NYPD Crime Data Analysis

Anoosha Seelam
lseelam@buffalo.edu
Computer Science and Engineering
University at Buffalo, NY

Viswapujitha Suresh
viswapuj@buffalo.edu
Computer Science and Engineering
University at Buffalo, NY

***Abstract*—**The goal of this project is to clean the data in a dataset, analyse it and generate a descriptive summary. Also we have done some data exploration with the dataset. The cleaned data is used to generate a descriptive summary for the features and contents. The dataset used (NYPD Shooting Incidents Dataset) includes all shooting crimes reported to the New York City Police Department (NYPD) from 2006 to the end of 2019.

I. INTRODUCTION

One among the major crime activities in New York is "shootings" and many have occurred until today. The city was expected to finish 2020 with a 14-year high in that category of violence, according to Police Commissioner Dermot Shea. People are not aware of these crimes and even after so many incidents till today and we haven't witnessed any decrease in this count. Victims range from 18 - 65 years old. Our dataset addresses a shooting episode in NYC and incorporates data about the occasion, the area, and season of the event. Likewise, data identified with suspect and casualty demographics which we'll be able to obtain useful information for our analysis. It will provide an insight to what age category the perpetrators were and who are the major category of victims being attacked.It would also provide information about the crime hotspots of New York City.

II. DATASET DESCRIPTION

A. Source

List of every shooting incident that occurred in NYC during the current calendar year. The dataset is a breakdown of every shooting incident that occurred in NYC .This data is manually extracted every quarter and reviewed by the Office of Management Analysis and Planning before being posted on the NYPD website. Each record represents a shooting incident in NYC and includes information about the event, the location and time of occurrence. In addition, information related to suspect and victim demographics is also included. This data can be used by the public to explore the nature of police enforcement activity.

link: *https://data.cityofnewyork.us/d/5ucz-vwe8*

B. DESCRIPTION OF DATASET

The dataset consists of 19 columns labelled and 21.6K rows.

TABLE I. DESCRIPTION OF DATASET

Field name	Description
INCIDENT_KEY	Randomly generated persistent ID for each incident
OCCUR_DATE	Exact date of the shooting incident
OCCUR_TIME	Exact time of the shooting incident
BORO	Borough where the shooting incident occurred
PRECINCT	Precinct where the shooting incident occurred
JURISDICTION_CODE	Jurisdiction where the shooting incident occurred. Jurisdiction codes 0(Patrol), 1(Transit) and 2(Housing) represent NYPD whilst codes 3 and more represent non NYPD jurisdictions
LOCATION_DESC	Location of the shooting incident
STATISTICAL_MURDER_FLAG	Shooting resulted in the victim's death which would be counted as a murder
PERP_AGE_GROUP	Perpetrator's age within a category
PERP_SEX	Perpetrator's sex description
PERP_RACE	Perpetrator's race description
VIC_AGE_GROUP	Victim's age within a category
VIC_SEX	Victim's sex description
VIC_RACE	Victim's race description
X_COORD_CD	Midblock X-coordinate for New York

	State Plane Coordinate System, Long Island Zone, NAD 83, units feet (FIPS 3104)
Y_COORD_CD	Midblock Y-coordinate for New York State Plane Coordinate System, Long Island Zone, NAD 83, units feet (FIPS 3104)
Latitude	Latitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)
Longitude	Longitude coordinate for Global Coordinate System, WGS 1984, decimal degrees (EPSG 4326)

III. DATA CLEANING

Task 1 - Removing Duplicates

It is given in the dataset footnotes that a shooting incident can have multiple victims involved and as a result duplicate INCIDENT_KEY's are produced.Each INCIDENT_KEY represents a victim but similar duplicate keys are counted as one incident. So, duplicate rows are checked and removed if any.

Task 2 - Conversion of object data to categorical data

The object data is converted to categorical data to perform encoding.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 17031 entries, 0 to 21624
Data columns (total 19 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   INCIDENT_KEY            17031 non-null  int64
 1   OCCUR_DATE              17031 non-null  object
 2   OCCUR_TIME              17031 non-null  object
 3   BORO                    17031 non-null  category
 4   PRECINCT                17031 non-null  int64
 5   JURISDICTION_CODE       17029 non-null  float64
 6   LOCATION_DESC           7124 non-null   category
 7   STATISTICAL_MURDER_FLAG 17031 non-null  bool
 8   PERP_AGE_GROUP          10662 non-null  category
 9   PERP_SEX                10692 non-null  category
 10  PERP_RACE               10692 non-null  category
 11  VIC_AGE_GROUP           17031 non-null  category
 12  VIC_SEX                 17031 non-null  category
 13  VIC_RACE                17031 non-null  category
 14  X_COORD_CD              17031 non-null  object
 15  Y_COORD_CD              17031 non-null  object
 16  Latitude                17031 non-null  float64
 17  Longitude               17031 non-null  float64
 18  Lon_Lat                 17031 non-null  object
dtypes: bool(1), category(8), float64(3), int64(2), object(5)
memory usage: 1.6+ MB
```

Task 3 -Conversion all the categorical feature names to capitals

All the categorical feature names are converted to capitals to avoid any duplicate names.

Task 4 - Removing unnecessary features

There are some NaN values in the dataset. So, the total null values or missing values in each column are checked. Almost 60 percent of the LOCATION_DESC data is missing, so it does not make sense to replace the missing data with mode of the category. So, this feature is removed from the dataset.

Task 5 - Dealing with missing values

One among many techniques is mode imputation in which the missing values are replaced with the mode value or most frequent value of the entire feature column. When the data is skewed, it is good to consider using mode value for replacing the missing values.Therefore we replace NaN values of other features of categorical data with mode of the category.

Task 6 - Removing 'UNKNOWN' categories from the features

In our dataset we have seen that there is an "UNKNOWN" category in 'PERP_RACE','PERP_SEX','PERP_RACE','VIC_SEX', 'VIC_RACE'. So, we can safely remove these data as it comprises only a small percentage of data .

```
index_names = data[ data['VIC_RACE'] == 'UNKNOWN'].index
data.drop(index_names, axis=0,inplace=True)
data['VIC_RACE'].value_counts()
```

```
BLACK                              11239
WHITE HISPANIC                      2047
BLACK HISPANIC                      1429
WHITE                                412
ASIAN / PACIFIC ISLANDER             199
AMERICAN INDIAN/ALASKAN NATIVE         6
UNKNOWN                                0
Name: VIC_RACE, dtype: int64
```

Age range '1020', '940', '224' also does not make sense in the dataset. So, these are removed from the dataset.

```
index_names = data[ data['PERP_AGE_GROUP'] == 'UNKNOWN'].index
data.drop(index_names, axis=0,inplace=True)

index_names = data[ data['PERP_AGE_GROUP'] == '1020'].index
data.drop(index_names, axis=0,inplace=True)

index_names = data[ data['PERP_AGE_GROUP'] == '940'].index
data.drop(index_names, axis=0,inplace=True)

index_names = data[ data['PERP_AGE_GROUP'] == '224'].index
data.drop(index_names, axis=0,inplace=True)

data['PERP_AGE_GROUP'].value_counts()
```

```
18-24      9892
25-44      2979
<18         870
45-64       284
65+          35
1020          0
224           0
940           0
UNKNOWN       0
Name: PERP_AGE_GROUP, dtype: int64
```

Task 7 - Convert the date and time format

Convert the date and time format to datetime64 datatype.

Task 8 - Splitting date

Split the date to different columns like day, month and year and understand the data better. Day of the month does not make much sense and we cannot draw any insights. So,the day is converted into weekdays of the week.

```
def weekDay(a):
    return a.strftime('%A')

data['WEEKDAY'] = data['OCCUR_DATE'].apply(lambda x : weekDay(x))
```

time is split into different columns like hours, minutes and seconds to better understand the data.

```
# code
data['time_HOUR'] = data['OCCUR_TIME'].dt.hour
data['time_MINUTE'] = data['OCCUR_TIME'].dt.minute
data['time_SECOND'] = data['OCCUR_TIME'].dt.second
```

Task 9 - Dropping extra columns

Now that we have our time and date data in other new columns, we can drop the 'OCCUR_DATE' and 'OCCUR_TIME' .

Task 10 - splitting of age

The age column is described in the form of categories like "18-24", "<18".....etc., so we have splitted these ranges into two separate columns and have found the mean of them.

```
data['PERP_AGE_GROUP'] = data['PERP_AGE_GROUP'].replace(to_replace ="65+", value =
"65-100")
data['PERP_AGE_GROUP'] = data['PERP_AGE_GROUP'].replace(to_replace ="<18", value =
"0-18")
```

```
data[['AGE_GROUP_MIN', 'AGE_GROUP_MAX']] = data['PERP_AGE_GROUP'].str.split('-|<', 1, expand=True)
data["AGE_GROUP_MIN"] = data["AGE_GROUP_MIN"].astype("int")
data["AGE_GROUP_MAX"] = data["AGE_GROUP_MAX"].astype("int")
```

```
data['AGE_MEAN'] = (data['AGE_GROUP_MIN']+data['AGE_GROUP_MAX'])/2
```

Task 11 - Check if the data is skewed

We see that data is skewed with 'PERP_RACE' of 'BLACK', However, we choose not to remove any skewed data as we might lose some insights. Same with the 'PERP_SEX','VIC_RACE','VIC_SEX'.

Task 12 - Encoding the categorical data with numerics

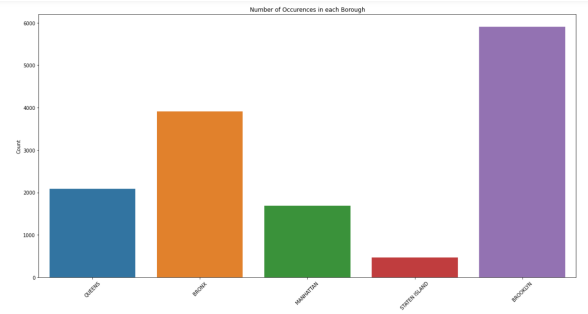
Machine learning models require all input and output variables to be numeric. This means that if our data contains categorical data, we must encode it to numbers before we can fit and evaluate a model. Encoding is a required pre-processing step when working with categorical data for machine learning algorithms.

```
# code
data["PERP_AGE_GROUP"] = data["PERP_AGE_GROUP"].cat.codes
data["PERP_SEX"] = data["PERP_SEX"].cat.codes
data["PERP_RACE"] = data["PERP_RACE"].cat.codes
data["VIC_AGE_GROUP"] = data["VIC_AGE_GROUP"].cat.codes
data["VIC_SEX"] = data["VIC_SEX"].cat.codes
data["VIC_RACE"] = data["VIC_RACE"].cat.codes
data["BORO"] = data["BORO"].cat.codes
data["STATISTICAL_MURDER_FLAG"] = data["STATISTICAL_MURDER_FLAG"].astype("category")
data["STATISTICAL_MURDER_FLAG"] = data["STATISTICAL_MURDER_FLAG"].cat.codes
```

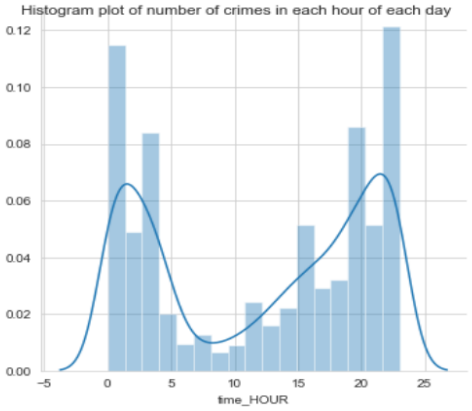
IV. EXPLORATORY DATA ANALYSIS

Exploratory data analysis (EDA) is a term for certain kinds of initial analysis and findings done with data sets, usually early on in an analytical process.functions that are applied on our dataset be performed are — the description of data, handling outliers, getting insights through the plots.

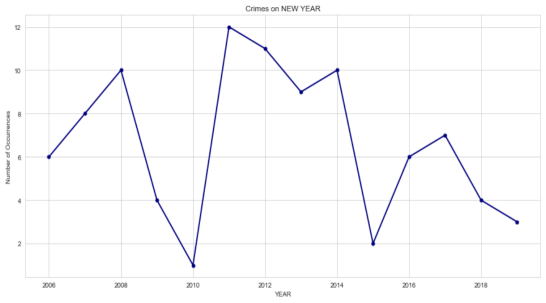
- **Hypothesis 1**: Finding which BORO has the highest number of shootings helps in analysing the safest neighbourhood in the New york city.



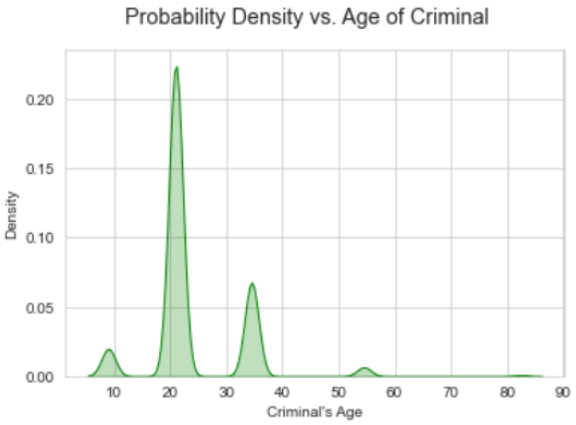
- **Hypothesis 2**: The Borough which has the highest number of shootings may or may not have the highest number of "MALE" victims or "FEMALE" victims or Race of the victim for that sake. So, we have analysed which sex/race has been most affected.
- **Hypothesis 3**: By analysing the peak time of crimes, we can have insights on which hour of the day is not safe to be out in NYC.



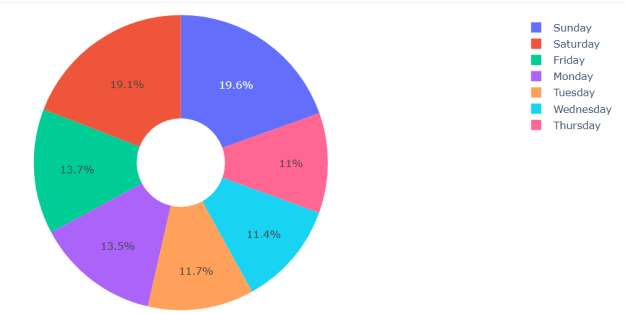
- **Hypothesis 4**: By analysing the trend of crimes over the years from 2006 to 2019, we can have insights on how New york City has become safe over all these years.
- **Hypothesis 5**: Generally crime rates are more on occasions like halloween,new year and july 4. We can analyse if there's any particular trend in shootings.



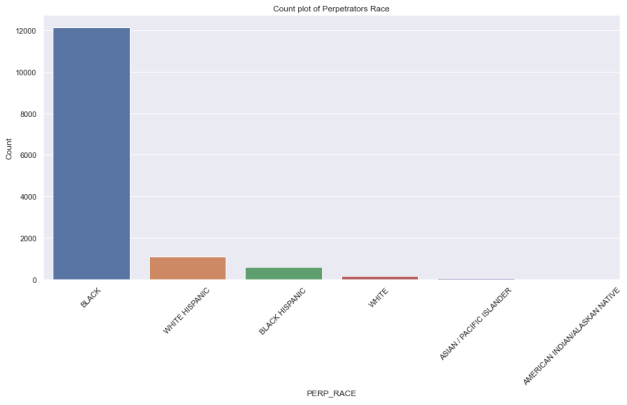
- **Hypothesis 6**: Analysing perpetrator's age we can derive which age group is involved in the shootings.



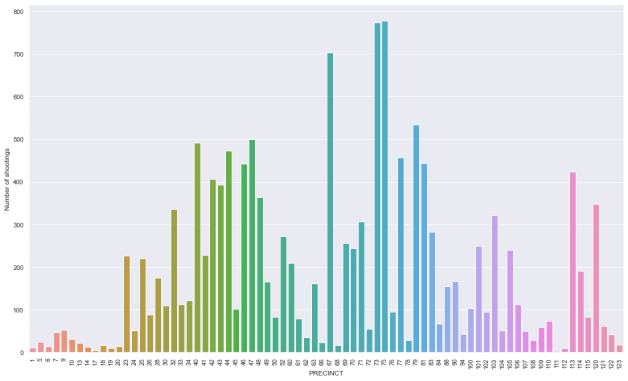
- **Hypothesis 7**: Analysing victim's ages we can derive which age group of victims have got shot more
- **Hypothesis 8:** By analysing which day of a week has recorded more number of shootings we can derive the most unsafe day of a week to travel in NYC.



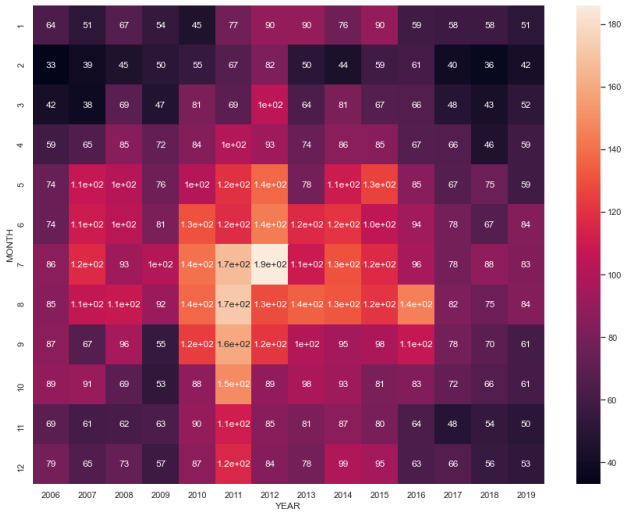
- **Hypothesis 9**: With the insights from previous hypotheses we try to find the dangerous hours of the weekend for these crimes to occur.



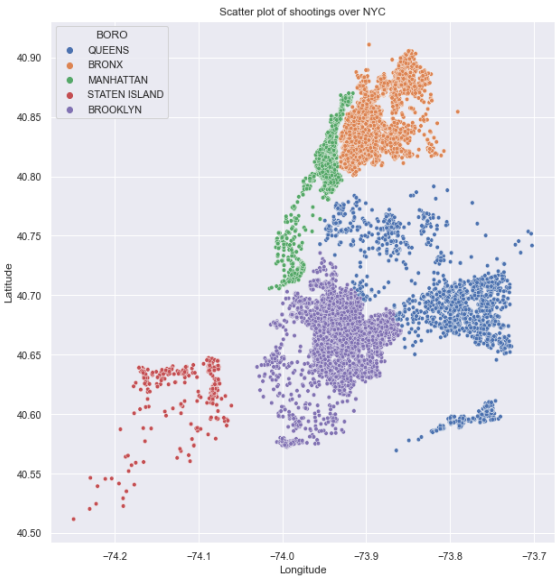
- **Hypothesis 10**: Analysing the number of shootings in precincts to derive the top 5 safe and unsafe precincts in NYC.



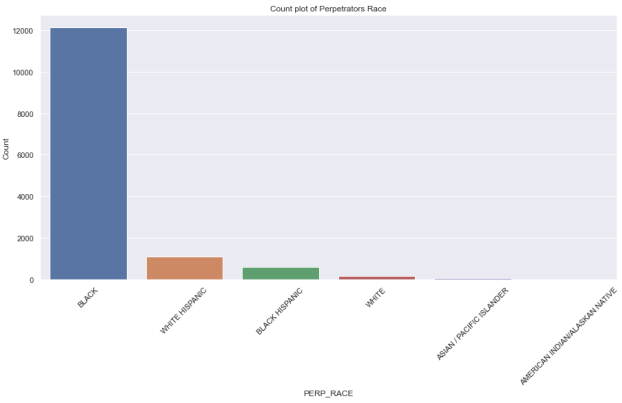
- **Hypothesis 11**:To analyse perpetrator's sex and race to know which sex and race has committed most of the shootings.
- **Hypothesis 12**:By analysing victim's sex and race we can derive which sex and race of victim's has been affected the most.
- **Hypothesis 13**: By analysing the crimes occured over the months of each year we can derive which months has recorded more number of shootings.



- **Hypothesis 14**: Visualising the crimes over different boroughs of NYC to know how scattered the crimes have occured.



- **Hypothesis 15**: Analysing jurisdiction codes to derive which has recorded more shootings like "TRANSIT","PATROL" or "HOUSING".
- **Hypothesis 16**: Analysing the stastical murder flag we can derive if the Shooting resulted in the victim's death which would be counted as a murder.



V. MODELLING

Data modeling is the process of producing a descriptive diagram of relationships between various types of information .It is a way of mapping out and visualizing all the different places that a software or application stores information, and how these sources of data will fit together and flow into one another. We have used 5 different machine learning algorithms .

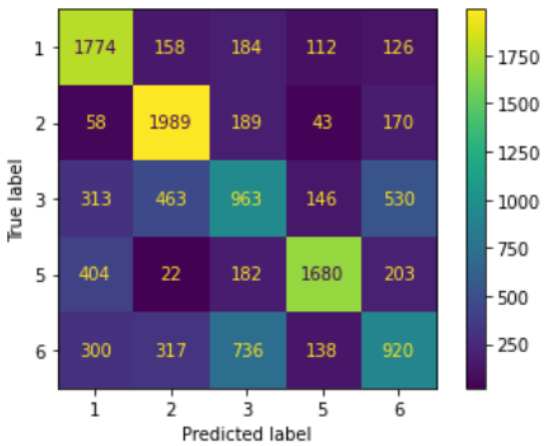
Model -1 LOGISTIC REGRESSION

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Suppose there is crime happened at a particular location at a particular

time and we know all the victim's details like age, sex and race. We want to see what is the most probable race of the perpetrator. As we assumed that observations are independent to each other and also, we have large enough dataset to work with simple classification algorithms like Logistic, we chose this and also, Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval independent variables.We use confusion matrix (precision and recall) and Area under the curve to check the goodness the model.

OUTCOME

We could achieve 62% accuracy and AUC of 0.87 which says our models are good enough with decent precision and recall for each class According to the metric values we got, our model is good enough since we have 5 different classes and although the accuracy is low, we could achieve better area under the curve. As data is skewed, I used SMOTE technique to oversample the data and predict. From the classification report, we can say that both precision and recall are good enough. If you are a police inspector and you want to catch criminals, you want to be sure that the person you predicted the criminal's RACE correct(Precision) and you also want to capture as many criminals (Recall) as possible. The F1 score manages this tradeoff.

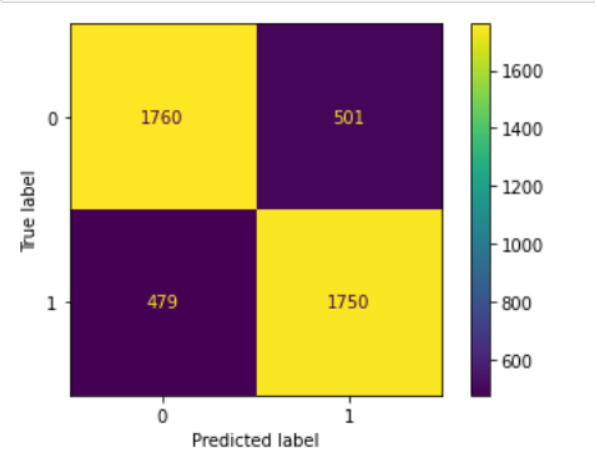


Model -2 RANDOM FOREST

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.In our project,suppose,STATISTICAL_MURDER_FLAG represents the safety of the neighbourhood as it says if the shooting resulted in a murder of the victim, we want to see if we can predict the safety of a location for a given particular time. Random forest Classifiers are robust to outliers and work well for non-linear data. As only a few number of murders were reported, we don't want any overfitting of the data. We know that Random forest classifiers works well for large datasets and low risk of overfitting.¶Since accuracy scores may not give the correct analysis due to our imbalanced dataset, we use AUC curve and Confusion matrix and Precision and recall to predict if a location is safe or not in a particular time.

OUTCOME

We have achieved an accuracy of 78% and Area under the curve as 0.86 which is good enough for the dataset we have.Although 78% is not as good as a model, we can see that Area under the curve is good enough. As the dataset is imbalanced, if we train our model with the original dataset, the accuracy is high but f1_score is zero and the model always tries to classify location as SAFE. So, we have used SMOTE technique to oversample and balance the dataset which could achieve a better f1_score and AUC.



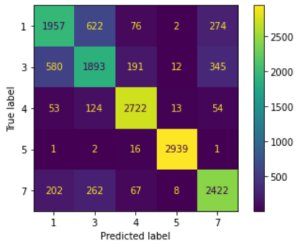
Model -3 DECISION TREE CLASSIFIER

Decision Trees are a type of Supervised Machine Learning that is you explain what the input is and what the corresponding output is in the training data where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. Suppose there is crime happened at a particular location at a particular time and we know all the victim's details like age, sex and race. We want to see what age group does the perpetrator belongs to and therefore to find that we have used a decision tree classifier.Compared to other algorithms, data preparation requires less time for this algorithm. It does not even require data to be scaled. And Decision gives better accuracy than other classification algorithms. Since accuracy score may not give the correct analysis due to our imbalanced dataset, we use AUC curve and Confusion matrix and Precision and recall to predict the age group which criminal belongs to.

OUTCOME

We could predict a 5-multiclass target with 80% accuracy and AUC of 87%.According to the visualized tree, the root node feature is the Perpetrator's race. As data is skewed, we have used SMOTE technique to oversample the data and predict. From the classification report, we can say that both precision and recall are good enough. If you are a police inspector and you want to catch criminals, you want to be sure that the person you predicted the criminal's age is correct(Precision) and you also want to capture as many criminals (Recall) as possible. The F1 score manages this tradeoff.

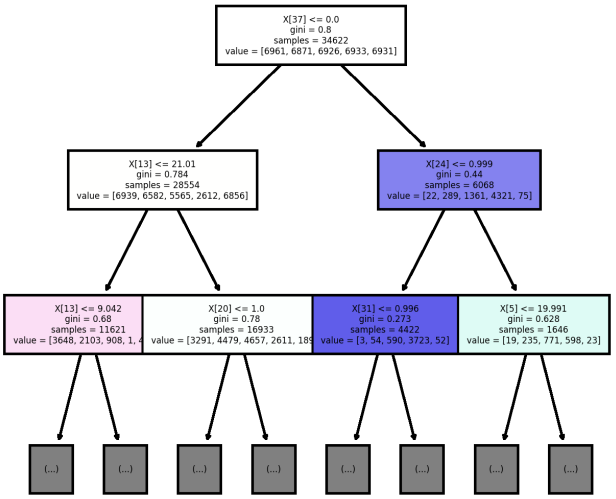
```
plot_confusion_matrix(selector, X_test, y_test)
plt.show()
```




```
# metrics.plot_roc_curve(selector, X_test, y_test)
metrics.roc_auc_score(y_test, selector.predict_proba(X_test), multi_class='ovr')
```

```
0.8778389366373386
```

Visualising the tree

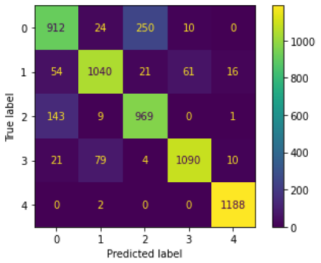


Model -4 K-NEAREST NEIGHBOURS

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.The data is in feature space, which means data in feature space can be measured by distance metrics such as Manhattan, Euclidean, etc. We assume that Each of the training data points consists of a set of vectors and a class label associated with each vector. This is easy to implement and is a non-parametric algorithm.We use confusion matrix (precision and recall) and Area under the curve to check the goodness of this model.

OUTCOME

We could achieve 88% accuracy and AUC of 0.97 which says our models are good enough with decent precision and recall for each class. Based on the crime details like time and criminal details and victim details, we could perfectly classify the location where this type of crime could occur. If you are a police inspector and you want to catch criminals, you want to be sure about the crime location (Precision) and you also want to capture as many crime locations (Recall) as possible. The F1 score manages this tradeoff.




```
metrics.roc_auc_score(y_test, model.predict_proba(X_test), multi_class='ovr')
```

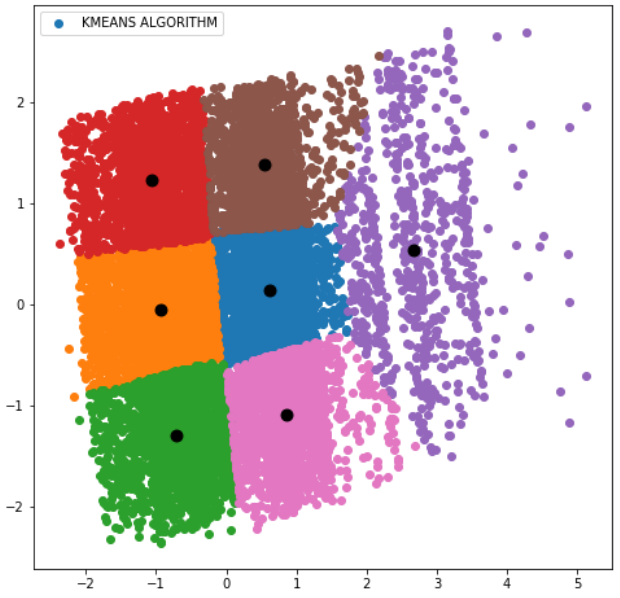
```
0.9777004949435849
```

Model-5 KMeans

Kmeans algorithm is an iterative algorithm that tries to partition the dataset into Kpre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster. Suppose, police want to patrol the cities in most-crime happening places. Let us assume that the police know all the crime locations and the time that the crime is occuring. We want to help them decide the locations and time where more patrolling has to be done.The ability to notice otherwise unseen patterns and to come up with a model to generalize those patterns onto observations is precisely why tools like PCA and k-means are essential in any data science project. After observing the data, we don't know how many prime crime locations there are and what is the prime time, and we don't have labelled ground truth data. So we need a clustering algorithm. K Means clustering is one of the most popular clustering algorithms. Therefore, we choose the k-means algorithm. We did PCA to map my data to 2 dimensions for the k-means algorithm. We use explained variance ratio and silhouette_score to determine the goodness of the clustering.

OUTCOME

The best score we could achieve is with 7 clusters. As we could see the explained variance ratio, almost 60% of the data is given by only 2 components. So, with PCA we could all the data in the features to just two input features and then proceed for modelling. But, since 40% of the data is missing, we could achieve a silhouette_score of 0.38. Anyway, we are able to classify the location and crime data into 7 different clusters and we could see the locations and time of crime where police have to do patrolling from the centroids of each cluster .



VI. SUMMARY

- We can see that 2010-2012 recorded a high number of shooting crimes and is decreasing since then and is least in the recent years (2017,2018,2019).
- Count wise Brooklyn contributed the most amount of shootings.
- July month of 2012 reported the maximum number of shootings.

- Highest shootings occurred on important days like new year in 2011, july 4 th in 2012 and 2016, halloween in 2015.
- Most of the crimes occured from night 9PM to 5AM. So past mid-night is the peak time for shooting crime.
- Peak days for Crimes are Saturday and Sunday. So, it is seen that most of the crimes happen on weekends. Although, we don't see a major difference in the percentage of crimes throughout the weekdays.
- 75th Precinct reported the most amount of shootings.
- We also plotted the maps identifying the areas with maximum shooting activities in each borough and came up with coordinates of the locations, thus identifying safe neighbourhood areas.
- We can observe the similar trend with the number of female and male victims in each borough and similar with the race also. In each borough, most victims are BLACK males.
- People with age group 25-44 are more prone to getting shot.
- Peak days for Crimes are Saturday and Sunday. Now we want to understand which hour of the day is more dangerous. We can see that most of the crimes occured from 12AM to 5AM on weekends.
- Most of the Perpetrators who commited crime are MALE and most of them belong to BLACK race
- most affected sex of victim is MALE.
- High crimes happened from 2010 to 2016 and these happened in the months of June to September.
- Even though a high number of shootings were recorded the number of murders is comparatively less.

VII. CONCLUSION

The analysis performed would produce helpful information such as who were the ones affected in these crimes and which place of New York City is the most dangerous and vulnerable for these crimes. This would help people to be aware of the crimes happening in NYC. Most importantly we can derive which race has been affected the most.

VIII. TOOLS USED

- Jupyter Notebooks for neat and clean data processing in Python.
- Pandas library for Data Analysis and exploration.
- Standard Matplotlib, seaborn, pyplot libraries for visualizations.