**IST 707: Data Analytics**

**San Francisco Crime Analysis**



**Members:**

Ashokkumar Sharma

Eashani Deorukhkar

Hamza Mighri

Shivani Kulkarni

Professor: Humayun Khan

[**Introduction**](#_23qw4vqzd4k2) **3**

[Objective](#_n7d6f23za1h) 3

[Data:](#_7z9roetkspy5) 3

[Business Questions:](#_ltdvgaip51d9) 4

[Methodology:](#_8hx5afycvfz2) 4

[**Data Exploration**](#_qot1n1kwx7yq) **5**

[**Data Pre-processing**](#_efy1s7pmd1h8) **8**

[Performance Measure: Log Loss](#_gsqjzy89qgep) 9

[**Models**](#_dka40u5ehq50) **10**

[Association Rule Mining](#_nkik4z6skihm) 10

[Data Cleaning and Preprocessing:](#_q2t3lmtru0fx) 10

[Exploratory Data Analysis:](#_jnwm75jxarjv) 10

[Association Rule Mining Computation](#_f2s3pjdcc8t8) 11

[Association Rules Selection:](#_z3nu1bypabqm) 12

[Visualizing Association Rules:](#_llfwheamx1f6) 12

[K - nearest neighbours](#_q6t4ixeq3uj0) 14

[Z-Scoring](#_z6ntqqax0k9f) 14

[Prediction Model](#_3gtow19rvhnz) 15

[Random Forest](#_s5cbcgnst8g8) 15

[Naive Bayes](#_z1oeu76z1gra) 16

[**Recommendations**](#_fyfdn9ytlt9s) **17**

[**Appendix**](#_pbhhcf1ucr9) **18**

# 

# **Introduction**

## **Objective**

San Francisco, a city mainly known for its technology scene, is no stranger to criminal activity. The rising wealth inequality and increasing population is increasing the rate of crimes in the city. This project aims to analyze the patterns in crimes occurring in the districts of San Francisco and try to predict the category of the crime based on the location and time of day. We also aim to generate insights

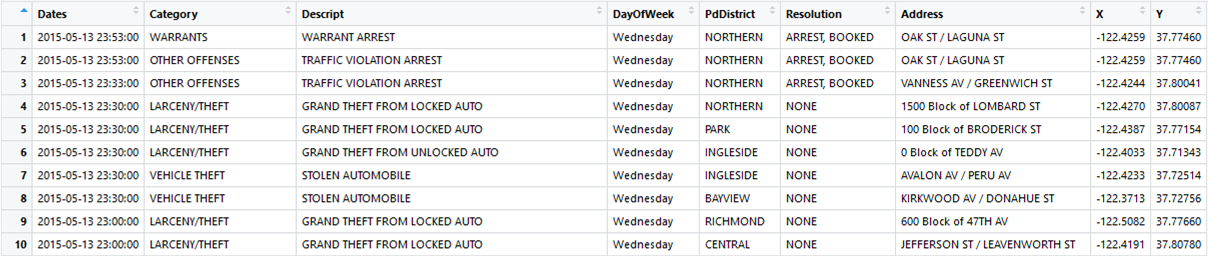
## **Data:**

Source: <https://www.kaggle.com/c/sf-crime>

The dataset which has been sourced from kaggle contains incidents derived from SFPD Crime Incident Reporting system. The data ranges from 1/1/2003 to 5/13/2015. The training set and test set rotate every week, meaning week 1,3,5,7... belong to test set, week 2,4,6,8 belong to training set. It contains fields like:

* Dates - timestamp of the crime incident
* Category - category of the crime incident. This is the target variable you are going to predict.
* Descript - detailed description of the crime incident (only in train.csv)
* DayOfWeek - the day of the week
* PdDistrict - name of the Police Department District
* Resolution - how the crime incident was resolved (only in train.csv)
* Address - the approximate street address of the crime incident
* X - Longitude
* Y - Latitude

Out of these columns, the columns Category and Resolution are not present in the test data.This data is a part of a competition on Kaggle and the results from this project will be submitted to kaggle for evaluation.



## **Business Questions:**

The insights generated by our analysis will be used to answer the following business questions:

1. Which districts have the highest crime rate so that action could be taken in those areas?
2. On what days is the crime rate high?
3. Which crime has the most number of unresolved cases?

## **Methodology:**

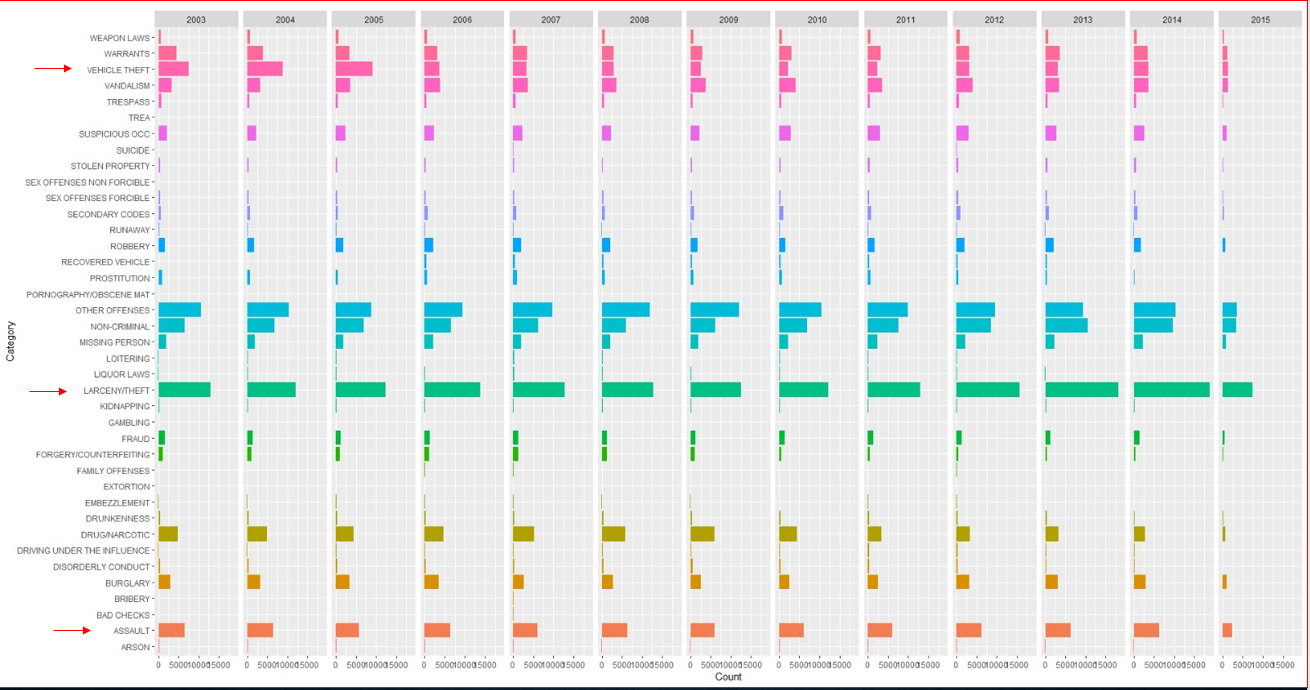
A part of the analysis will be devoted to answering these business questions and a part of the analysis will be devoted to prediction of the categories of crime based on day, time and location. The data will be explored and visualized using line, bar and scatter charts to observe patterns.As the data is geographical in nature, maps of San Francisco will also be used for visualizations. This will be followed by running various algorithms on the data to generate insights and create prediction models. The following algorithms will be used:

* Association rule mining: This is a [rule-based machine learning](https://en.wikipedia.org/wiki/Rule-based_machine_learning) method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness. This will be used to answer the business questions.
* K-Nearest Neighbours: this is a non-parametric method used for classification. The input consists of the k closest training examples in the feature space. The output is the class that a particular data point belongs to. This method will be used to classify and predict the categories of the crimes.
* Random Forests: Random forests or random decision forests are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for classification, [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks that operates by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the classes. This method will be used to classify and predict the categories of the crimes.
* Naive Bayes Classifier: This is a probabilistic classifier based on applying Bayes' theorem with strong independence assumptions between the features. This method will be used to classify and predict the categories of the crimes.

# 

# **Data Exploration**

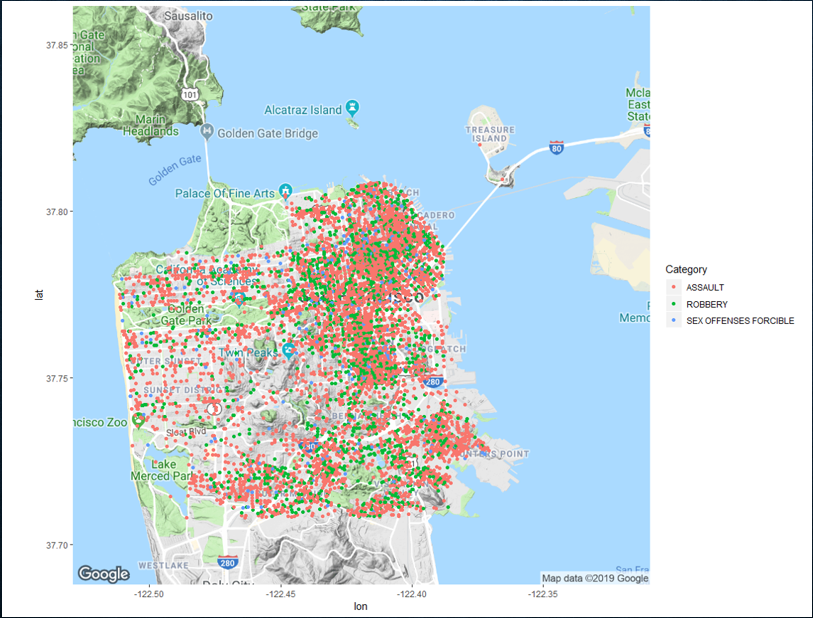
A preliminary exploration of the data reveals the most frequently occurring crime categories over the years.



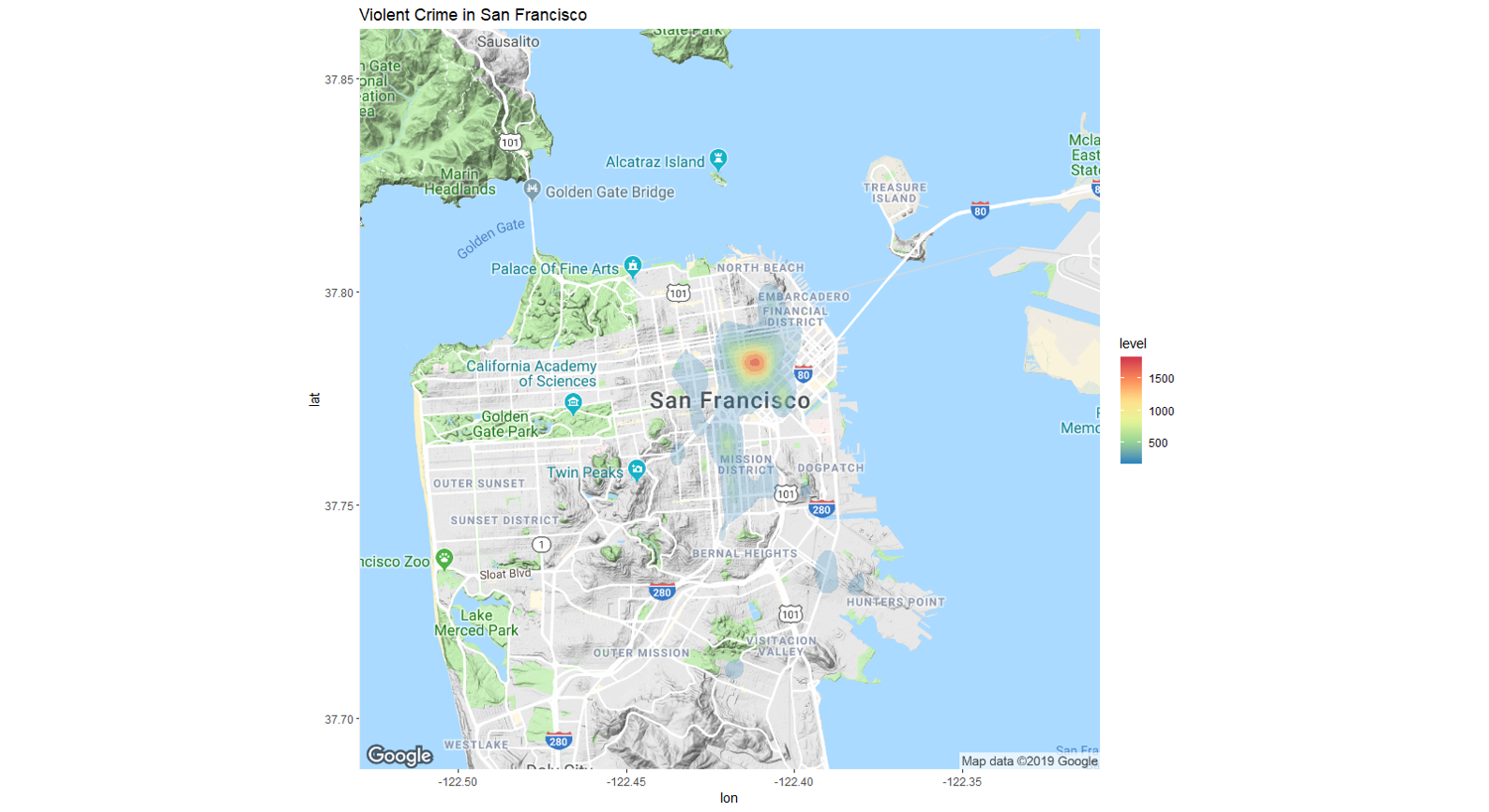
Excluding the categories “OTHER CRIMES” AND “NON CRIMINAL” offences as they are ambiguous, the categories “VEHICLE THEFT”, “ASSAULT”,”LARCENY/THEFT” seem to have the maximum amount of crimes.

It is seen that the crimes drop off in 2015 but the reason for this is that the data ends in May 2015.

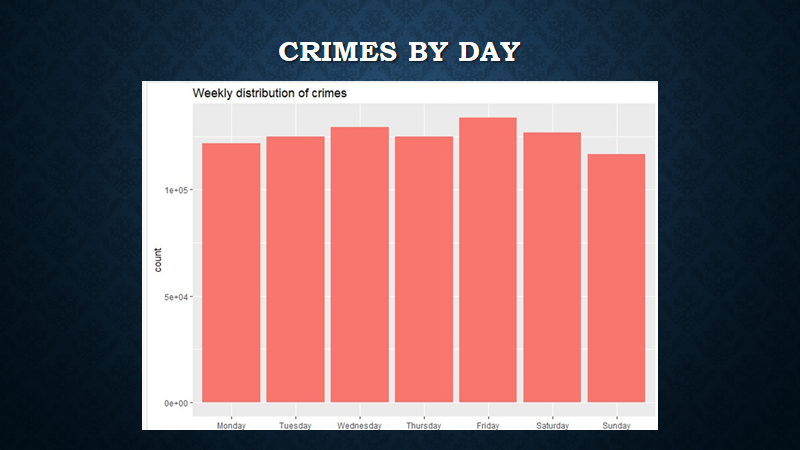
From this we can categorize Vehicle theft, Assault and Larceny/Theft as violent crimes and have an idea of the districts which are known for violent crimes.



A heat map also is generated to see the distribution of crimes



The following graph shows the distribution of crimes over the week. It may seem that the crimes are constant throughout the week but there is a peak on Fridays. The reason for this may be because people usually go out on friday nights and hence are more vulnerable to crimes.



# 

# **Data Pre-processing**

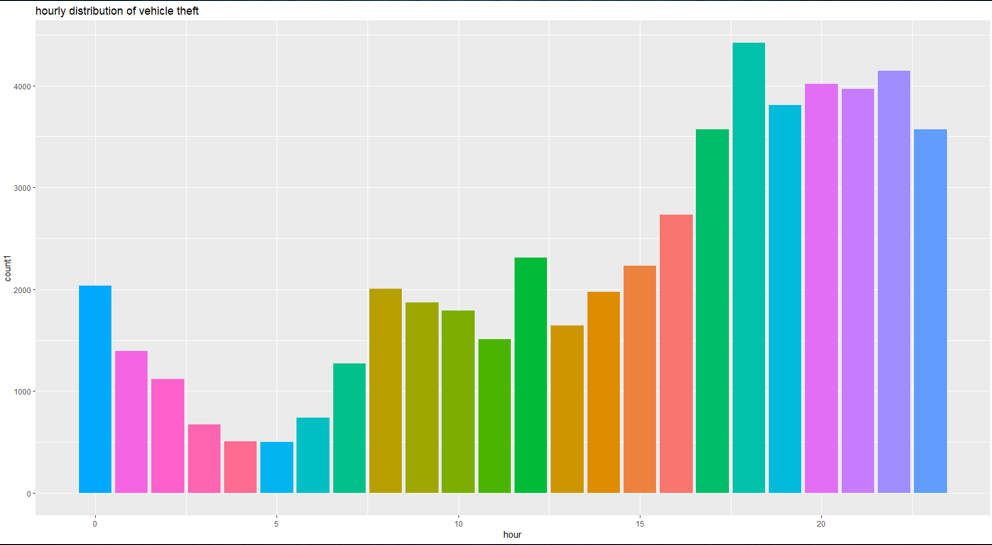
* The raw dataset contained a column named ‘Dates’- timestamp of the crime that had occured. For the purpose of modelling, it was split into fields like date and hour separately.
* The raw dataset also consisted of column ‘DayofWeek’, based on which another column was generated -’iswkend’. This column determines whether the day of the crime was weekend or not.
* The fields thus generated were:

-Date

-Hour (Range 0 to 23)

-iswekend (Yes or No)

By this transformation, we can see the distribution of crimes throughout the day. As a small part of it, VEHICLE THEFT was considered to be visualized



This graph shows that crimes have a higher rate of occurrence during the latter part of the day, that is from evening 4 till night.

## **Performance Measure: Log Loss**

**Accuracy:**

Predicted Value = Actual Value

**Log Loss:**

* Log loss is the uncertainty on how much Predicted Value varies from Actual Value.
* It uses the function -1 \* Log(Likelihood Function)
* In contrast to how accuracy works, that is better the accuracy better is the model, Lower the log loss value, better is the model.
* For this dataset, kaggle used log loss as a measure of performance for the models. While the benchmarking was at 2.4

# 

# 

# 

# 

# 

# 

# **Models**

## **Association Rule Mining**

The topic of the project is to identify frequent patterns in a crime dataset in the city of San Francisco. As we will explore later in the process, the dataset covers various attributes of a series of crimes that took place between 01/01/2003 and 05/13/2015 including: Date, Crime Category, Crime Description,Day of Week, District,Crime Resolution, Address,Latitude and Longitude. The Association Rule Mining technique has been gaining more and more ground in the field of business analytics, more particularly in retail businesses like Amazon and eBay. It allowed to determine items that are often bought together and based on them makes recommendations for customers. In the context of crimes rate data, the technique could help us detect hidden pattern that we would not be able to detect using other predictive techniques. At glance, it would allow us to collect information on what characteristics the areas with high crime rates have. Once revealed, we could make valuable policy recommendations to reduce crime rates. We start by depicting the strongest associations in the entire dataset before defining the target variable to be the type of crime.

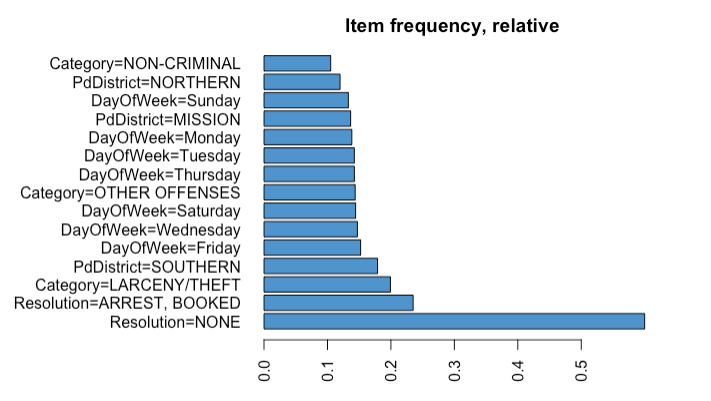
### **Data Cleaning and Preprocessing:**

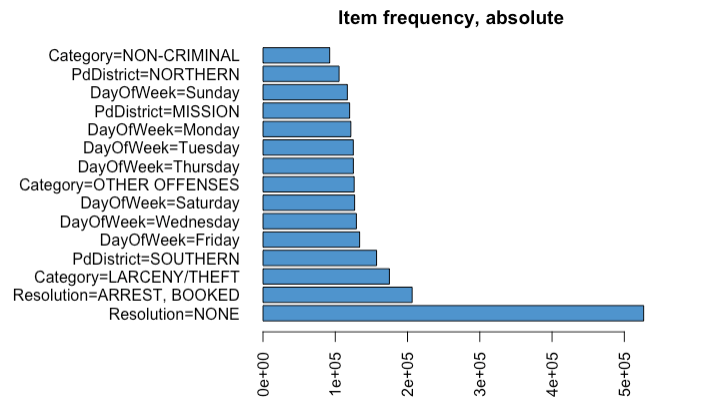
The key pre-processing element in preparing the data set for association rules mining is to ensure that the data is discretized. As suggested by the R code in the appendix below, our data is not continuous and further columns selection ensure that we neglect time and date data. We then proceed to convert our cleaned data into a sparse matrix (transaction data type), a type of data required to compute the association rule mining. The following columns are withdrawn from the data due to the either redundancy or having a low importance in the analysis:

* Date and time of crime case
* Description of the crime
* Exact address of the crime
* Longitude of the location
* Latitude of the location

### **Exploratory Data Analysis:**

After transforming the data set into a sparse matrix, we compute basic exploratory data analysis for the association rules mining. This consists of the most frequent items (instances) in the entire data set. The following plots provide items frequencies in an increasing order for the top 15 items, in both relative and absolute terms:



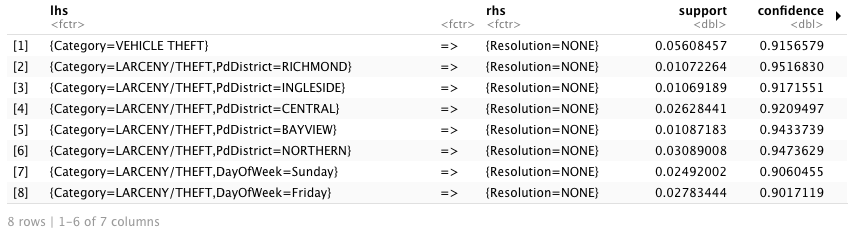


The initial insight gained from the items frequency plotting in both absolute and relative terms, we can observe that Resolution=NONE is the most frequent item in the dataset for both plots. Our interpretation of the EDA is that the overall performance of the security and police department is not satisfactory, since most cases have not been resolved yet. This observation is of critical importance, as it will allow us to narrow down our focus to the main policy issue for the San Francisco city. In other words, by detecting the most frequent issue, we can tune the model to focus on its determinant in the hope of addressing the issue by influencing those parameters. More technically, we can fix our target variable (right hand side of the association equation to be the unresolved crimes, i.e. rhs= Resolution=NONE). Nonetheless, our code below show our preliminary attempts to account for the strongest rules without controlling for the target variables.

### **Association Rule Mining Computation**

After trying various iterations for our model to ensure consistency in the various results, we fixed our association rule strength to be as follow: Support = 0.01 and Confidence = 0.8. The algorithm yields 21 rules in total. We, then, decide to filter our output to have the highest level of Lift (ratio of confidence/expected confidence) to be at least 1.5. As a result, the algorithm detects eight strong association as displayed below:

### **Association Rules Selection:**



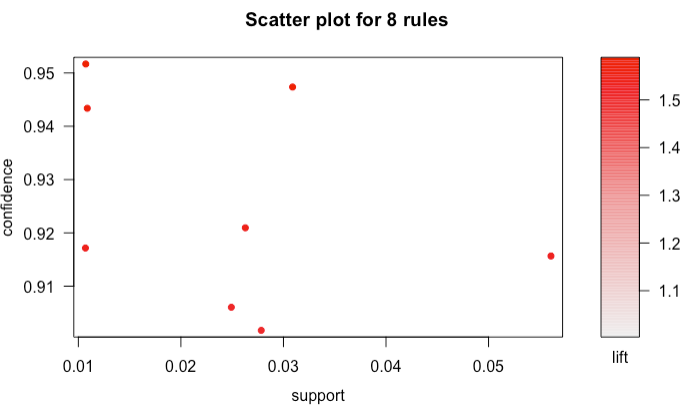
The obtained rules are in strong support of the following associations:

Unsolved crime cases are mostly cases of vehicle theft.

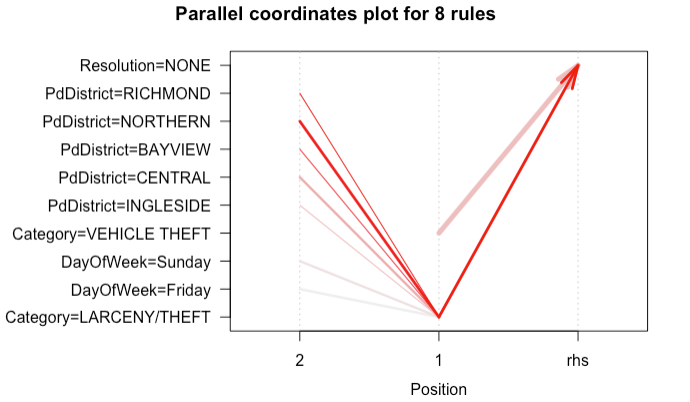
* Unsolved crime cases were mostly related to Larceny/Theft.
* Unsolved crime cases occurred in the following districts: Richmond, Ingleside, Central, Bayview and Northern.
* Most unsolved crime cases happened over the weekend, more particularly on Friday and Sunday.

### **Visualizing Association Rules:**

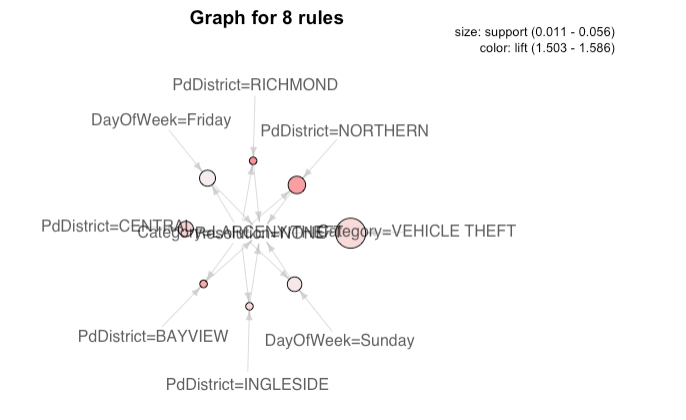
To better communicate results to decision-makers, we provide various visualizations of our association outcomes. Below is the scatterplot of the eight strongest associations. The point are visualized as a point coordinate of support level (x axis) and confidence level (y axis).



The below parallel coordinates plot is another tool of displaying the strongest association rules. The arrow indicate the direct of the association (i.e the target variable is Resolution=None). The shorter the arrow, the stronger the association. For instance, Category= Vehicle Theft, is the strongest rule, as suggested by the table output above.



Finally, the network plot provides another visualization of association rules. At the center of the graph, we have out target variable, Resolution = None, and the branches display the 8 pre-selected association rules. The shorter the branch to the center, the stronger the association (higher lift value) and vice-versa.



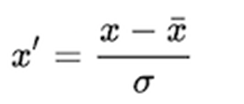
## K - nearest neighbours

### **Z-Scoring**

Hour ranges from 1 – 24 and Longitude & Latitude have a vast range. If the prediction were to be done on raw data, it would provide a biased results. To solve this problem, we use Z- Scoring

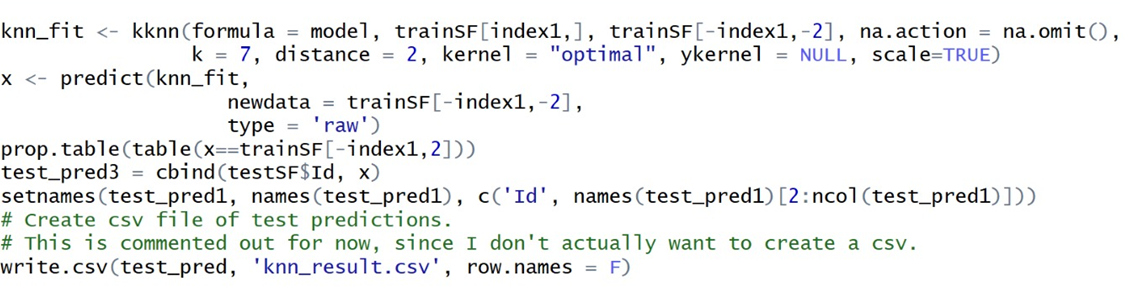
Z-Scoring uses Euclidean distance and Standardising using Mean & Standard Deviation

Formula:



### Prediction Model

The kNN model was tried for all Ks from 1 to 23. However, the best result was found at k = 7 using the elbow curve. This model tries to predict the category of the crimes. By grouping similar crimes together.





We can see that the kaggle score is 16.7 for kNN model whereas the benchmark was at 2.4 and hence the model does not have optimum output. Further increasing or decreasing of the k value did not have any significant change in the accuracy.

## Random Forest

We tried running the Random Forest model on the data with variables day\_of\_week, pd\_district, x, y, hour as the independent variables.

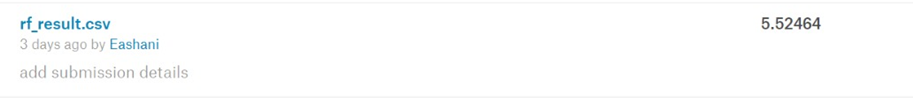
As a starting point, we ran a model with only 10 trees. This model was ran fast but the result was not very good. The log loss (the parameter of evaluation on Kaggle) was 15.92 when the value should have been closer to 2.5.



We decided to tune the model and add 100 trees to the model instead of 10. This increased the complexity of the model and the model took approximately 2 hours to run.

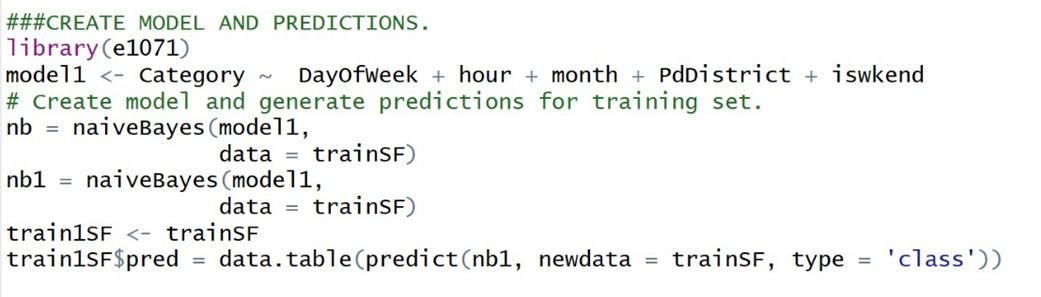
However the result of the model was considerably better with a kaggle score of 5.5 which was an update on the previous score of 15.9

Further increasing of trees did not give any greater accuracy and our machines were not able to support it.



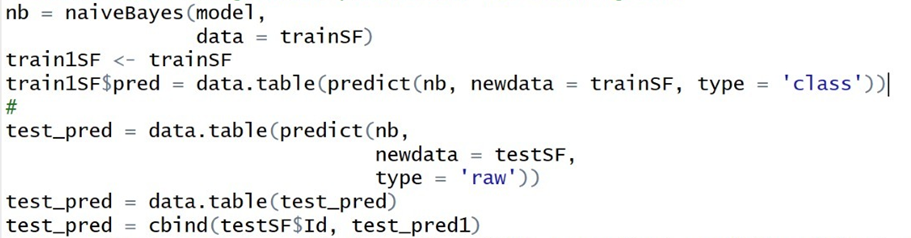
## Naive Bayes

**Model 1:** Naive Bayes Classifier model was run with parameters like DayofWeek, hour, month, PdDistrict, iswkend



**Problem faced**: Naive Bayes predicted all the crimes as Assault and hence could not be uploaded to kaggle

**Model 2**: Naive Bayes Classifer model was run with parameters like DayofWeek, hour, month, PdDistrict, iswkend , longitude and latitude values.





The score is 3.445 which is close to the benchmark of 2.4. The Naive Bayes classifier proved to be the best model for predicting crime based on location, day and time of the day.

# **Recommendations**

We conclude by making the following recommendations to the San Francisco security and police authorities:

* Deploy more security forces to the following districts: Ingleside, Northern, Central, Bayview, and Richmond.
* Increase security forces in the public areas during the weekend especially on Friday and Sunday due to high crime rates during those days.
* Increase public surveillance in public areas by investing in more technological surveillance tools like high quality cameras.
* Implement awareness campaigns among the general population to be more cautious about their properties in the specific above rich districts, since private properties seem to be the main target for criminals.
* More broadly, we recommend dealing with social and economic disparities as they play a main role in the steady crimes rates against relatively rich areas.

# **Challenges**

* The dataset had multiple categories to predict (39) and thus the accuracy was very low (approximately 30%).
* We earlier planned to use Support Vector Machines as a possible model but the complexity of the model was too high and it did not execute even after running it for 12 hours.
* Gathering data on social and economic indicators in the areas where the crimes occurred. That would help us better shape effective policies

# **Appendix**

**#Model 1**

R code for Association Rule Mining:

```{r}

#Loading the dataset

mydata <- read.csv("~/Desktop/Conference Paper/trainsfcrime.csv")

```

```{r}

# Loading necessary packages

library(arules)

library(arulesViz)

library(ggplot2)

#Removing optional variables:

finaldata <- mydata[,-c (1,3,7,8,9)]

#Converting the data into a sparse matrix dataset:

TransactionCrimeData <-as(finaldata,"transactions")

#Exploratory Data Analysis for Association Rules Mining: we display the 15 most frequent items in the dataset both in terms of absolute and relative frequencies.

itemFrequencyPlot(TransactionCrimeData,type="relative",topN=15,

horiz=TRUE,col='steelblue3',xlab='',main='Item frequency, relative')

itemFrequencyPlot(TransactionCrimeData,type="absolute",topN=15,horiz=TRUE,col='steelblue3',xlab='',main='Item frequency, absolute')

# => Observation: Resolution =None is the most frequent item both in abolsute and and relative terms.

```

```{r}

# Association rules mining:

myRules =apriori(TransactionCrimeData, parameter =list(supp =0.001, conf =0.3,maxlen =3))

summary(myRules)

length(myRules)

#Removing possible redundant rules

subsetRules <- which(colSums(is.subset(myRules, myRules)) > 1) # get subset rules in vector

length(subsetRules) #> 437

cleanedrules <- myRules[-subsetRules] # remove subset rules.

length(cleanedrules)

myRules1 =apriori(TransactionCrimeData, parameter=list(supp =0.05, conf =0.7,maxlen =3))

length(myRules1)

subsetRules1 <- which(colSums(is.subset(myRules1, myRules1)) > 1) # get subset rules in vector

length(subsetRules1) #> 437

cleanedrules <- myRules1[-subsetRules] # remove subset rules.

length(cleanedrules)

length(myRules1)

```

# Defining the target variables:

```{r}

finalrules <- apriori(TransactionCrimeData, parameter = list(support= 0.01, confidence = 0.8), appearance = list(rhs=c('Resolution=NONE'),default="lhs"))

summary(finalrules)

# Defining the goodrules for high satisfaction

goodrules1 <- finalrules[quality(finalrules)$lift> 1.5]

inspect(goodrules1)

length(goodrules1)

```

# Plotting meaningful associations

```{r}

plot(goodrules1)

plot(goodrules1, method="paracoord", control=list(reorder=TRUE))

plot(goodrules1, method ="graph")

```

**# R code for Random forest model**

**## RF WITHOUT TUNING**

library(randomForest)

# Define model.

model = category\_predict ~ day\_of\_week + pd\_district + x + y + hour

# Set seed for reproducibility.

set.seed(1)

# Create random forest.

rf = randomForest(model,

data = train,

ntree = 10,

importance = T)

# View feature importance.

varImpPlot(rf)

# Generate predictions for training and test data.

# For training data, I want accuracy.

# For test data, the submission format allows probabilities for each crime type for each crime (instead of

1 category as the final prediction).

# These different needs inform the different 'type' of predictions.

train\_pred = data.table(predict(rf,

newdata = train,

type = 'response'))

test\_pred = data.table(predict(rf,

newdata = test,

type = 'prob'))

#####

## **RF WITH TUNING**

# Define model.

model = Category ~ DayOfWeek + hour +month + PdDistrict

# Set seed for reproducibility.

set.seed(1)

library(randomForest)

library(MLmetrics)

# Create random forest.

index1 <- sample(1:nrow(trainSF), 530557)

rf = randomForest(model, data = trainSF[index1,], ntree = 100, importance = T)

# View feature importance.

varImpPlot(rf)

# Compute model performance on training data.

train\_pred =predict(rf,trainSF[-index1,-2])

test\_pred =predict(rf,testSF,type="prob")

print(test\_pred)

x <- table(test\_pred,trainSF[,2])

table(trainSF[-index1,2]==train\_pred)

# Add training set predictions to 'train'.

trainSF$pred = trainSF\_pred$V1

mtry <- 3

control <- trainControl(method='cv',

number=3)

tunegrid <- expand.grid(.mtry=mtry)

rf\_default <- train(model,

data=trainSF,

method='rf',

metric='accuracy',

tuneGrid=tunegrid,

trControl=control)

print(rf\_default)

library(data.table)

library(rfUtilities)

test\_pred = data.table(test\_pred)

test\_pred = cbind(testSF$Id, test\_pred)

setnames(test\_pred, names(test\_pred), c('Id', names(test\_pred)[2:ncol(test\_pred)]))

# Create csv file of test predictions.

# This is commented out for now, since I don't actually want to create a csv.

write.csv(test\_pred, 'rf\_result.csv', row.names = F)

**#Model 4**

**R code for naive bayes model**

###CREATE MODEL AND PREDICTIONS.

library(e1071)

# Create model and generate predictions for training set.

model = Category ~ DayOfWeek + hour +month + PdDistrict

nb = naiveBayes(model,

data = trainSF)

# Generate predictions for training and test data.

# For the training data, I only want to compute accuracy.

# For the test data, I need to put predictions in a specific format for submission, as specified by Kaggle.com.

train1SF <- trainSF

train1SF$pred = data.table(predict(nb, newdata = trainSF, type = 'class'))

#

test\_pred = data.table(predict(nb,

newdata = testSF,

type = 'raw'))

test\_pred = data.table(test\_pred)

test\_pred = cbind(testSF$Id, test\_pred)

setnames(test\_pred, names(test\_pred), c('Id', names(test\_pred)[2:ncol(test\_pred)]))

# Create csv file of test predictions.

# This is commented out for now, since I don't actually want to create a csv.

write.csv(test\_pred, 'nb\_result.csv', row.names = F)

#####

# CHECK TRAINING SET ACCURACY.

# View training accuracy.

print('Training Accuracy:')

table(train$Category == train$pred)

prop.table(table(train$Category == train$pred))