Los Angeles Crime Analysis and Prediction

DATA SCIENCE INTENSIVE CAPSTONE PROJECT

ROJA RANI JYOTHI

11/01/2020

Table of Contents

Introduction

Los Angeles, California is one of the largest metropolitan areas in the country, a port city, and incredibly ethnically diverse. It is within close proximity to the Mexican drug trade, home to the entertainment industry, and has residents across the socio-economic spectrum. Due to these and a combination of other
factors, Los Angeles has a colorful history of crime, including organized criminal activity, gang wars, riots and more. It is also important to note that the city has experienced an overall decline in crimes committed in the last several decades, according to the State of California Department of Justice and the Office of the Attorney General. In 2015, it was revealed that the Los Angeles Police Department had been under-reporting crime for eight years, making the crime rate in the city appear much lower than it really is.

1. Problem Identification

Crime is a social phenomenon as old as societies themselves, and although there will never be a free from crime society - just because it would need everyone in that society to think and act in the same way - societies always look for a way to minimize it and prevent it. Until recently crime prevention was studied based on strict behavioral and social methods, but the recent developments in Data Analysis have allowed a more quantitative approach in the subject. We will explore a dataset of nearly 9 years of crime reports from across all of Los Angeles neighborhoods, and we will create a model that predicts the category of crime that occurred, given the time and location.

Approach

An analysis of crime data within Los Angeles will reveal hidden patterns, trends or relationships between some factors and major crimes. The problem may be articulated as – finding major crime trends in Los Angeles neighborhoods, identifying potential factors related to major crimes, and using these factors to build a predictive model.

To examine the specific problem, we will apply a full Data Science life cycle composed of the following steps:

1. Data Wrangling to audit the quality of the data and perform all the necessary actions to clean the dataset.
2. Data Exploration for understanding the variables and creating intuition on the data.
3. Feature Engineering to create additional variables from the existing.

4. Data Normalization and Data Transformation for preparing the dataset for the learning algorithms (if needed).
    a. Training / Testing data creation to evaluate the performance of our models.
5. Model selection and evaluation. This will be the final goal; creating a model that predicts the type and probability of each crime based on the location and the time.
    a. fine-tune their hyperparameters.
6. Data storytelling - Review the results,Presenting and sharing the findings,Finalize code and Final documentation.

2. Data Wrangling

2.1 Data Collection

The source of the data set we used here is collected from
https://data.lacity.org/A-Safe-City/Crime-Data-from-2010-to-2019/63jg-8b9z

This dataset contains data from 2010 to 2011 and has 2.11M rows and 28 columns in csv format.

The following pdf which contains the description for MO Codes is converted to csv and is used in this project:
https://data.lacity.org/api/views/63jg-8b9z/files/3db69cd3-446c-4dcd-82eb-3436dc08d3be?download=true&filename=MO_CODES_Numerical_20180627.pdf

2.2 Data Definition

In this step we checked the columns, their data types to make sure the data type is loaded properly in the data frame. And converted some of the data types to the appropriate once.

Description of columns:

Column Name	Description	Type
DR Number	Division of Records Number: Official file number made up of a 2 digit year, area ID, and 5 digits	Plain Text
Date Reported	MM/DD/YYYY	Date & Time
Date Occurred	MM/DD/YYYY	Date & Time
Time Occurred	In 24 hour military time	Plain Text
Area ID	The LAPD has 21 Community Police Stations referred to as Geographic Areas within the department. These Geographic Areas are sequentially numbered from 1-21	Plain Text

Area Name	The 21 Geographic Areas or Patrol Divisions are also given a name designation that references a landmark or the surrounding community that it is responsible for	Plain Text
Reporting District	A four-digit code that represents a sub-area within a Geographic Area. All crime records reference the "RD" that it occurred in for statistical comparisons	Plain Text
Crime Code	Indicates the crime committed (Same as Crime Code 1)	Plain Text
Crime Code Description	Defines the Crime Code provided	Plain Text
MO Codes	Modus Operandi: Activities associated with the suspect in commission of the crime	Plain Text
Victim Age	Two character numeric	Plain Text
Victim Sex	F - Female M - Male X - Unknown	Plain Text
Victim Descent	Descent Code: A - Other Asian B - Black C - Chinese D - Cambodian F - Filipino G - Guamanian H - Hispanic/Latin/Mexican I - American Indian/Alaskan Native J - Japanese K - Korean L - Laotian O - Other P - Pacific Islander S - Samoan U - Hawaiian V - Vietnamese W - White X - Unknown Z - Asian Indian	Plain Text
Premise Code	The type of structure, vehicle, or location where the crime took place	Plain Text
Premise Description	Defines the Premise Code provided	Plain Text
Weapon Used Code	The type of weapon used in the crime	Plain Text
Weapon Used Description	Defines the Weapon Used Code provided	Plain Text
Status Code	Status of the case (IC is the default)	Plain Text
Status Description	Defines the Status Code provided	Plain Text
Crime Code 1	Indicates the crime committed. Crime Code 1 is the primary and most serious one. Crime Code 2, 3, and 4 are respectively less serious offenses. Lower crime class numbers are more serious	Plain Text
Crime Code 2	May contain a code for an additional crime, less serious than Crime Code 1	Plain Text
Crime Code 3	May contain a code for an additional crime, less serious than Crime Code 1	Plain Text
Crime Code 4	May contain a code for an additional crime, less serious than Crime Code 1	Plain Text
Address	Street address of crime incident rounded to the nearest hundred block to maintain anonymity	Plain Text
Cross Street	Cross Street of rounded Address	Plain Text

Location	The location where the crime incident occurred. Actual address is omitted for confidentiality. XY coordinates reflect the nearest 100 block	Location

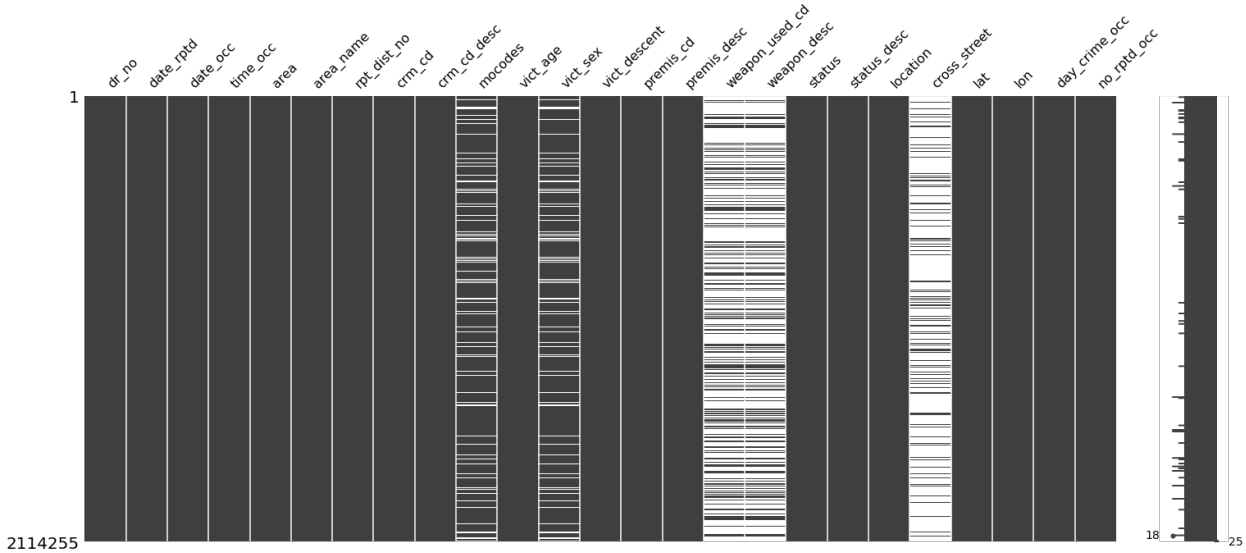
2.3 **Data Cleaning**

- Since the data column titles were mixed of upper and lowercase letters with space in between them, we changed the original column names into a more accessible format to make data operations manageable.

  Before: ['DR_NO', 'Date Rptd', 'DATE OCC', 'TIME OCC', 'AREA ', 'AREA NAME',
  'Rpt Dist No', 'Part 1-2', 'Crm Cd', 'Crm Cd Desc', 'Mocodes',
  'Vict Age', 'Vict Sex', 'Vict Descent', 'Premis Cd', 'Premis Desc',
  'Weapon Used Cd', 'Weapon Desc', 'Status', 'Status Desc', 'Crm Cd 1',
  'Crm Cd 2', 'Crm Cd 3', 'Crm Cd 4', 'LOCATION', 'Cross Street', 'LAT',
  'LON']

  After: ['dr_no','date_rptd','date_occ','time_occ','area','area_name','rpt_dist_no','part_1_2',
  'crm_cd','crm_cd_desc','mocodes','vict_age','vict_sex','vict_descent','premis_cd',
  'premis_desc','weapon_used_cd','weapon_desc','status','status_desc','crm_cd_1',
  'Crm_cd_2','crm_cd_3','crm_cd_4','location','cross_street','lat','lon']

- Converted some of the data types which are not properly loaded to the appropriate data types.
- The main issue with raw data was missing values. I visualized the missingness to know the missing values are by incorrect data entry or any relation with the other variables and filled the NaN's accordingly.



- Dropped the columns 'crm_cd_1','crm_cd_2','crm_cd_3','crm_cd_4','part_1_2' which have more than 90% NaN values.
- Since most of the columns are of object data type, we filled NaN's with most frequent values for some columns and related values for other columns by exploring their respected column values.
- Replaced the strange values in some of the columns with the appropriate value. For example, in vict_desct column I replaced '-' with 'X' (X description is UNKNOWN).
- Replaced the code values with their actual description for some columns. For example

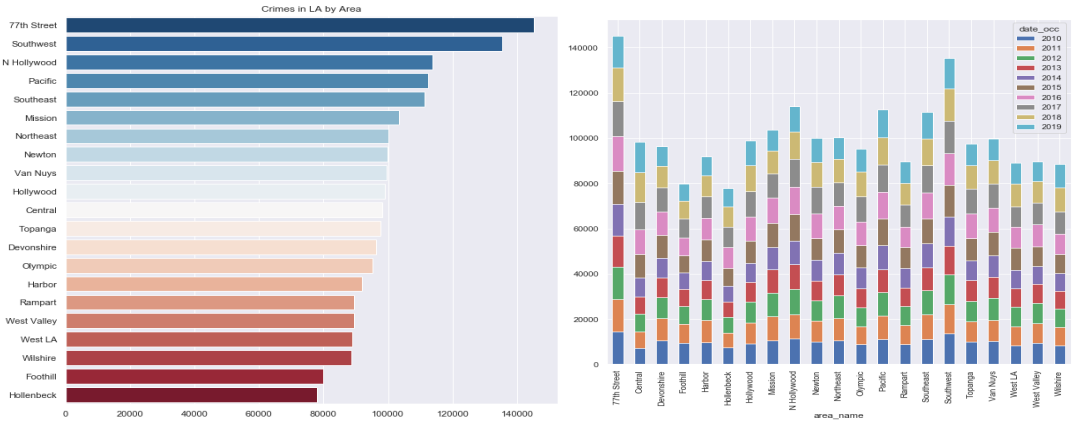
```
#Replacing the 'vict_desct' values with the actual description
replace_values = {"A": "Other Asian",
                  "B": "Black",
                  "C": "Chinese",
                  "D": "Cambodian",
                  "F": "Filipino",
                  "G": "Guamanian",
                  "H": "Hispanic/Latin/Mexican",
                  "I": "American Indian/Alaskan Native",
                  "J": "Japanese",
                  "K": "Korean",
                  "L": "Laotian",
                  "O": "Other",
                  "P": "Pacific Islander",
                  "S": "Samoan",
                  "U": "Hawaiian",
                  "V": "Vietnamese",
                  "W": "White",
                  "X": "Unknown",
                  "Z": "Asian Indian"}
crime_data = crime_data.replace({'vict_descent': replace_values})
```

- Extracted additional information from Time, Date reported and Date Occurred such as hour, minute, day of the week, date, month, year, number of days between Occurred and reported.
- The initial raw data set is of shape (2114451, 28), now after removing the unnecessary columns and additional extracted columns we have the data set of shape (2114451, 29).

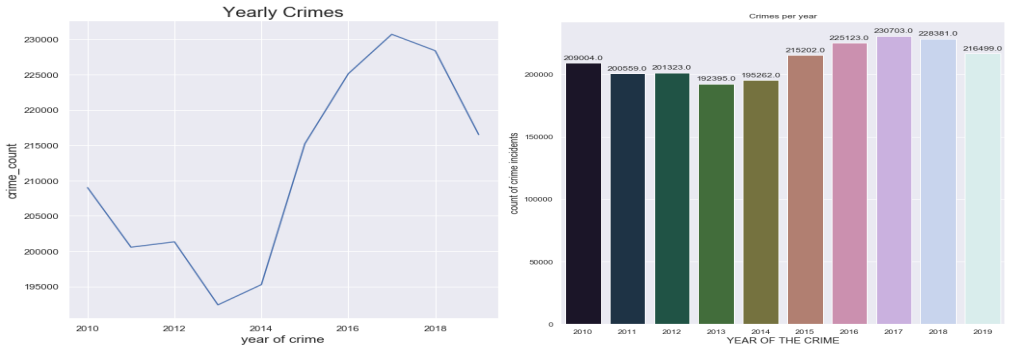
3. Exploratory Data Analysis

I visualized the data from year 210 to 2019 using different ways.

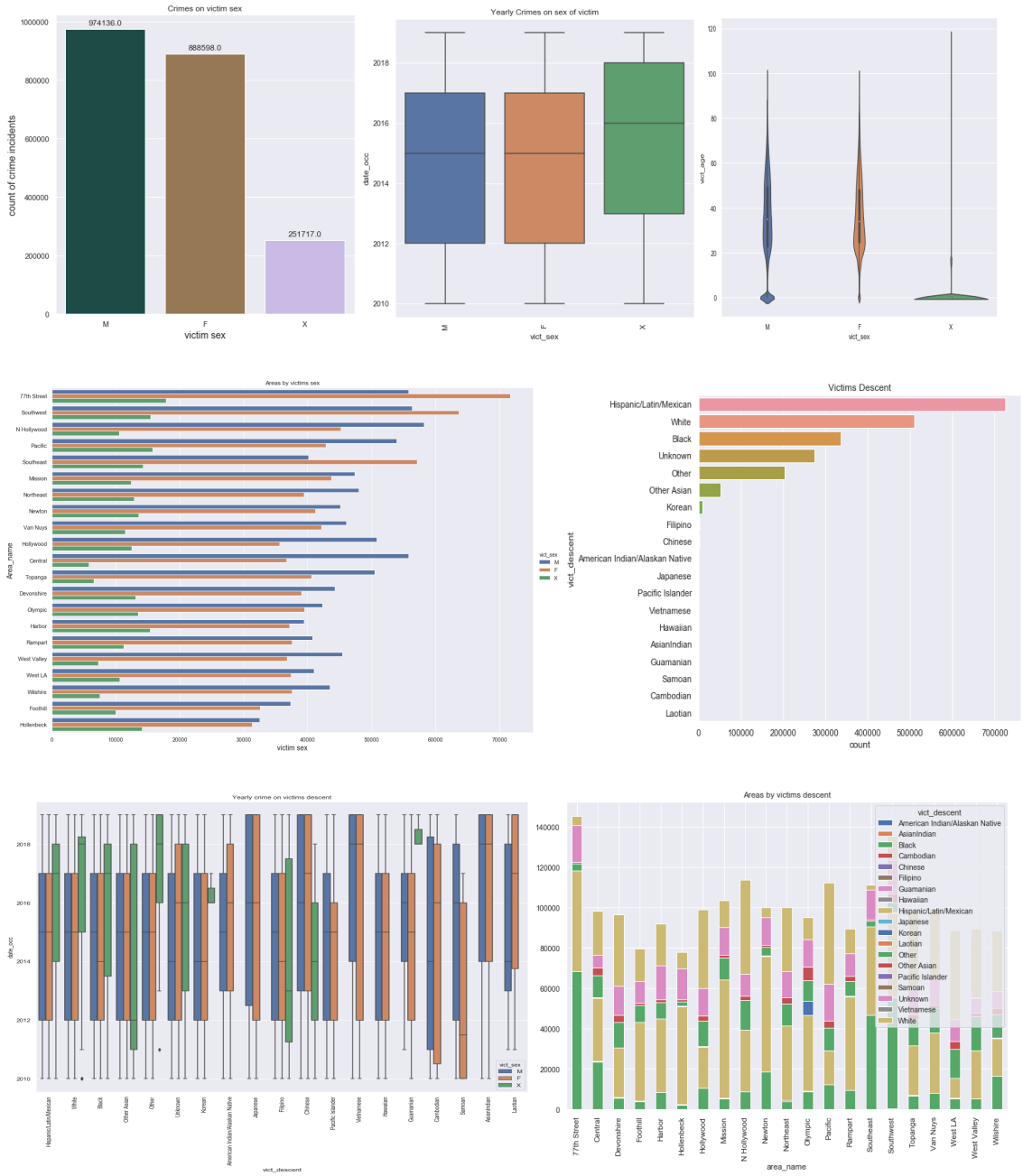
- Initially I explored the LA neighborhoods and their crime counts to know which neighborhood is a hot hub to crimes. It is clear that 77th Street, Pacific and Southwest are not the best neighborhoods in the LA city. And I plotted yearly crimes by area also.



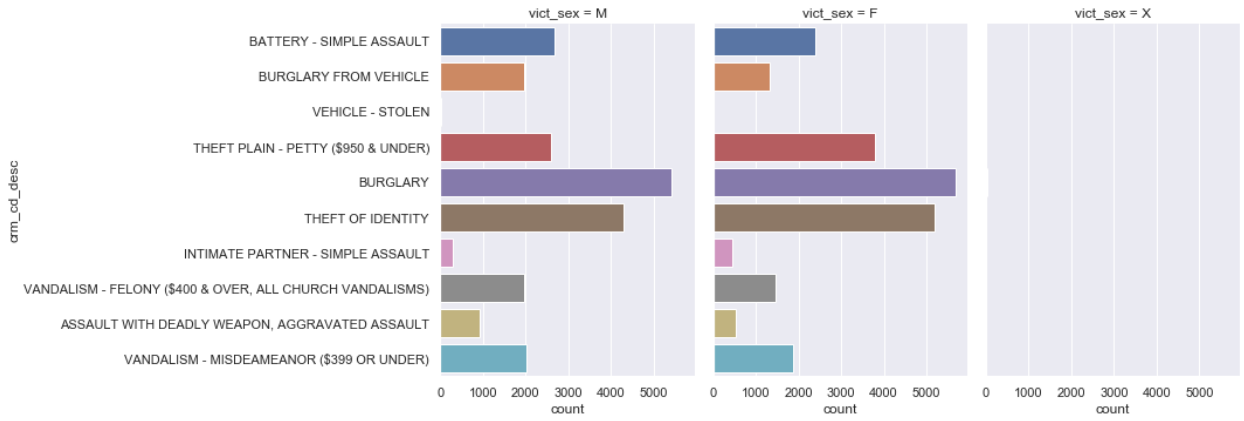
- I analyzed the trend of crime occurrences each year and witnessed that 2017 had the highest crime rate through all the years from 2010-2019.



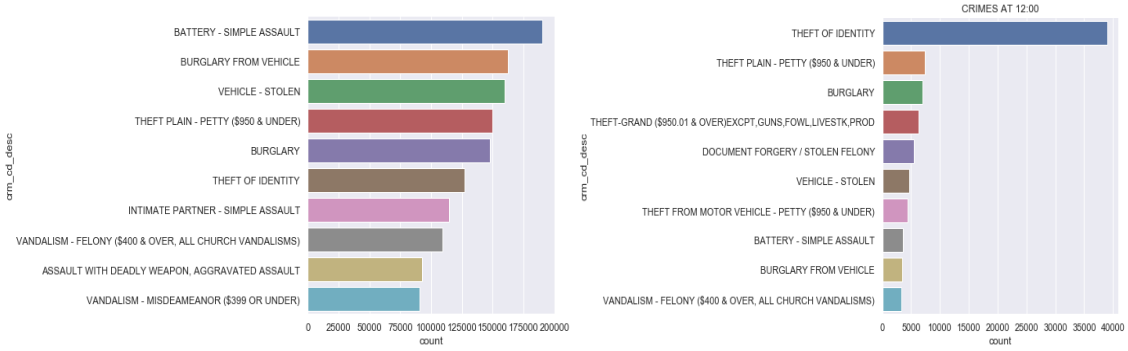
- Likewise, I analyzed the crime count with respect to month, day of month, day of week and time of day. Observation says that August had the highest number of crimes with respect to month. And 1st day of each month with respect to day of month, Friday with respect to day of week and at 12:00 clock with respect to time have the highest crime intensity.
- I analyzed the factors sex, descent, age of victims and how the crimes vary depends on those factors. Summary results that males experienced more victimization than females, crime intensity is high on victims of age between 20-50 and Hispanic/Latin/Mexican, White and Black are the top categories where victims are present. And I also analyzed the crimes on victims' descent in each area.

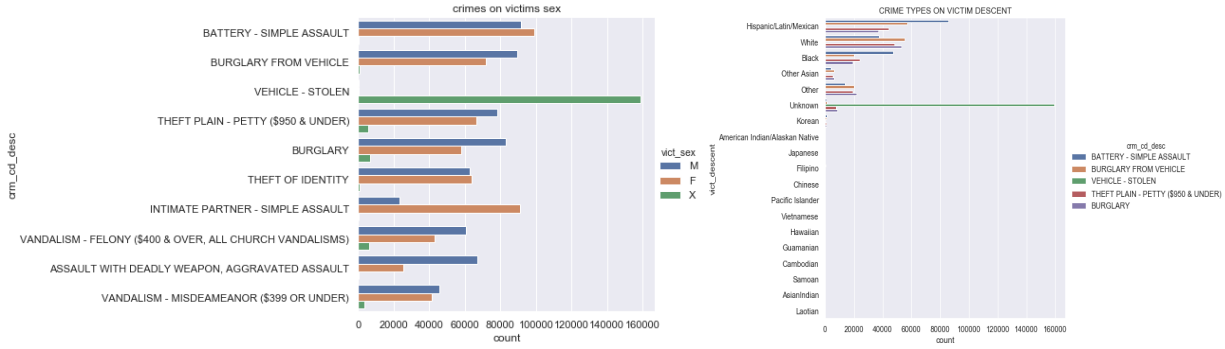


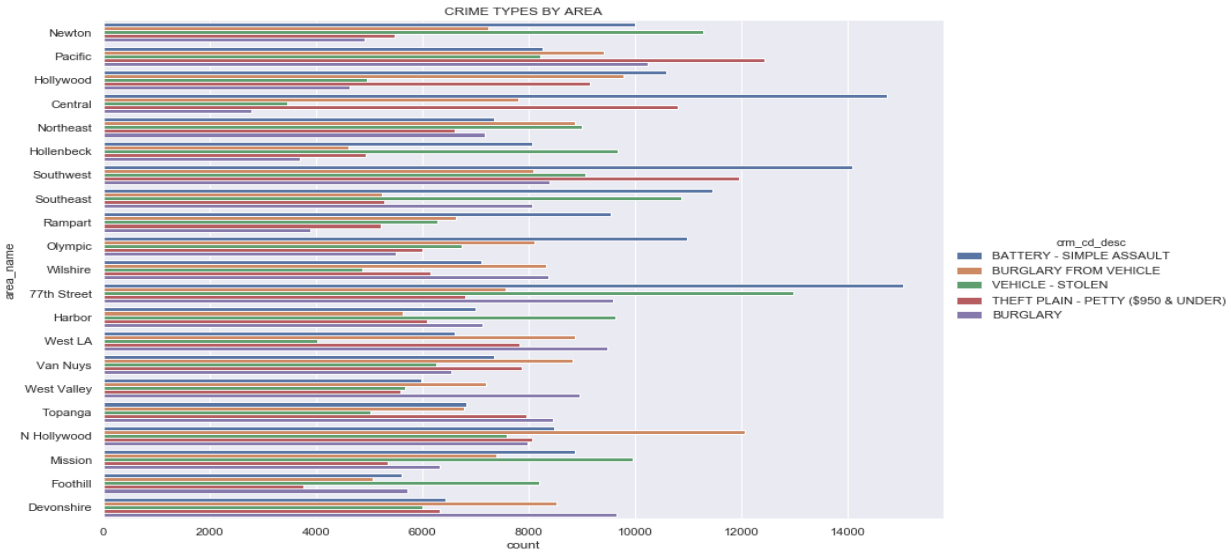
- I analyzed the crimes of victims of age above 70 years to know what type of crimes were done frequently on them. Observation results that BURGLARY is the primary crime happened more frequently followed by THEFT OF IDENTITY.



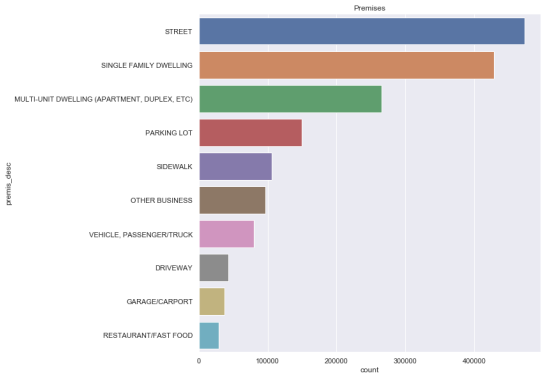
- To dive deeper into the data, I focused on only top 10 crime types with respect to area, victims' sex, victim's descent, time. This gives me a broader analysis of crime types.

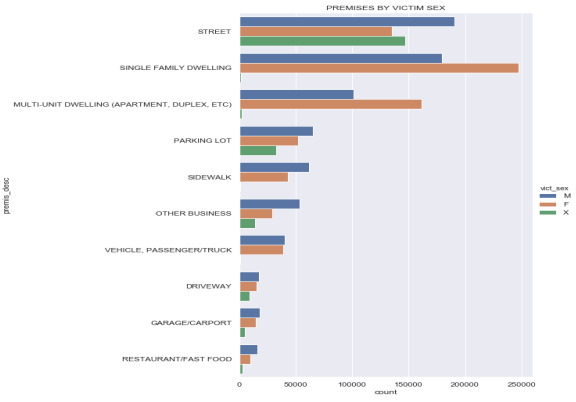


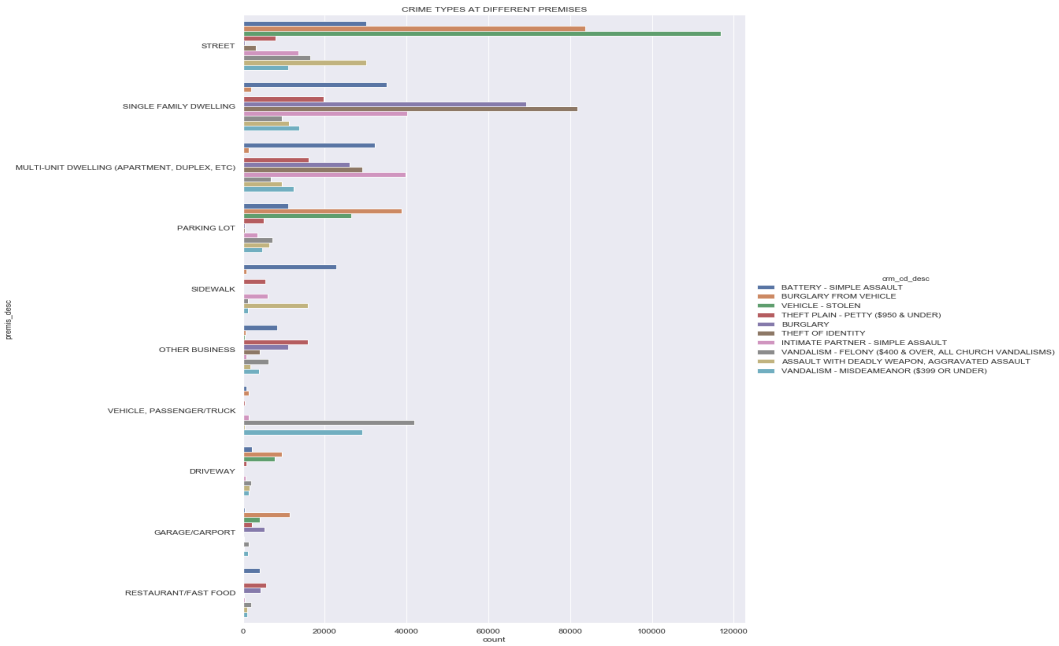




- I also analyzed the locations that are more prone to crimes with respect to victim sex and focused on the top 10 locations for broader analysis. Summary findings gave some interesting details about types of crimes at different premises. VEHICLE-STOLEN and BURGGLARY FROM VEHICLE are the top most crimes happened at the top crime location STREET.







- By analyzing the crimes at different locations, summary findings that STREET, SINGLE FAMILY DWELLING were the location where most of the crimes happened. And I witnessed that crime types vary depending on the location. I made some hypotheses to prove it statistically.

HYPOTHESIS:

- Null hypothesis(H0): There is no relationship between crime type and location.
- Alternate hypothesis(H1): There is a relationship between crime type and location.

The Pearson's Chi-Squared test:

- The Chi-Squared test is a statistical hypothesis test that assumes (the null hypothesis) that the observed frequencies for a categorical variable match the expected frequencies for the categorical variable. The result of the test is a test statistic that has a chi-squared distribution and can be interpreted to reject or fail to reject the assumption or null hypothesis that the observed and expected frequencies are the same.
- The chi-square test of independence works by comparing the categorically coded data that you have collected (known as the observed frequencies) with the frequencies that you would expect to get in each cell of a table by chance alone (known as the expected frequencies).
- If p-value <= alpha: significant result, reject null hypothesis (H0), dependent.
- If p-value <= alpha: significant result, reject null hypothesis (H0), dependent.
- Results we got are:

probability=0.950, critical=45331.701, stat=4325092.187

Dependent (reject H0)

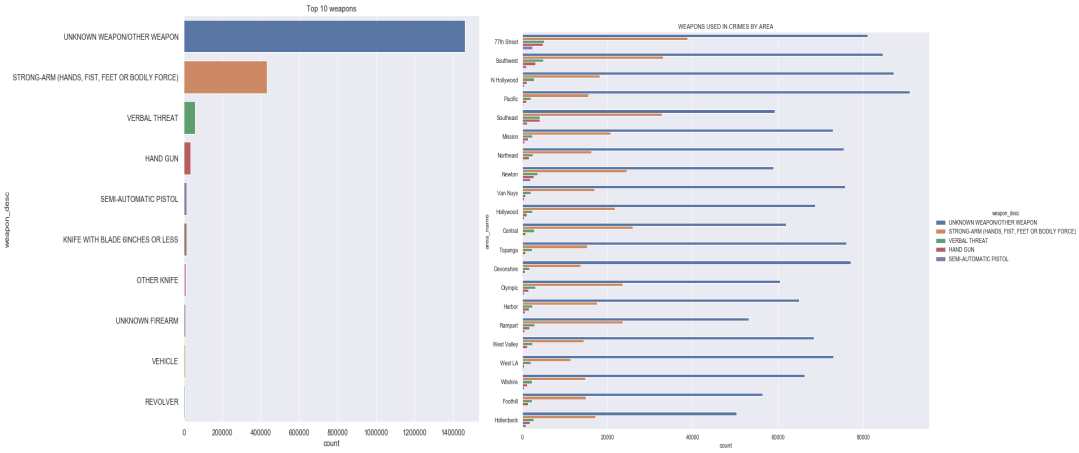
significance=0.050, p=0.000

Dependent (reject H0)

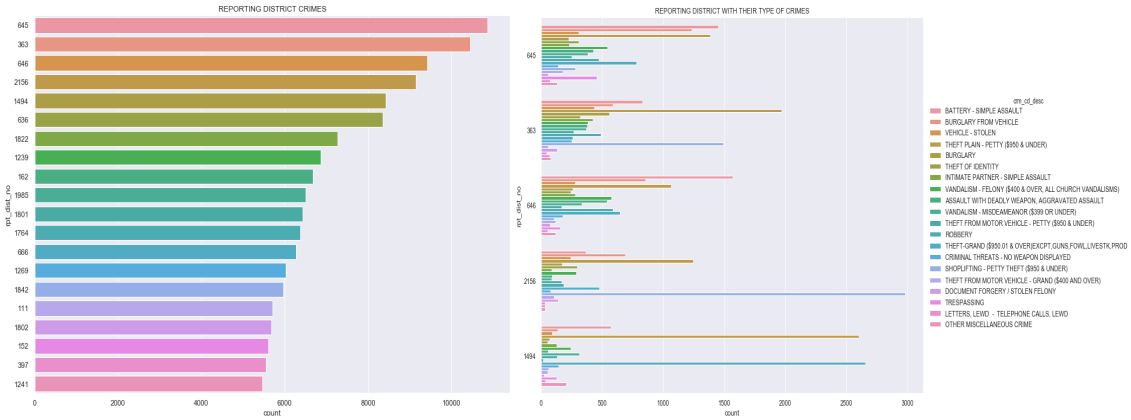
- From above results:

  As we have rejected the H0 that there is no relationship between these two variables, **we can conclude that there is a relationship between crime and the location.**

- I also analyzed the weapons used in crimes with respect to location, area, victim sex, victim descent, summary findings that in most of the crimes, criminal did not use any weapon, or the weapon is UNKNOWN.

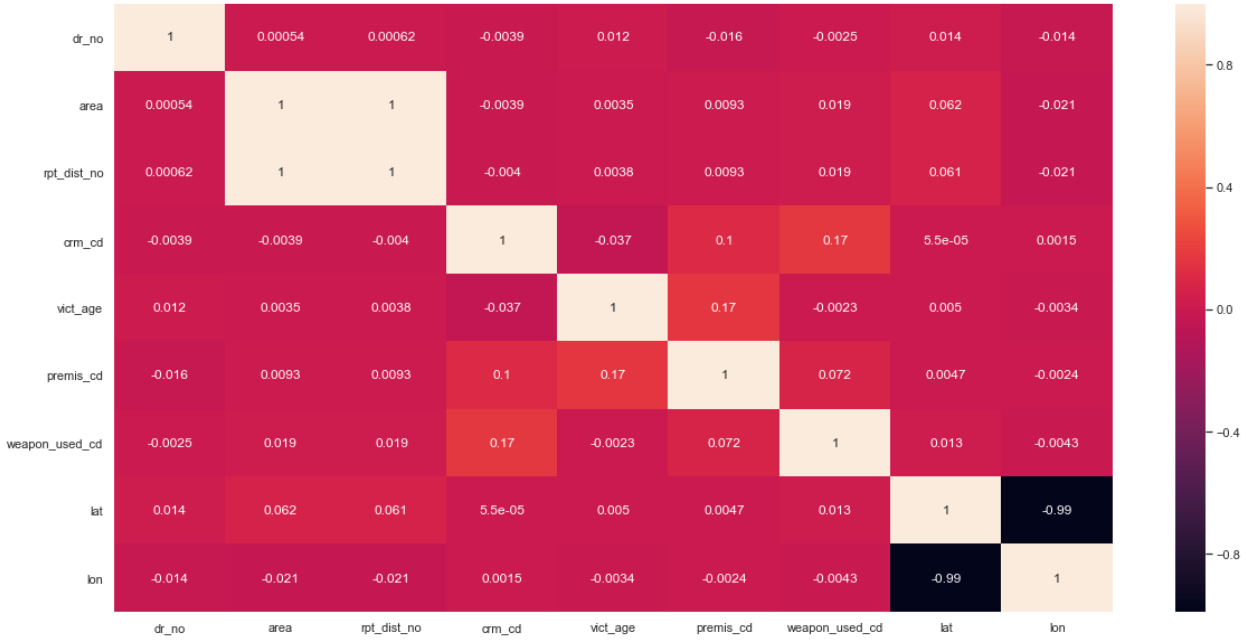


- I analyzed the count of crimes reported in each district and crime types with respect to each district. Observation results that the district 645 had more reported crimes with the primary crime type as BATTERY - SIMPLE ASSAULT.



- At last I analyzed the relationship between numerical features using correlation matrix (Figure 1) to select and remove collinear features (features that are more than 95% correlated for removal).

Figure 1 : Heat map



EDA Conclusion:

We have discovered so many interesting insights and trends in data through exploratory data analysis. We discovered the higher crime intensity areas, year, month, day and time. Observed the age group where most of the crimes happened with respect to their sex and descent. And discovered that in most of the crimes, criminals did not use any weapons. Analyzed the most prone crime areas, locations and how crime type varies depends on the location and area. By analyzing this we got to know that the prediction of crime type depends on location, area, time and some other factors. With this keep in mind we are moving on to the preprocessing and training step.

4. Data Preprocessing

4.1 Feature extraction:

In this step we added extra features that are extracted from other features like from ‘time_occ’ we extracted hours, minutes features, etc and dropped those unnecessary features(['date_rptd','date_occ','time_occ','mocodes','rpt_dist_no'].Also dropped the columns('crm_cd','premis_cd','weapon_used_cd','status') to reduce the redundancy. Redundant features slow down the training process. Because these code columns already have their description columns, we need to choose either one of them. I choose the description columns because the column with code values doesn’t give any information to the model except giving ordered data to model.

4.2 Stratified Sampling:

Since the dataset is too large with 2000000+ records, I took samples from data to make processing easy and save computational time. I did stratified sampling to take a sample of 20000 from the population. Stratified sampling is a method of sampling from a population which can be partitioned into subpopulations." This method of sampling can be advantageous because it tries to keep in the sample the same proportion of each desired variable (strata) that is present in the population. A simple random sample could ignore this fact.

4.3 Handling Rare Categories:

Though we hope to classify every single type of crime, it is not really practical since some crimes are happening at extremely low frequency as low as 1. Based on the sorted list of crimes, we can see that some crimes are really rare, though these rare records won't be greatly useful to our model, we choose the categories that occur less than or equal to 6 are moved to OTHER CRIME category for simplicity.

4.4 Create dummy features for categorical variables:

Since most machine learning models only accept numeric variables, preprocessing the categorical variables is mandatory. We need to convert these categorical variables to numbers such that the model is able to understand and extract valuable information. Since our categorical data doesn’t have any inherent order and has many categories as high as 142, I choose binary encoding.

- **Binary encoding** is a combination of Hash encoding and one-hot encoding. In this encoding scheme, the categorical feature is first converted into numerical using an ordinal encoder. Then the numbers are transformed in the binary number. After that binary value is split into different columns. Binary encoding works really well when there are a high number of categories.

4.5 Stratified Train-Test Splits:

I split the data into 80% train data and 20% test data. By splitting the data in this way, we can improve model generalization to the new data. Since our dataset is highly imbalanced dataset, I did stratified train test split that preserves the same proportions of examples in each class as observed in the original dataset.

4.6 Scale the data to prepare for model creation:

I standardize the data using StandardScaler from sklearn.preprocessing.Standardization is useful when our data has varying scales and the algorithms we are using does make assumptions about our data having a Gaussian distribution, such as linear regression, logistic regression, and linear discriminant analysis.

5. Modeling:

In this part, we are trying to build two models

1.Multiclass classification for all the crime types including low frequency crimes.

2.Multiclass classification for high frequency (top 10) crimes.

Since target is a categorical variable with many classes and data is labelled data, it is a **Supervised multi-class classification** problem. The dataset is **imbalanced** because the classes in target have an unequal distribution. For modeling I choose to work with a machine learning library - **scikit.learn**.

Metrics: Choosing the right metrics is the key to assess the performance of a model. I choose to take "**weighted" F1 score**. For multi-class problems with imbalance data, we have to average the F1 scores for each class. The weighted F1 score averages the F1 score for each class by taking the class imbalances into account. In other words, the number of occurrences of each class does figure into calculation when using "weighted" score.

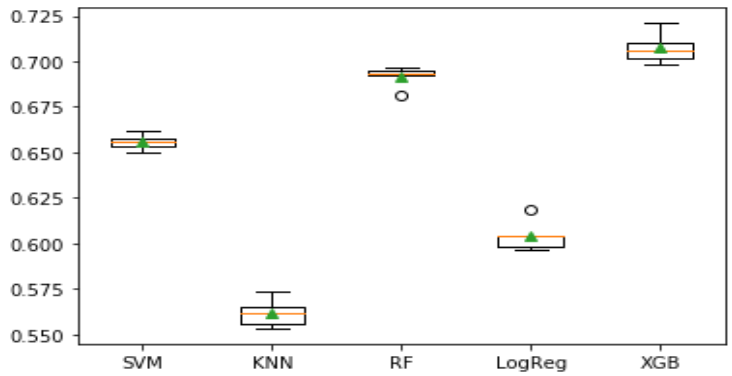
1.MULTICLASS CLASSIFICATION FOR ALL THE CRIMES INCLUDING LOW FREQUENCY CRIMES

It would be a good idea to spot check a suite of different nonlinear algorithms on a dataset to quickly flush out what works well and deserves further attention and what doesn't.

I evaluated the following machine learning models:

Support Vector Machine (SVM)
k-Nearest Neighbors (KNN)
LogisticRegression (LogReg)
Random Forest (RF)
XGBoost(XGB)

The output results are:



Even though XGBoost shows higher accuracy than other models we are going to choose a random forest classifier to try as our model because of its fast run time speed.

Random Forest Classifier:

It is an ensemble tree-based learning algorithm. The Random Forest Classifier is a set of decision trees from a randomly selected subset of training set. It aggregates the votes from different decision trees to decide the final class of the test object.

Model-1:

To assess our model, I used Stratified K-Folds cross-validation - This cross-validation object is a variation of KFold that returns stratified folds. The folds are made by preserving the percentage of samples for each class.

I got an **F1-score(weighted) - 0.67** with the model using a Random Forest classifier with StratifiedKFold of 5 splits. Figure-1 shows the distribution of predicted probabilities.

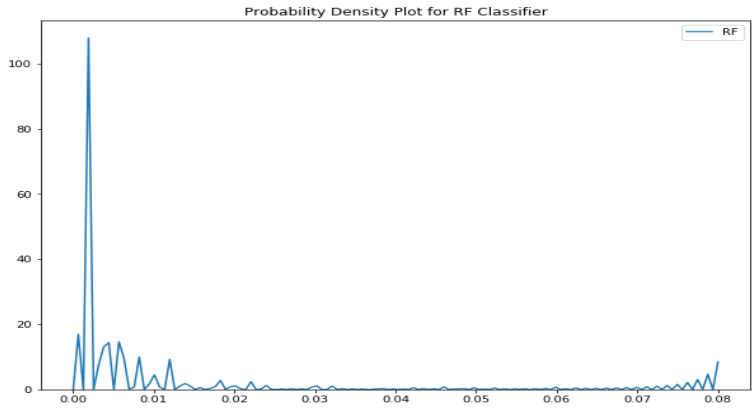


Figure-1

Because our dataset is severely imbalance, we are going to trying other imbalanced methods. There are four types of imbalanced classification techniques to spot check:

- Cost-Sensitive Algorithms
- Data Resampling Algorithms
- One-Class Algorithms
- Probability Tuning Algorithms
- Here I tried the first two methods.

Cost-Sensitive Algorithms

Cost-sensitive algorithms are modified versions of machine learning algorithms designed to take the differing costs of misclassification into account when fitting the model on the training dataset.

Model-2:

A **cost-sensitive** version of random forest with custom class weightings was found to achieve better performance. I computed the class weights of the target variable using the library **class_weight** from sklearn.utils and passed it to the class_weight parameter in RandomForest classifier. And I got the resultant accuracy **F1-score(weighted) - 0.68(**increased by 1%)**.**Figure-2 shows the predicted probabilities of a model with class weights.

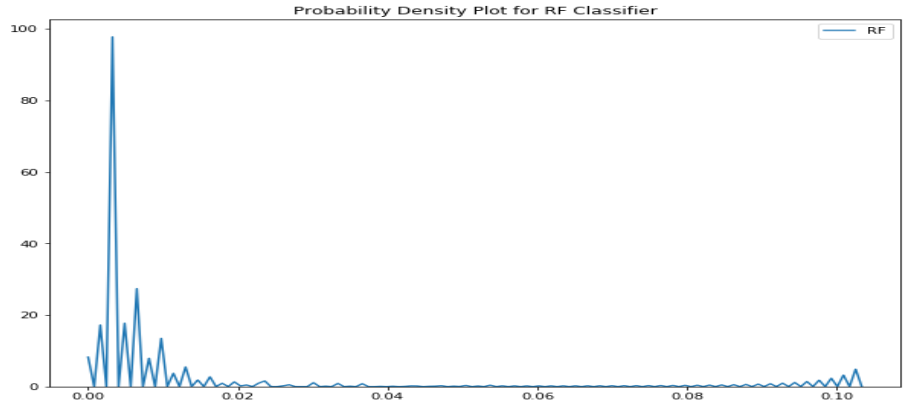


Figure-2(Model-2)

I tried and tested the second imbalance method which is a data resampling algorithm.

Data Resampling Algorithms

Resampling methods are designed to add or remove examples from the training dataset in order to change the class distribution. Two approaches to make a balanced dataset out of an imbalanced one are under-sampling and oversampling.

- ### Under-sampling

  Under-sampling balances the dataset by reducing the size of the abundant class. This method is used when quantity of data is sufficient. By keeping all samples in the rare class and randomly selecting an equal number of samples in the abundant class, a balanced new dataset can be retrieved for further modelling.

- ### Over-sampling

  On the contrary, oversampling is used when the quantity of data is insufficient. It tries to balance the dataset by increasing the size of rare samples. Rather than getting rid of abundant samples, new rare samples are generated by using e.g. repetition, bootstrapping or SMOTE (Synthetic Minority Over-Sampling Technique).

Here I tried the combination of under and over sampling which is **SMOTETomek.**

SMOTE is an oversampling method that synthesizes new plausible examples in the majority class. **Tomek Links** refers to a method for identifying pairs of nearest neighbors in a dataset that have different classes. Removing one or both of the examples in these pairs (such as the examples in the majority class) has the effect of making the decision boundary in the training dataset less noisy or ambiguous.

Model-3:

I built a model with **SMOTETomek** training data using RandomForest Classifier and achieved an accuracy **F1-score(weighted) - 0.71.** Figure-3 shows the predicted probabilities**.** Even though we achieved a pretty good across for this highly imbalanced data, SMOTETomek is computationally so expensive.

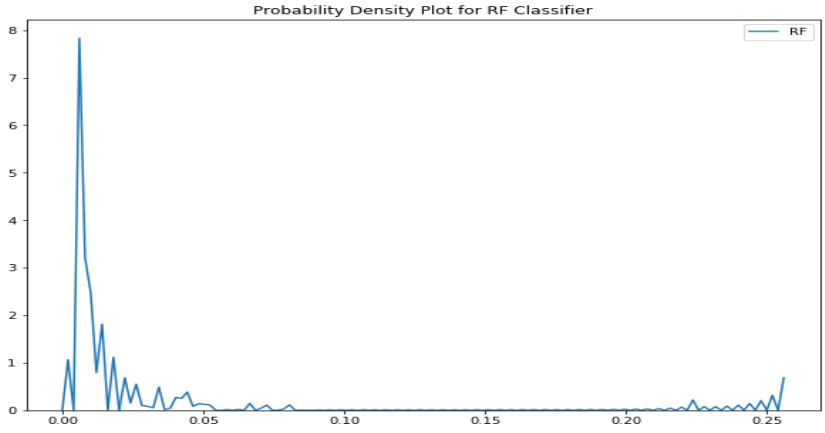


Figure - 3(Model-3)

Model-4

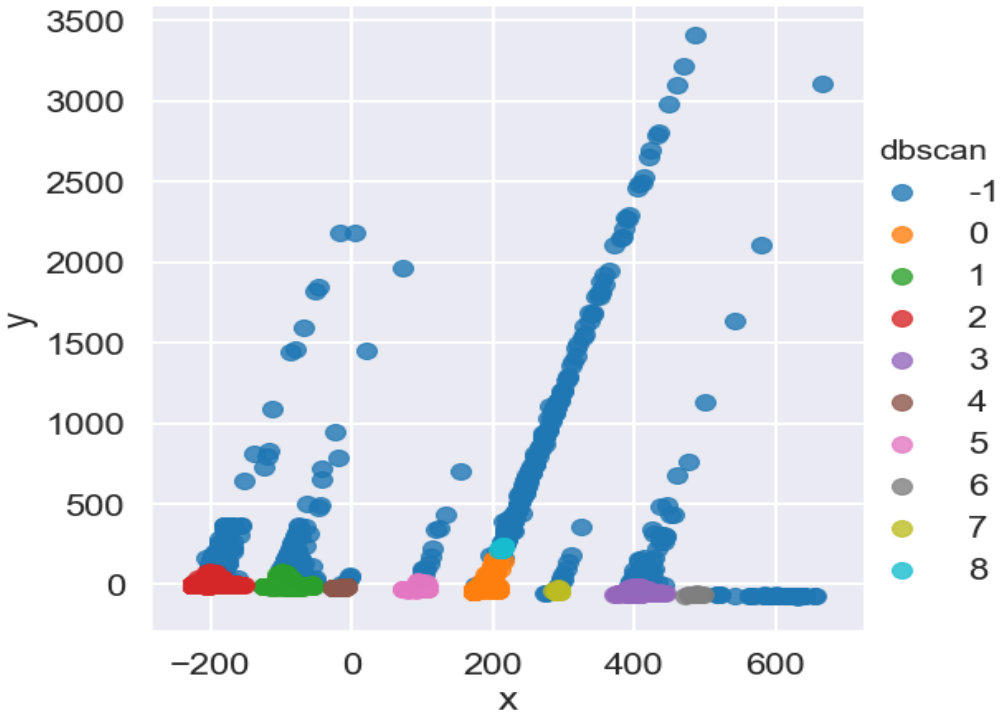
In this part I approached a different method with the combination of supervised and unsupervised learning. First, I tried to remove anomalies and cluster the data.

- **Outliers** or **anomalies** are rare examples that do not fit in with the rest of the data. Identifying outliers in data is referred to as outlier or anomaly detection and a subfield of unsupervised machine learning algorithm that attempts to model “normal” examples in order to classify new examples as either normal or abnormal (e.g. outliers).
- **Cluster** analysis, or **clustering**, is an unsupervised machine learning task. It involves automatically discovering natural grouping in data. Unlike supervised learning (like predictive modeling), **clustering** algorithms only interpret the input data and find natural groups or **clusters** in feature space.
- **Anomaly detection and Clustering**: Here I choose to work with DBSCAN because it works best for detecting anomalies while clustering the data.

DBSCAN is based on point density rather than distance. It groups together points with many nearby neighbors. DBSCAN is one of the most cited algorithms in the literature. It does not require knowing the number of clusters a priori but does require specifying the neighborhood size. And it is also useful for outlier detection.

Choosing the optimum epsilon and right number of min_sample is the key part. I choose them using **silhouette score**. The silhouette score is computed on *every datapoint in every cluster*. The silhouette score ranges from -1 (a poor clustering) to +1 (a very dense clustering) with 0 denoting the situation where clusters overlap.

The below figure shows the clusters with anomalies as ' -1'. I removed the anomalies which are cluster '-1' from the clustered data.



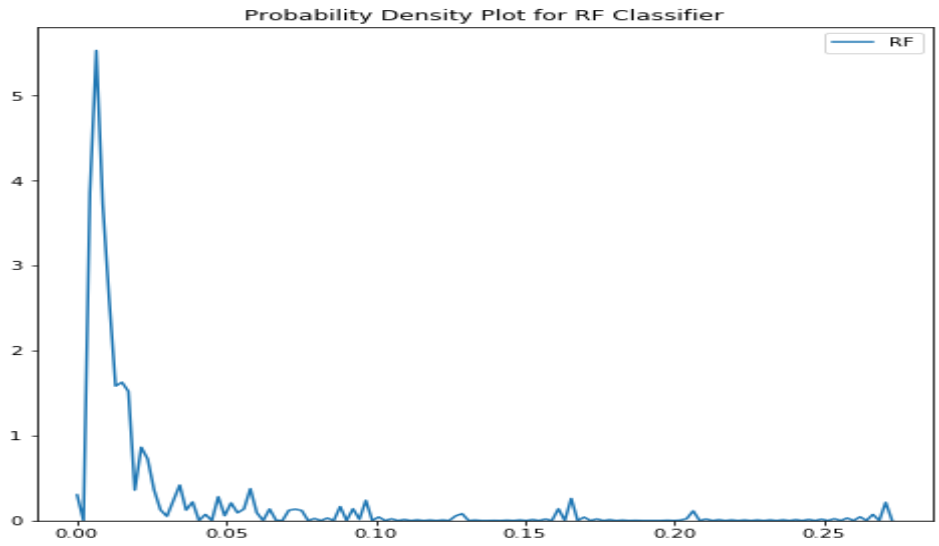
After DBSCAN we got the clustered data without anomalies and we are all set to use this data to perform supervised algorithms.

I built the 3 models with this clustered data using the above 3 algorithms same as we did earlier with unclustered data i.e. RandomForest Classifier, Cost sensitive method, Resampling algorithm. And we got F1-score(weighted) as 0.66,0.67,0.70 respectively. We can clearly say that this combination method did not improve the performance particularly on this highly imbalanced data.

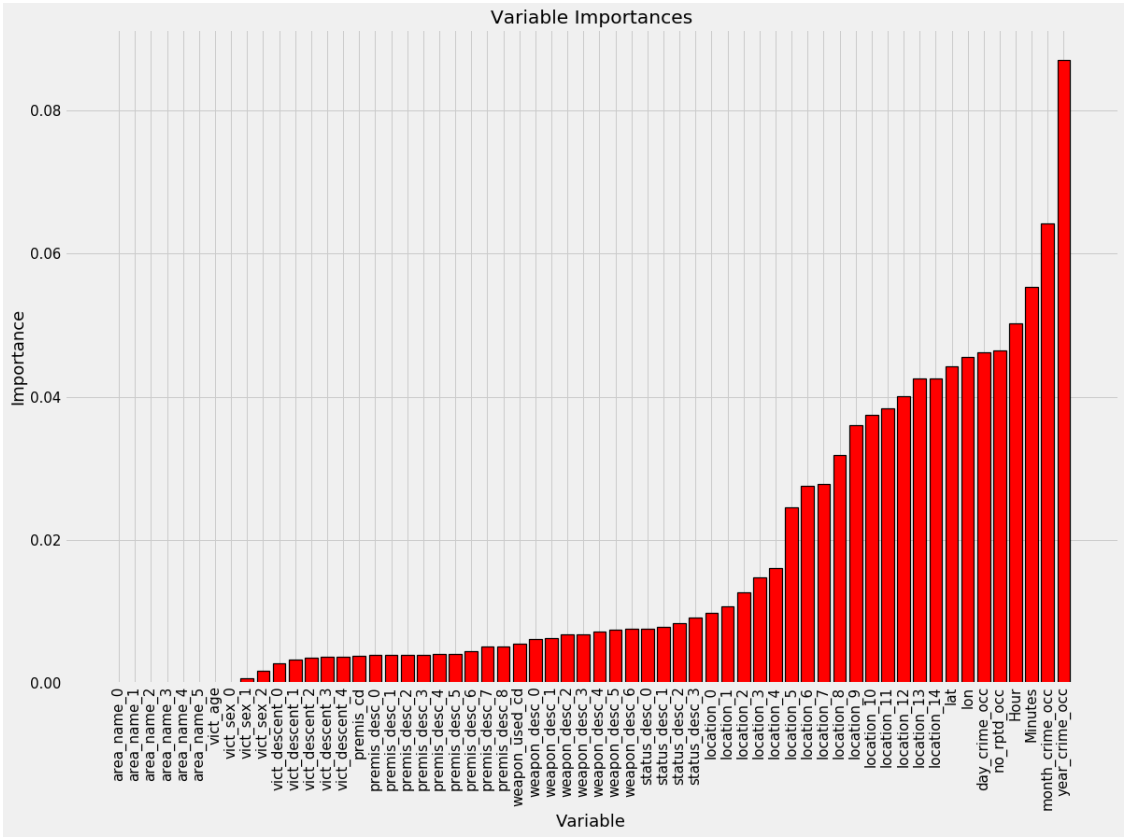
Choosing the best model:

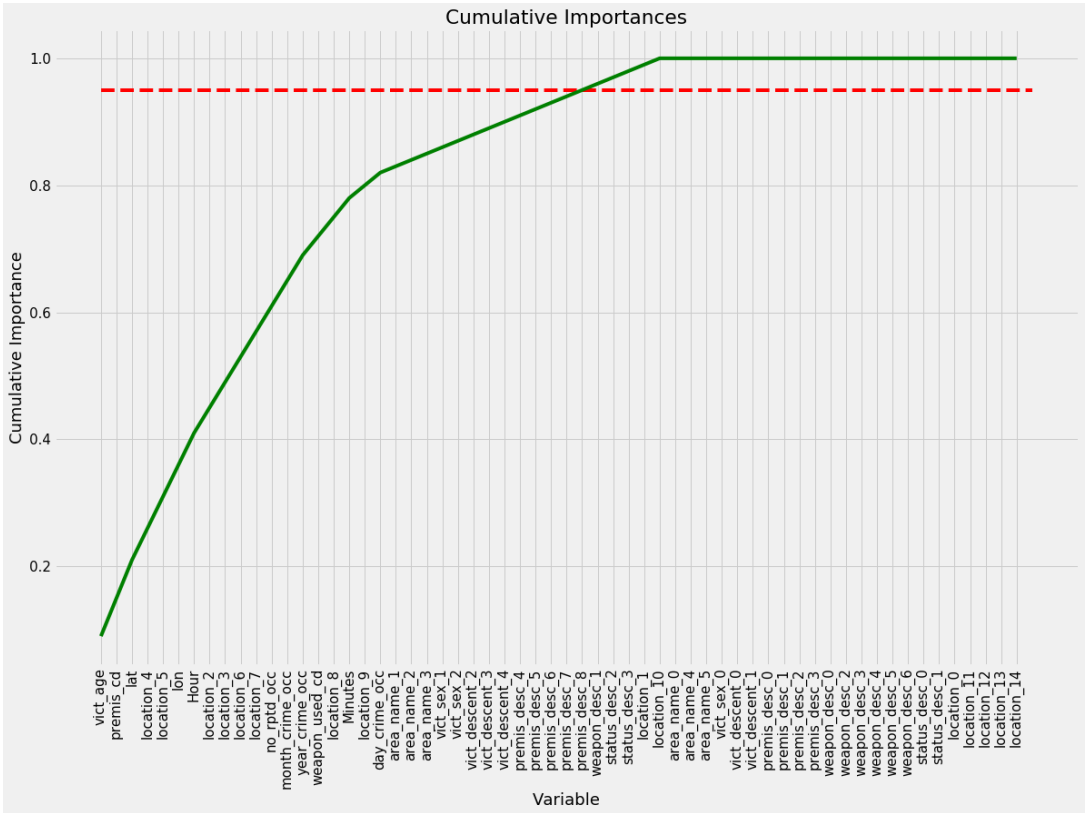
Model	Algorithm	f1 score(weighted)	Precision	Recall	Training	Testing
Without Cluster	Random Forest	0.665	0.687	0.692	1.000	0.692
Without Cluster	CostSenstivity	0.684	0.712	0.705	1.000	0.705
Without Cluster	SMOTETomek	0.708	0.719	0.720	0.992	0.720
With Cluster	Random Forest	0.662	0.6815	0.689	1.000	0.689
With Cluster	CostSenstivity	0.678	0.705	0.701	1.000	0.701
With Cluster	SMOTETomek	0.703	0.713	0.716	0.992	0.716

From the above table, we found the best model as SMOTETomek (without cluster). Even though SMOTETomek outperforms all other models I choose next best model Cost Sensitivity (without cluster) method as the final model for this project because SMOTETomek is computationally so expensive. Next I did hyperparameter tuning for best parameters to improve the performance of our model. With the best parameters that we got after tuning we were able to increase the accuracy by 3% in other words achieved a pretty good accuracy F1-score(weighted) as 0.71. The below figure shows the predicted probabilities resulted by final model.



Next I reviewed the final model to determine feature importance. Below are the plots that show the feature importance and their cumulative sum.


Variable Importances
Importance
0.00
0.02
0.04
0.06
0.08
area_name_0
area_name_1
area_name_2
area_name_3
area_name_4
area_name_5
vict_age
vict_sex_0
vict_sex_1
vict_sex_2
vict_descent_0
vict_descent_1
vict_descent_2
vict_descent_3
vict_descent_4
premis_cd
premis_desc_0
premis_desc_1
premis_desc_2
premis_desc_3
premis_desc_4
premis_desc_5
premis_desc_6
premis_desc_7
premis_desc_8
weapon_used_cd
weapon_desc_0
weapon_desc_1
weapon_desc_2
weapon_desc_3
weapon_desc_4
weapon_desc_5
weapon_desc_6
status_desc_0
status_desc_1
status_desc_2
status_desc_3
location_0
location_1
location_2
location_3
location_4
location_5
location_6
location_7
location_8
location_9
location_10
location_11
location_12
location_13
location_14
lat
lon
day_crime_occ
no_rptd_occ
Hour
Minutes
month_crime_occ
year_crime_occ
Variable


Cumulative Importances
Cumulative Importance
0.2
0.4
0.6
0.8
1.0
vict_age
premis_cd
lat
location_4
location_5
lon
Hour
location_2
location_3
location_6
location_7
no_rptd_occ
month_crime_occ
year_crime_occ
weapon_used_cd
location_8
Minutes
location_9
day_crime_occ
area_name_1
area_name_2
area_name_3
vict_sex_1
vict_sex_2
vict_descent_2
vict_descent_3
vict_descent_4
premis_desc_4
premis_desc_5
premis_desc_6
premis_desc_7
premis_desc_8
weapon_desc_1
status_desc_2
status_desc_3
location_1
location_10
area_name_0
area_name_4
area_name_5
vict_sex_0
vict_descent_0
vict_descent_1
premis_desc_0
premis_desc_1
premis_desc_2
premis_desc_3
weapon_desc_0
weapon_desc_2
weapon_desc_3
weapon_desc_4
weapon_desc_5
weapon_desc_6
status_desc_0
status_desc_1
location_0
location_11
location_12
location_13
location_14
Variable

I reduced the number of features in use by the model to only those required to account for 95% of the importance. The model with only important features we were able to reduce the run time but accuracy got decreased by 3%(F1-score(weighted) - 0.68).

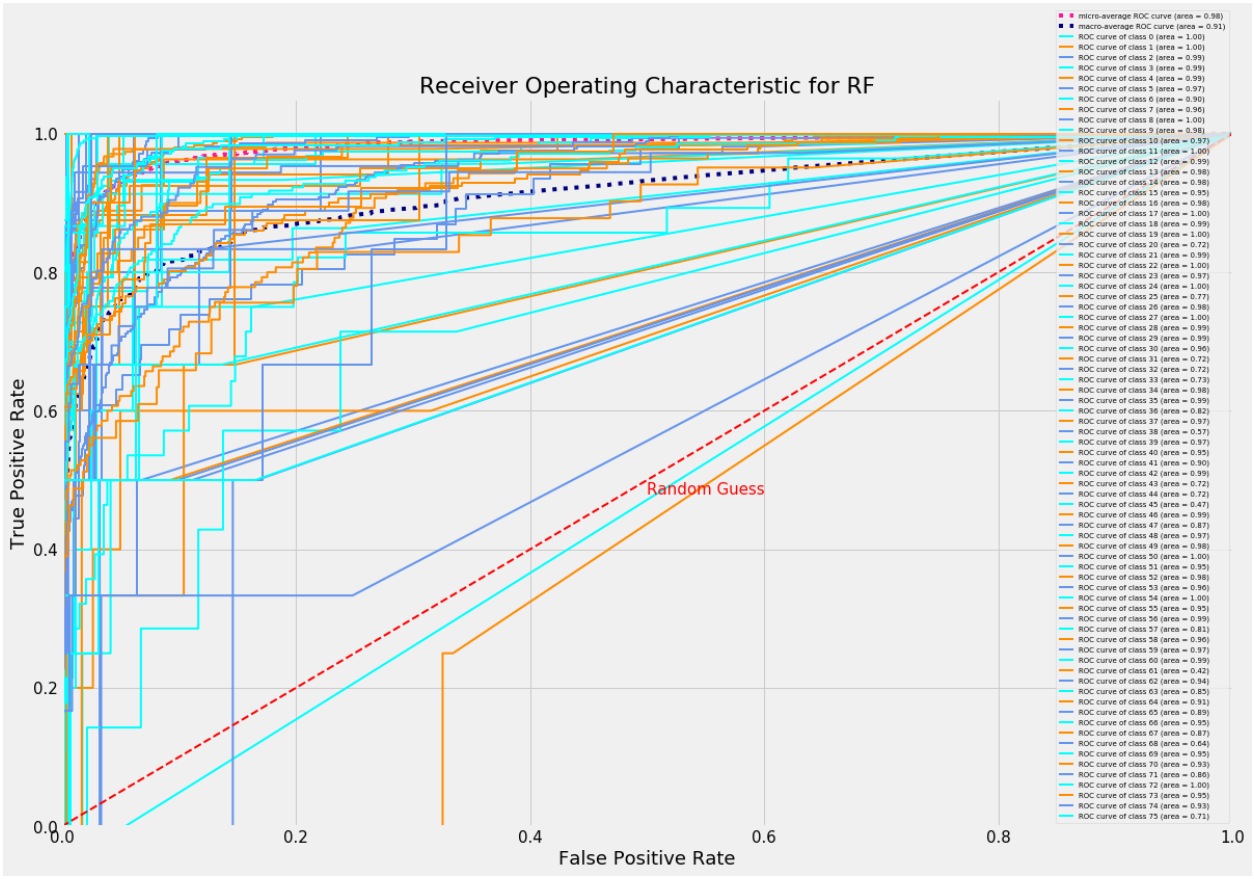
Model with all the features:
CPU times: user 2min 6s, sys: 7.04 s, total: 2min 13s
Wall time: 1min 50s
f1_score : 0.71

Model with only important features:
CPU times: user 1min 58s, sys: 6.64 s, total: 2min 5s
Wall time: 22.8 s
f1_score: 0.68

So I need to tradeoff between performance runtime and accuracy depending on the requirements.
For this project I considered accuracy(f1-score) as an important metric so we are going to take the model with all the features as our final model.

AUC-ROC for Multi-Class Classification

The AUC-ROC curve is only for binary classification problems. But we can extend it to multiclass classification problems by using the One vs All technique.
So, if we have three classes 0, 1, and 2, the ROC for class 0 will be generated as classifying 0 against not 0, i.e. 1 and 2. The ROC for class 1 will be generated as classifying 1 against not 1, and so on.



2.MULTICLASS CLASSIFICATION FOR HIGH FREQUENCY (TOP 10) CRIMES

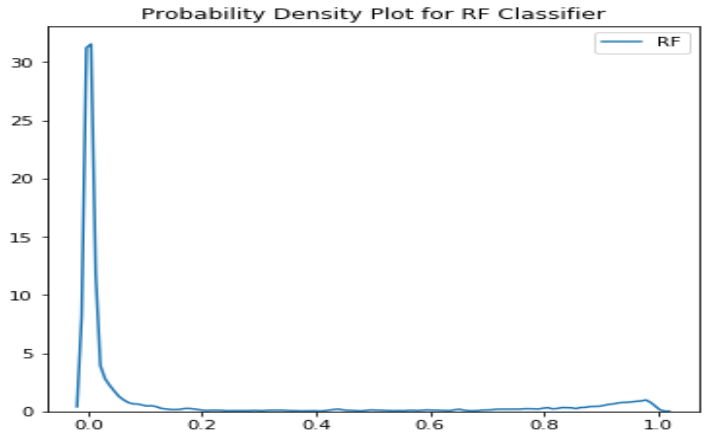
Since we choose only the high frequency (top 10) crimes, all the classes in the response variable have merely an equal distribution, so the dataset is not imbalanced anymore. I evaluated the following machine learning models :

Support Vector Machine (SVM)
k-Nearest Neighbors (KNN)
LogisticRegression (LogReg)
Random Forest (RF)
XGBoost (XGB)

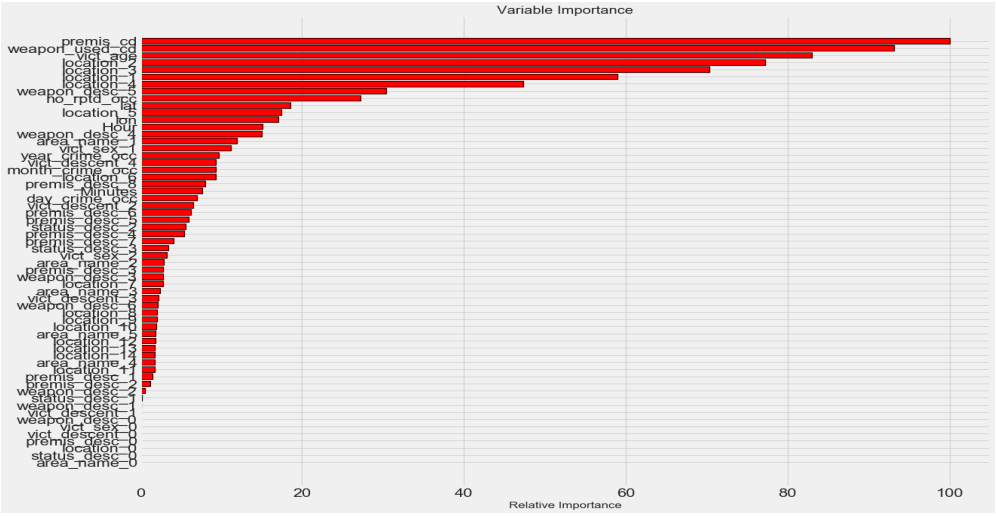
The output results are:

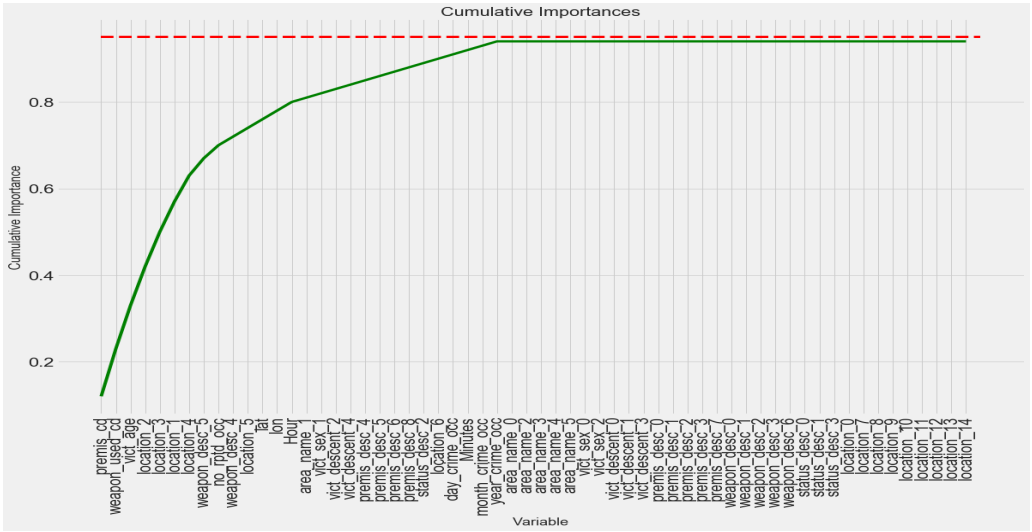
Model	Accuracy(mean)	Accuracy(std)
SVM	0.656	0.0041
KNN	0.562	0.0072
RF	0.692	0.0053
LogReg	0.604	0.0084
XGB	0.708	0.008

Again, here I am a little biased to RandomForest because of its fast runtime speed. So, I choose RF as our model even though XGB outperforms RF. I built a model using RandomForest with StratifiedKFold(5 splits) and achieved a f1-score as 0.831.After that I did hyperparameter tuning for best parameters to improve the performance of our model. With the best parameters that we got after tuning we were able to increase the accuracy by 1% in other words achieved a pretty good accuracy **F1-score - 0.84**. The below figure shows the predicted probabilities resulted by final model.



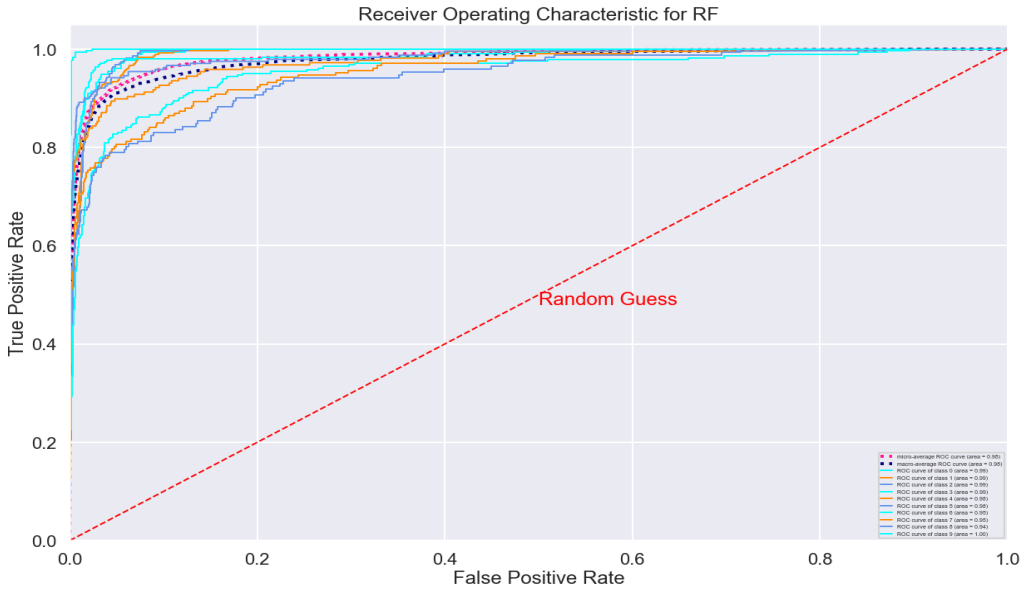
Next I reviewed the final model to determine feature importance. Below are the plots that show the feature importance and their cumulative sum.





AUC-ROC for Multi-Class Classification

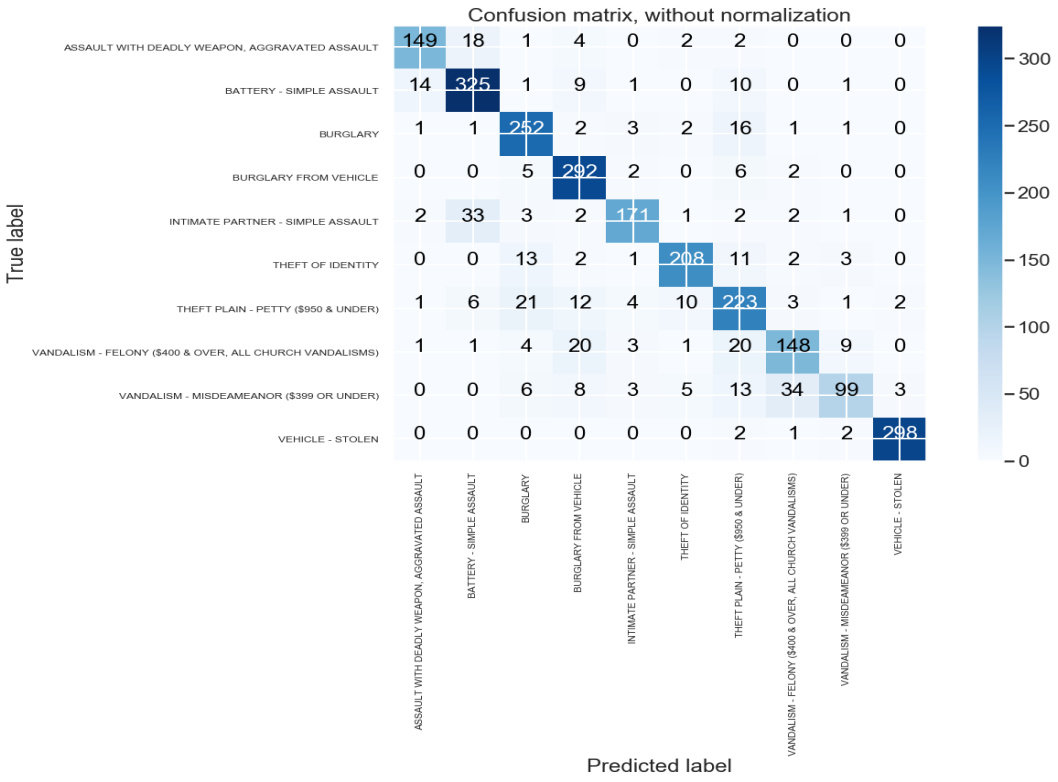
The AUC-ROC curve is only for binary classification problems. But we can extend it to multiclass classification problems by using the One vs All technique.



Predictions

Next I reviewed the predictions using a confusion matrix. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

Confusion matrix



6. Conclusion

Our work started with the Data cleaning process, during which we changed the original data column names to more into a more accessible format to make data operations manageable. We also added several useful columns to the original dataset (such as year occurred, month occurred, etc). And dropped the columns which have more than 90% NaN's. In Exploratory Data Analysis section, we analyzed the dataset using plots such as bar plots, box plots and histograms. Furthermore, this section has figure out other significant analysis about our dataset, regarding for example the most frequent crimes and weapons used in the city of LA. We also show interesting statistics about crime distribution over years, months, timeslots and weekdays, and some of them were a bit surprising. During the analysis we figured out the outliers and collinear features and removed them. We also revealed that several features, for example location of the crime and the time of the crime, are associated with the type of crime, providing the basis for our modeling later. One single feature may not be sufficient for determining the type of crime, but a combination of various features can be powerful. This is indeed the case in our modeling.

In our modeling, we started with multiclass classification to predict all the crimes including low frequency crimes. And we evaluated several algorithms like SVM, RF, KNN, LogReg (Logistic Regression),XGB with metrics as F1-score(weighted).Even though XGB outperforms RF, we choose RF as our model because

of its fast runtime speed and achieved 67%.Since dataset is highly imbalanced, we tried the two of the imbalanced methods. One is Cost sensitive method and other is data sampling algorithm (**SMOTETomek**). And achieved performance accuracy as 68%,71% respectively.

Then we approached a different method - Classification with DBSCAN (combination of supervised and unsupervised learning algorithms). This was hardest part of our work because we need to find the optimum epsilon and min_sample values for DBSCAN. In this method we used unsupervised method DBSCAN for clustering (also useful to detect anomalies while clustering) and removed the anomalies from the dataset. After that again we built 3 models using the above methods and achieved 66%,68%,70% respectively. So, our Classification with DBSCAN approach did not any significant results. And we choose the best model as cost sensitive (without cluster) even though SMOTETomek got higher accuracy than cost sensitive because SMOTETomek is computationally so expensive. And we did hyperparameter tuning for better parameters to improve accuracy and were able to increase the performance accuracy by 3% (**F1-score(weighted) - 71%**) which pretty good score for highly imbalanced dataset.

In addition to multiclass classification for all crime types including low frequency crimes, we also tried multiclass classification for high frequency (top 10) crimes. Since we choose high frequency(top 10) crimes, all the classes in target have nearly the same frequency. We evaluated the several machine learning algorithms like SVM, RF, KNN, LogReg (Logistic Regression), XGB with metrics as F1-score(weighted).Even though XGB outperforms RF, here again I was little biased to RF because if its fast runtime speed we choose RF as our model and achieved accuracy (f1- score) 83%.After hyperparameter tuning we were able to increase the performance by 1% (**F1-score - 84%)**.

In conclusion, our work led to interesting results, analysis and statistics, but also provided useful tools both for authorities and population, which allows a better understanding of crimes in LA.

This project gave me an opportunity to explore this freely available dataset using a proper data science pipeline of data wrangling, data analysis, data visualization, prediction, and data storytelling.

7. Future Improvements

- In order to boost the classification accuracy, it is necessary to incorporate other information like Modus Operandi (MO) which we did not use in our model. Additionally, some events and the outcomes of the events may be associated with some crime types. Events information and weather information can also be incorporated. It will be interesting to see whether these features can help the classification.
- We could use a Clustering algorithm, namely K-Modes Clustering, which is similar to K-Means, but uses modes instead of means, making itself usable for cluster computation on categorical variables. As a result of the clustering process, we could figure out some *cluster centroids* which can potentially be interesting for authorities, since they can indicate "standard frequent crimes", allowing authorities to concentrate their forces in order to contrast crimes indicated by centroids and similar ones.
- Due to memory constraints on Jupyter notebook, I had to train a sample of size 20000 from the original dataset. Without resource limitations, I would love to train on the full dataset. Preliminary tests showed that the bigger the training size, the higher the accuracy.