses randomness when making split decision to help avoid overfitting on the training data.

A random forests classifier is an ensemble classifier, which aggregates a family of classifiers $h(x|\theta_1), h(x|\theta_2), ..h(x|\theta_k)$. Each member of the family, $h(x|\theta)$, is a classification tree and k is the number of trees chosen from a model random vector.

Also, each θ_k is a randomly chosen parameter vector. If $D(x, y)$ denotes the training dataset, each classification tree in the ensemble is built using a different subset $D_{\theta_k}(x, y) \subset D(x, y)$ of the training dataset. Thus, $h(x|\theta_k)$ is the k^{th} classification tree which uses a subset of features $x_{\theta_k} \subset x$ to build a classification model. Each tree then works like regular decision trees: it partitions the data based on the value of a particular feature (which is selected randomly from the subset), until the data is fully partitioned, or the maximum allowed depth is reached.

The final output y is obtained by aggregating the results thus:

$$y = \text{argmax}_{p \in \{h(x_1)..h(x_k)\}} \{ \sum_{j=1}^{k} (I(h(x|\theta_j) = p)) \} \quad (8)$$

where I denotes the indicator function.

C. Support Vector Machines

We used Support Vector Machines for binary classification in the latter part of the project, where we worked on the classification problems with collapsed categories. We ran SVMs using the Gaussian (RBF) kernel to map the original features to a high-dimensional feature space:

$$K(x, z) = \exp\left(-\frac{||x - z||^2}{2\sigma^2}\right) \quad (9)$$

The optimal margin classifier with l_1 regularization was used.

$$\min_{\gamma,w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i \quad (10)$$

$$\text{s.t} \quad y^i(w^T \phi(x^i) + b) \geq 1 - \xi_i \quad (11)$$

$$\xi_i \geq 0, i = 1 \ldots m \quad (12)$$

D. Gradient Boosted Decision Trees

Gradient Tree Boosting[6] is another popular ensemble method used for regression and classification. Given a training sample $(\mathbf{x}, y)$, the goal is to find a function $F^*(\mathbf{x})$ that maps $\mathbf{x}$ to y such that the expected value of some loss function $\Psi(y, F(\mathbf{x}))$ is minimized. Boosting approximates $F^*(\mathbf{x})$ by the following equation:

$$F(\mathbf{x}) = \sum_{m=0}^{M} \beta_m h(\mathbf{x}; \mathbf{a}_m) \quad (13)$$

where the functions $h(\mathbf{x}; \mathbf{a}_m)$ (called "base learners") are simple functions of $\mathbf{x}$ with parameters $\mathbf{a}$. Starting with $F_0(\mathbf{x})$, the parameters β_m and $\mathbf{a}_m$ are found in a "stage-wise" manner and the function $F_m(\mathbf{x})$ is updated as:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \beta_m h(\mathbf{x}; \mathbf{a}_m) \quad (14)$$

In tree boosting, the base learner $h(\mathbf{x}; \mathbf{a})$ is an L-terminal node regression tree. At each iteration m, a regression tree partitions the $\mathbf{x}$-space into L-disjoint regions $\{R_{lm}\}_{l=1}^{L}$and predicts a separate constant value in each one. The update rules for calculating $F_m(\mathbf{x})$ given $F_{m-1}(\mathbf{x})$ are as follows:

$$\tilde{y}_{im} = -\left[\frac{d\Psi(y_i, F(\mathbf{x}_i))}{dF(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})} \quad (15)$$

$$\bar{y}_{lm} = \text{mean}_{\mathbf{x}_i \in R_{lm}} (\tilde{y}_{im}) \quad (16)$$

$$h(\mathbf{x}; \{R_{lm}\}_1^L) = \sum_{l=1}^{L} \bar{y}_{lm} . 1(\mathbf{x} \in R_{lm}) \quad (17)$$

$$\gamma_{lm} = \text{argmin}_{\gamma} \sum_{\mathbf{x}_i \in R_{lm}} \Psi(y_i, F_{m-1}(\mathbf{x}_i) + \gamma) \quad (18)$$

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu . \gamma_{lm} 1(\mathbf{x} \in R_{lm}) \quad (19)$$

V. Experimental Results

In this section, we detail the results of running the classifiers we described in the previous section on our data, on both the full dataset, and on collapsed categories.

A. Performance on our Original Dataset

With our original dataset, we ran two different learning algorithms, i.e, Naive Bayes and Random Forests classifiers, to get an initial understanding of the quality of our feature set, and the amount of predictability in the data.

The Naive Bayes Model was tested using cross validation, i.e, 70 percent of the data was used for training and the rest for testing purposes. We got the following results:

- The classifier gave 30% accuracy on the training set and 25% accuracy on the cross validation set. Hence, both training and cross validation error were very high.
- The above trend was observed even on varying the size of the training set. The accuracy did not go above 30%, even on the training set. Note that this is still significantly better than random guessing, since we have not 2, but 39, output classes.

We also implemented Random Forests for our classification problem. This was done keeping in mind that most machine learning algorithms work with numerical features and that our features are almost all categorical in nature. The Random Forests classifier works well with categorical features and does not need any preprocessing. We got the following results:

- After building the model, we ran it on the training set itself to get a training error of 5%, which initially looked too good to be true.
- However, on performing cross validation on the training data using 10 folds, we got a test error of 84%, which was huge compared to our training error, indicating high variance.

B. Performance on Collapsed Classes:

After the feature enriching process detailed above, where we augmented our training data with data from the Census, we ran 3 different algorithms on two separate classifications of the data. We split our crime categories into Blue Collar/White Collar Crimes in one case and Violent/Non-Violent Crimes in the other. Since blue-collar crimes far outnumbered white-collar crimes, and non-violent crimes far exceeded violent crimes, we decided to duplicate the minority class while training our classifier. This penalized the classifier more for mislabeling a training example in the minority class, than for mislabeling a training example in the majority class, during the training phase, and over-rode the tendency of the classifier to minimize its error by simply labeling all data as belonging to the majority class. By doing this, we were able to improve our precision and recall values on the minority class. Using this classification, we ran the following algorithms:

Random Forests:
For random forests algorithms, the parameters to tune are the number of trees and the maximum depth of each tree. In order

to pick optimum values for these, we tested the algorithm on the data for different combinations of the parameters. Finally, we picked the set of parameters that gave not only the overall highest accuracy but also the highest precision and recall values for both crime classes. The graphs below show the variation of accuracy, precision and recall values for parameter values for Blue/White Crime classification. From this, the maximum accuracy obtained was **79.18**% for number of trees = **200** and maximum depth = **15**. Thus, random forests worked fairly well on this problem, especially for blue collar crimes.

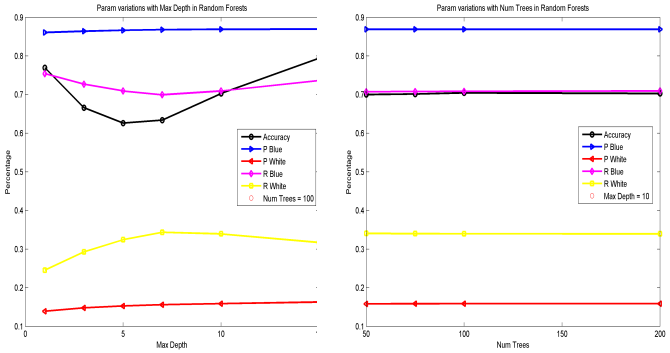


TABLE III: Precision and recall for random forests on Blue Collar/White Collar crime classification

Precision_Blue	Precision_White	Recall_Blue	Recall_White
0.869700021	0.163540533	0.741358669	0.312845342

The graphs below show the variation of accuracy, precision and recall values for parameter values for Violent/Non-Violent Crime classification. From this, the maximum accuracy obtained was **61.75**% for number of trees = **200** and maximum depth = **15**.

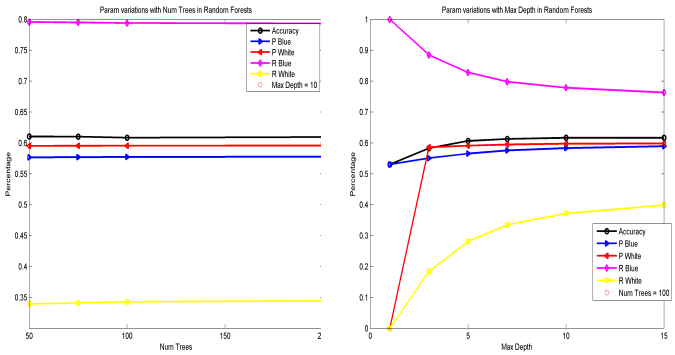


TABLE IV: Precision and recall for random forests on Violent/Non-Violent crime classification

Precision_Violent	Recall_Violent	Precision_Non_Violent	Recall_Non_Violent
0.589660627	0.598377214	0.762333658	0.400292642

Gradient Boosted Decision Trees:
For the gradient boosted trees algorithm, the parameters of interest are the number of trees and the maximum depth of each tree. Once again, we tested the algorithm on the data for different permutations of the parameters. Finally, we picked the set of parameters that gave not only the overall highest accuracy but also the highest precision and recall values for both crime classes for Blue/White Crime classification. The graphs below show the variation of accuracy, precision and recall values for parameter values. From this, the maximum accuracy obtained was **96.3**% for number of estimators = **200** and maximum depth = **13**.

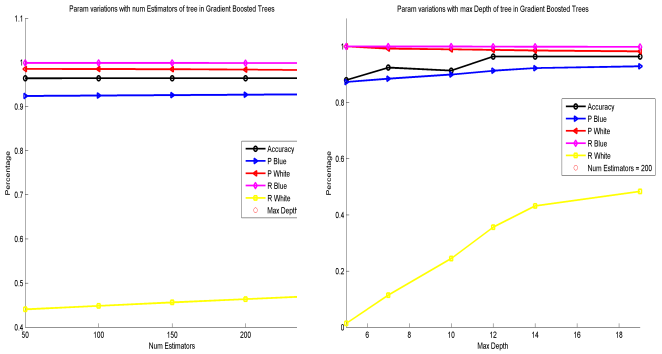


TABLE V: Precision and recall for gradient boosted trees on Blue Collar/White Collar crime classification

Precision_Blue	Precision_White	Recall_Blue	Recall_White
0.963603527	0.971364318	0.996785378	0.743345571

The graphs below show the variation of accuracy, precision and recall values for various configurations of parameter values for Violent/Non-Violent Crime classification. From this, the maximum accuracy obtained was **75.02**% for number of estimators = **200** and maximum depth = **11**.

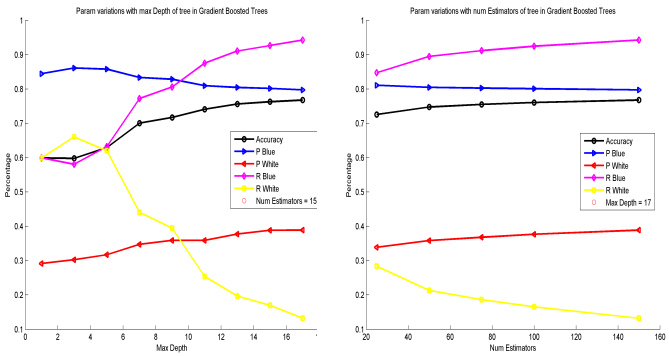


TABLE VI: Precision and recall for gradient boosted trees on Violent/Non-Violent crime classification

Precision_Violent	Recall_Violent	Precision_Non_Violent	Recall_Non_Violent
0.804631244	0.899938198	0.362965695	0.206925557

Support Vector Machines:
We used Support Vector Machine classifier with an "RBF" kernel as our final algorithm. The two parameters of interest for the classifier are the c and γ values. Once again, we tested the algorithm on the data for different permutations of the parameters and we picked the set of parameters that gave the highest accuracy. The graphs below show the variation of accuracy, precision and recall values for parameter values for Blue/White Crime Classification. From this, the maximum accuracy obtained was **96**% for $c = 4$ and $\gamma = 0.2$.
The graphs below show the variation of accuracy, precision and recall values for parameter values for Violent/Non-Violent Crime classification. From this, the maximum accuracy obtained was **62.80**% for c = **1.1** and γ = **0.01**.

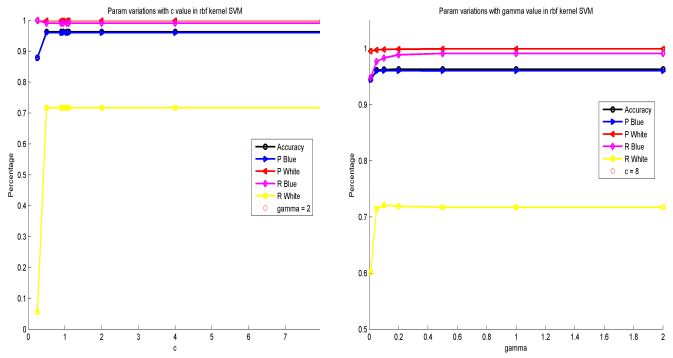


TABLE VII: Precision and recall for SVMs on Blue Collar/White Collar crime classification

Precision_Blue	Precision_White	Recall_Blue	Recall_White
0.9602589	0.998788205	0.988629185	0.718219367

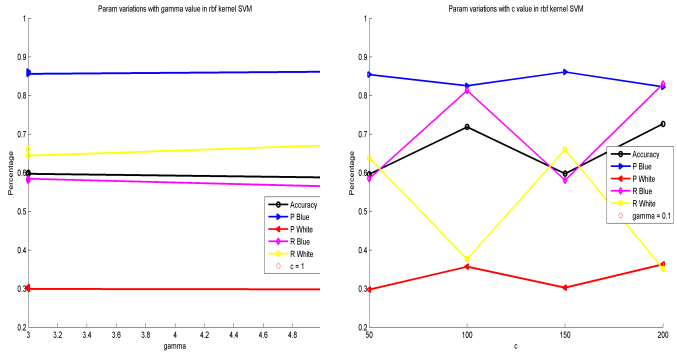
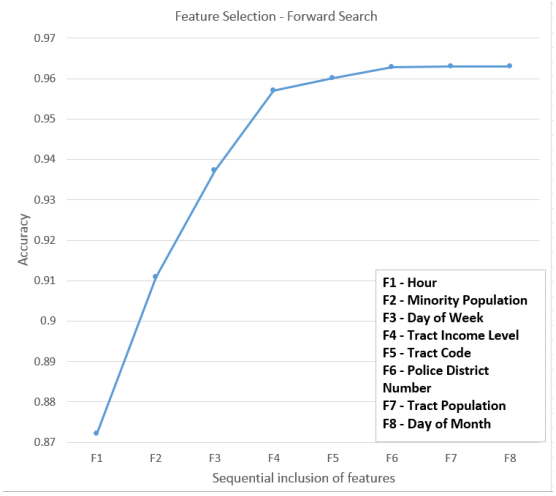


TABLE VIII: Precision and recall for SVMs on Violent/Non-Violent crime classification

Precision_Violent	Recall_Violent	Precision_Non_Violent	Recall_Non_Violent
0.823943997	0.668314728	0.285785578	0.481704753

C. *Feature selection*



More important than just our accuracy and precision is the interpretation of our model: which of our features actually help predict the category of crime? Analyzing the feature importances of our model gave us interesting insights.The following are the most relevant features used by our SVM (which we identified by running forward search) for classification, for the case of Blue Collar/White Collar Crimes: Hour, Minority Population, Day Of Week, Tract Income Level, Tract Code, Police District Number, Tract Population, Day of Month

VI. Conclusion and Future Work

The initial problem of classifying 39 different crime categories was a challenging multi-class classification problem, and there was not enough predictability in our initial data-set to obtain very high accuracy on it. We found that a more meaningful approach was to collapse the crime categories into fewer, larger groups, in order to find structure in the data. We got high accuracy and precision on the blue-collar/white-collar crime classification problem using Gradient Boosted trees and Support Vector Machines (the former famously robust and the latter well-suited to a 2-class classification problem, especially with an RBF Kernel that can translate the data to a high-dimensional space where it is linearly separable). However, the Violent/Non-violent crime classification did not yield remarkable results with the same classifiers – this was a significantly harder classification problem. Thus, collapsing crime categories is not an obvious task and requires careful choice and consideration.

Possible avenues through which to extend this work include time-series modeling of the data to understand temporal correlations in it, which can then be used to predict surges in different categories of crime. It would also be interesting to explore relationships between surges in different categories of crimes – for example, it could be the case that two or more classes of crimes surge and sink together, which would be an interesting relationship to uncover. Other areas to work on include implementing a more accurate multi-class classifier, and exploring better ways to visualize our results.

Acknowledgments

We would like to thank our project TA Youssef Ahrez for his thoughtful feedback and helpful ideas at every stage of our project. We also owe a debt of gratitude to Viswajith Venugopal, a student of the Department of Computer Science, for helping us to understand and implement parallel computing. And of course, our acknowledgements section would be incomplete without a mention of Professor Andrew Ng, whose excellent class enabled us to do this project in the first place.

References

[1] Bogomolov, Andrey and Lepri, Bruno and Staiano, Jacopo and Oliver, Nuria and Pianesi, Fabio and Pentland, Alex.2014. Once upon a crime: Towards crime prediction from demographics and mobile data, Proceedings of the 16th International Conference on Multimodal Interaction.

[2] Yu, Chung-Hsien and Ward, Max W and Morabito, Melissa and Ding, Wei.2011. Crime forecasting using data mining techniques, pages 779-786, IEEE 11th International Conference on Data Mining Workshops (ICDMW)

[3] Kianmehr, Keivan and Alhajj, Reda. 2008. Effectiveness of support vector machine for crime hot-spots prediction, pages 433-458, Applied Artificial Intelligence, volume 22, number 5.

[4] Toole, Jameson L and Eagle, Nathan and Plotkin, Joshua B. 2011 (TIST), volume 2, number 4, pages 38, ACM Transactions on Intelligent Systems and Technology

[5] Wang, Tong and Rudin, Cynthia and Wagner, Daniel and Sevieri, Rich. 2013. pages 515-530, Machine Learning and Knowledge Discovery in Databases

[6] Friedman, Jerome H. "Stochastic gradient boosting." Computational Statistics and Data Analysis 38.4 (2002): 367-378.sts

[7]Leo Breiman, Random Forests, Machine Learning, 2001, Volume 45, Number 1, Page 5