San Francisco Crime Classification

Veena Mounika Ruddarraju
Computer Science
California State University
Sacramento, CA, USA
vruddarraju@csus.edu

Swetha
Computer Science
California State University
Sacramento, CA, USA
syarraguntla@csus.edu

ABSTRACT

Classification of crime in different localities will bolster the efficiency of law enforcement by helping them to plan their resources and thereby reducing the crime. In this project, we intend to classify a crime category given the time and geographical location in San Francisco city. We designed this crime classification system in three phases. In the first phase, we built a model for a multi classification problem. In the second phase, we tried to improve accuracy and F1 score by converting this multi classification problem to binary classification problem. All the 39 crime categories in the dataset are categorized into 2 categories as White collar and Blue-collar crimes. In the third phase, we tried to improve our model accuracy by adding additional information that was extracted from US Census data. We used Yehya Abouelnaga's crime classification paper [2] as our baseline model and we go better results than the baseline model in our phase 1.

CCS CONCEPTS

Computing methodologies, Machine Learning and Deep neural networks.

KEYWORDS

Crime Classification, Random Forest, Deep Neural Networks, Naïve Bayes, Support Vector Machines.

1. INTRODUCTION

Crime is a major factor that determines the quality of life in an area. Many Cities in United States have signed the Open Data Initiative, thereby making crime data accessible to general public. Opening government data increases citizen participation in decision making by utilizing this data to uncover useful information out of this data [1]. Crime classification can improve law enforcements effort in mitigating crime by planning their resources efficiently and deploy the right officers when and where they are most needed. In the era of big data, with the help of efficient algorithms for data analysis, crime analysis is active and growing field of research.

In this project, we worked on crime occurrence data in San Francisco from 2003 to 2015 [3]. The main goal is to classify the crime based on time and location of the crime. We tried random forest decision tree, Naïve Bayes, Support Vector Machines, Deep Neural Networks, Logistic regression and XBG classifier classification algorithms.

This paper is organized as follows. Section 2 describes our model and our approach to the crime classification problem. Section 3 describes the various algorithms used in this project and Section 4 discusses the dataset, data preprocessing and the results obtained. Section 5 outlines the related work in the industry on San Francisco crime classification and section 6 has the conclusion and section 7 describes the work division and section 8 has learning experience.

2. PROBLEM FORMULATION

The goal of the project is to classify the crime (39 categories) based on time and location. We worked on this problem in three phases. In the first phase, the system used X(Longitude), Y(Latitude), year, month, day, hour, Day of the week, PdDistrict as inputs and predicted the category of the crime (Output). In second phase, we improved our model by classifying the 39 crimes categories to 2 types of crimes such as White collar and Blue collar. We used the same inputs for the system, and it predicted the crime as 0 or 1. 0 represents white collar crimes and 1 represents blue collar crimes. In phase 3, We further enriched out data by using US census data. We added multiple features like unemployment rate, population, poverty level, mean family income and minority data to improve the efficiency of the model

3. SYSTEM DESIGN

3.1 Deep Feedforward Networks

Deep Feedforward networks also often called feedforward neural networks, or multilayer perceptrons (MLPs), are the quintessential deep learning models. These have 3 or more hidden layers.

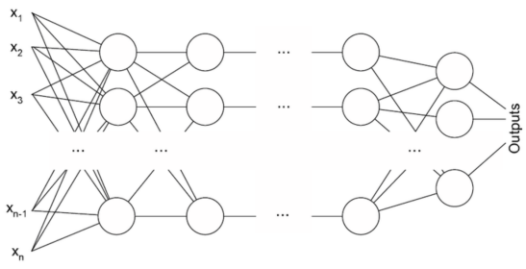


Fig 1: Perceptron [6]

The goal of a feedforward network is to approximate some function f*. These models are called feedforward because information flows through the function being evaluated from input x, through the intermediate computations used to define f, and finally to the output y. There are no feedback connections in which outputs of the model are fed back into itself. When feedforward neural networks are extended to include feedback connections, they are called Recurrent Neural Networks (RNN) .

3.2 Random Forests

Random forests are a very popular ensemble learning method which builds several classifiers on the training data and combines all their outputs to make the best predictions on the test data. Random Forests algorithm is a variance minimizing algorithm that uses randomness when making split decision to help avoid overfitting on the training data. If D(x, y) denotes the training dataset, each classification tree in the ensemble is built using a different subset Dθk (x, y) ⊂ D(x, y) of the training dataset. The final output y is obtained by aggregating the results:

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

3.3 K-Nearest Neighbors (KNN)

The KNN algorithm is supervised machine learning algorithm that can be used to solve both classification and regression problems. It assumes that similar things exist in proximity. We used different K Values and computed and tabulated their respective logloss, Accuracy and F1 score values in Table 1,2.

3.4 Naïve 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, φy was modeled as a multinomial distribution.

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

$$\phi_{j|y=l}(k) = P\{X_j = k | Y = l\} = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = k \wedge y^{(i)} = l\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = l\} + k_j}$$

3.5 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.

4.METHODOLOGY

4.1. Data set and data preprocessing:

4.1.1 Dataset

The dataset used for San Francisco Crime Classification is obtained from Kaggle [3]. The dataset contains incidents from SFPD Crime Incident Reporting system. The data is in between the time period of 1/1/2003 to 5/13/2015. The dataset has 9 features as

- Dates – timestamp of crime incident
- Category - crime category
- Descript – detailed description of the crime incident,
- DayOfWeek - day of week
- PdDistrict – name of the police department district,
- Resolution – how the crime incident was resolved,
- Address - address of the crime spot
- X – longitude
- Y – latitude.

4.1.2 Data Preprocessing

As discussed in section 2, this project is implemented in 3 phases. Each phase involved different data preprocessing. They are discussed below:

4.1.2.1 1st Phase: In Phase1, we removed Descript, Resolution and Address fields as they are known after the crime had occurred and don't really help in categorizing the crime. We removed address field as the X (longitude) and Y(latitude) values can exactly locate the crime spot. We also added new features called Year, Month, Day, Hour. They are obtained by using the given time stamp value. After all the preprocessing, we ended up with 9 features

We performed one hot encoding on categorical values DayOfWeek and PdDistrict and normalized the numeric values like X, Y, Year, Month, Day, Hour. The target variable Category is encoded using label encoding.

After the above preprocessing steps, we split the data into 75% training data and 25% testing data.

4.1.2.2 2nd Phase: In second phase, we utilized the 9 features from phase 1. Then, 39 crime categories are categorized into 2 types. White Collar crimes that includes crimes like forgery/counterfeiting, fraud, bribery. Blue Collar crimes that includes crimes like vandalism, robbery, assault. In this way we transformed our problem to a binary classification problem.

We performed one hot encoding and normalizing similar to phase 1 and split the data to 75% training and 25% testing data.

4.1.2.3 3rd Phase: In third phase, we boosted the existing data with additional features from US census data. The new features are

- Zipcode – zip code of the area according to latitude and longitude values
- Population – Population in a zip code
- UnEmployment_Percent – percentage of unemployed people in a zip code
- Mean_Family_Income – mean family income in a zip code

- Percent_of_Minorities – percentage of minorities in a zip code
- Median_Age – median age in a zip code,
- Uneducated_Percent – percentage of uneducated people in a zip code area
- Poverty_Percent – percentage of people under poverty.

After adding the new features, we performed one hot encoding and normalizing like phase 1 & 2. We also normalized these new features. And split our data to 75% training data and 25% testing data.

4.2. Data visualization:

In order to better understand the data, we created data visualizations described below.

4.2.1 Frequency of crimes in PdDistricts

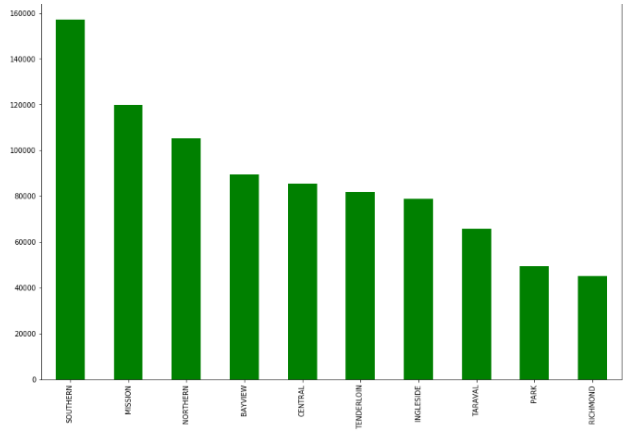


Fig. 2: Frequency of crimes in PdDistricts

Fig 2 represents number of crimes registered in each Police district. While Southern district had the highest crime frequency, Richmond had the least.

4.2.2 Frequency of crimes on each day of week

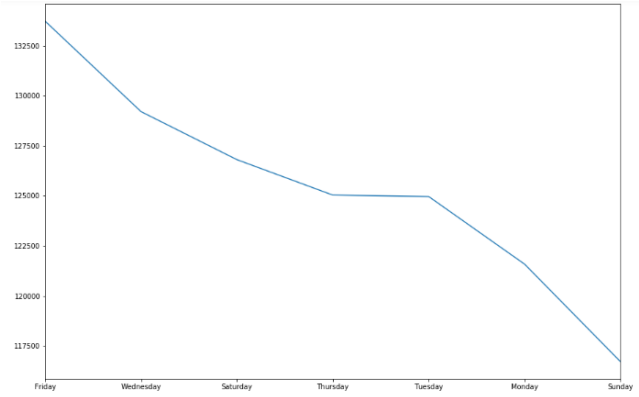


Fig. 3: Frequency of crimes in each day of week

Fig 3 represents number of crimes happening on each day (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday). Friday, Wednesday and Saturday had the highest crime rate.

4.2.3 Frequency of crimes on each category

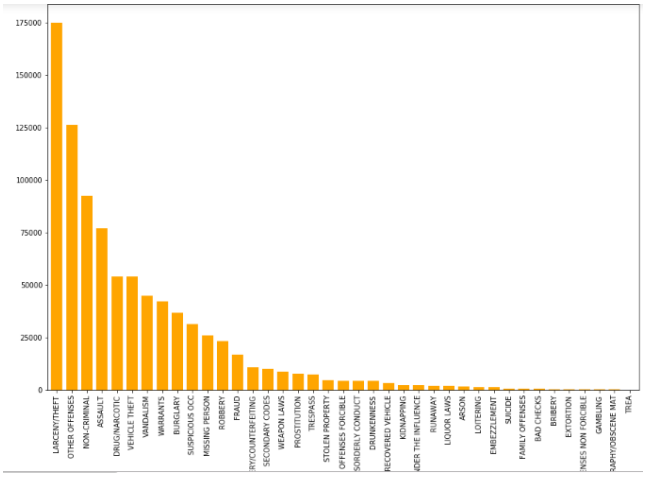
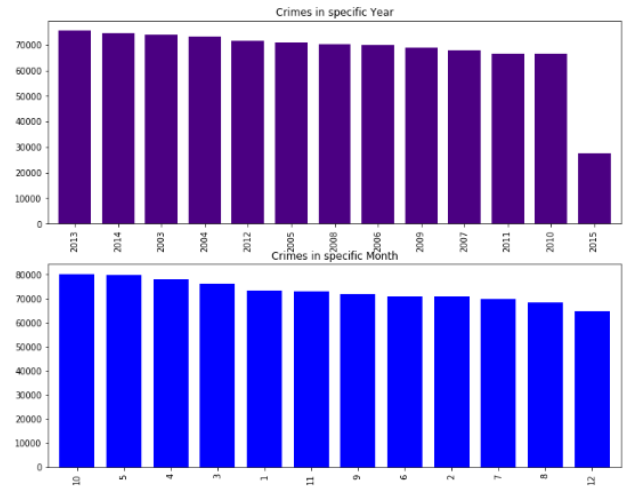


Fig 4: Frequency of crimes on each category

Fig 4 represents frequency of crimes in each category. We observed that last 17 categories of crime have only 5-8% of total crime.

4.2.4 Frequency of crimes based on time stamp



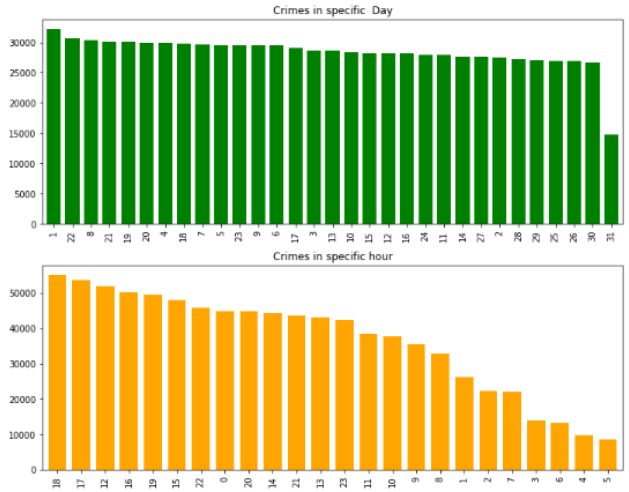


Fig 5: Frequency of crimes based on time stamp

Fig 5 represents frequncy of crime in each year, month, day, hour. We observed that:

- Of all the years, 2013 has the highest number of crimes and 2015 has least.
- Of all the months, October month has the highest number of crimes and december has least.
- Of all the days of a month, First day of the month had most crimes and 31st day has least
- Highest number of crimes were recorded at 6pm and least crimes were recorded at 5am.

4.3 Metrics and Methodology

In phase 1, we used log loss as the main metric and the base line model[2] was using it to faciliatate fair comparision between the models. In phase2, we used accuraccy as a key metric for the model.

We implemented were XGB classifier, random forest Decision tree, Naïve Bayes, KNN as implemented in base project. As an extension, we implemented SVM and Dense neural networks. Results are tabulated in the below section 4.5 for all the models.

4.4 Methodology

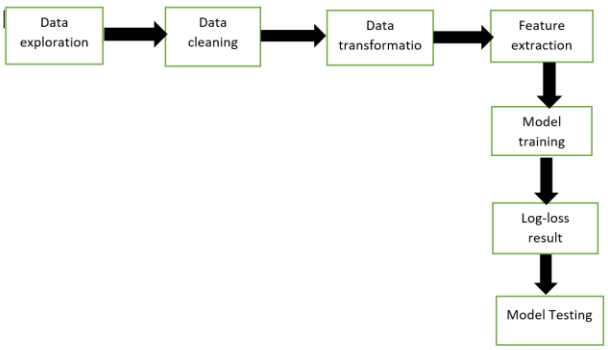
4.4.1 1st Phase

After completing data preprocessing we build our model using algorithms like KNN, Naïve Bayes, Logistic Regression, SVM, DNN, XBG classifier and random forest classifier. Of all the algorithms used random forest gave us the best result. In first phase we got better log loss values than the base line model. These results can be viewed in table1.

4.4.2 2st Phase

In 2nd phase after converting the crime categories into 2 types as white collar and blue collar crimes we used almost the same algorithms as in 1st phase, but we got better log loss, accuracy and f1 scores in 2nd phase and the results can be seen in table 2.

4.4.2 3rd Phase



In 3rd phase it was hard to get the census data. Initially we tried in several ways to get the census data and in that process we came across a medium article [6]. This article explained how to get census data. Using those steps we got the census data. We tried to combine census data and our dataset using latitude and longitude values but those values are not given in the census data.

We then tried to convert latitude and longitude values to zip code values. We performed reverse geocoding using Google reverse geocoding API and we also used Texas A & M GeoServices. After getting the zipcode values we retried 8 new features from census data. We then combined our dataset with US census data.

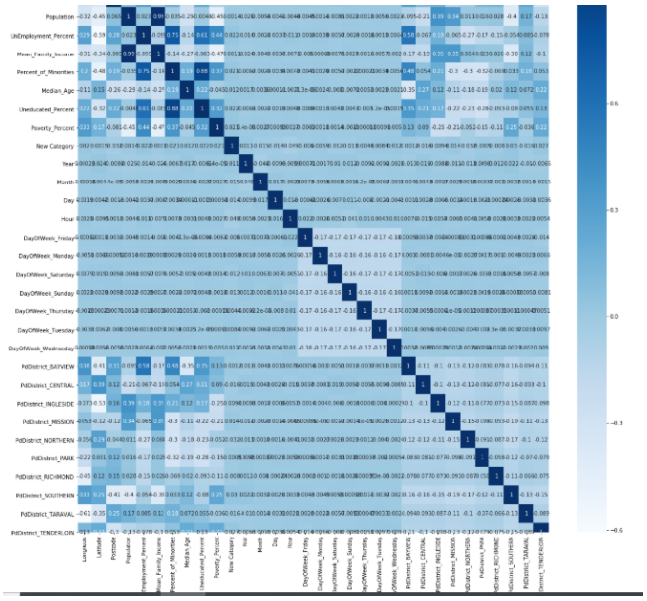


Fig 8: Heat Map

After preparing the data, we build a new model with the same algorithms used in the 2nd phase and the results are almost similar to the results in 2nd phase.

4.4 Results

Table1 for Phase-1 where crimes are classified into 39 categories.

model	Log loss	Base model
KNN(k=100)	3.89	3.96
KNN(k=3000)	2.61	2.62
KNN(k=1000)	2.67	2.70
Random Forest	2.369	2.41
Naïve Bayes	2.634	2.64
Logistic Regression	2.615	-
SVM	-	-
Neural network	2.642	-
XBG classifier	2.365	2.57
randomforest(n_estimators=100)	2.49	2.44
randomforest(n_estimators=200)	2.44	2.42

Table 1: Log loss for various models used in ths project.

Phase 2 results after using while collar and blue collar crime classification is shown in Table2.

Model	Log loss	Accuracy	F1-Score
Random forest	0.610	68	59
Naïve Bayes	0.631	67	54
KNN	13.14	65	65
Decision tree	13.33	61	61
Logistic regression	0.632	67	54
SVM	23.18	67	16
DNN	0.634	67	65

Table 2: Log loss, Accracy and F1score for various models used in the project

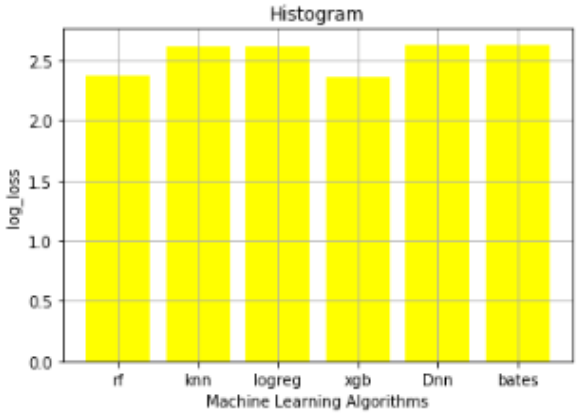


Fig 6: Histogram for results in phase1

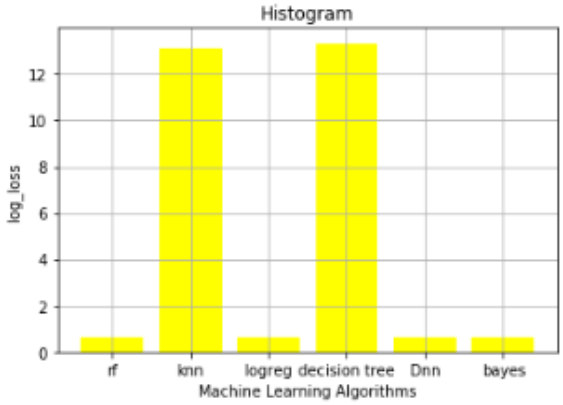


Fig 7: Histogram for results in phase2

5. RELATED WORK

The base line model we considered for this project is a submission of a student [2] to an online competition conducted by Kaggle Inc[3]. The base line model used K-Nearest Neighbors Classifier, XGB classifier, Decision Tree Classifier, Bayesian Classifier and Random Forest classifier. Stanford students[4][5] worked on a similar dataset and they classified the crime categories into white collar and blue collar crimes to improve their model. They also used US census data to augment their dataset.

When compared to the baseline project [2], We tried to improve the model using algorithms like Support Vector Machine and Deep Neural Networks in our 1st phase of work. In [2] the problem was to classify the crime categories where feature selection was not performed, and the address attributes were taken as input. Whereas in our model we drew a heatmap and selected the attributes that are corelated with the output label and as an extension we performed the collapsing of classes and got better results than the base project. We used [4][5] to improve our model by consolidating the crimes categories into white and blue collar crimes. In [4] the same problem was addressed with a different approach as the data is not good enough to predict the classes. They used collapsing of classes and US census data was added to the original crime data to improve the accuracy. We used this mehtodology to improve our prediction as well. In addition to thei r features from US census data, we were using more attributes like Unemployment at that place into criteria to predict the output label.

6. CONCLUSION

In the initial classification, we got the least log loss value of 2.36 for both random forests and XBG classifier which is better than any model in the Base paper[2]. We observed that SVM was bad for multi-classification problem and it took forever to execute it. The results from Neural network was good but not as expected. KNN gave good performance as the value of k is incresed. Naïve Bayes and Logistic regression are good and there was not enough predictability in our initial dataset to obtain very high accuracy on

it. When we classified data into white-collar and blue-collar crimes the log loss got down to 0.61 for Random forests and the accuracy also increased to 70 %. But for this classification SVM and KNN performance was not as expected. Naïve Bayes, Logistic regression and dense neural network were also good. When tried to add address column (street no and block number) as a trail to increase the accuracy there was not much difference. For both problems Random forest took lesser time compared to al other algorithms and gave best results.

As the top 4 categories of the data are composed of almost 53% of the data and the rest of the 47% is comprised of the remaining categories. As we observed this, we removed 15-17 categories and tried increasing the accuracy, but the differences were negligible. In phase 3, after adding additional data (US Census data) we hoped that it would improve our metrics but we almost got the same results.

7.WORK DIVISION

Phase1

Veena: KNN, SVM, Naive Bayes, Logistic regression

Swetha: XBG classifier, Decision Tree, Neural networks, Random Forest

Phase2

Veena: Collapsing 39 classes to 2 classes, KNN, SVM , Naïve Bayes

Swetha: Decision Tree, Neural networks, Random Forest, Logistic Regression

Phase3

Veena: Getting census data, Data preprocessing

Swetha: KNN, SVM, Naïve Bayes, Decision Tree, Neural networks, Random Forest, Logistic Regression

8.LEARNING EXPERIENCE

We learnt the following from this project:

- Handling huge datasets in this project.
- Working on multi classification problems.
- Learnt using SVM, KNN, decision tree, random forests, logistic regression, neural networks and comparing their performance and analyzing which algorithm is suitable in which situation.
- Parameter tuning and Feature Engineering.
- Drawing meaning full conclusing from data visualizations.
- Obtaining data from US Census using their API calls.
- Google Maps API for reverse geocoding.
- Texas A&M Geo Coding Service

REFERENCES

[1] Federal government's open data website: https://datasf.org/

[2] Yehya Abouelnaga. San Francisco Crime Classification. School of Sciences and Egnineering, Department of Computer Science and Engineering The American University in Cairo, New Cairo 11835, Egypt

[3]https://www.kaggle.com/c/sf-crime/data

[4] Addarsh Chandrasekar, Abhilash Sunder Raj and Poorna Kumar. Crime Prediction and Classification in San Francisco City. Stanford University.

[5] John Cherian and Mitchell Dawson. RoboCop: Crime Classification and Prediction in San Francisco. December 11, 2015.Stanford University.

[6]https://upload.wikimedia.org/wikipedia/ru/d/de/Neuro.PNG