Los Angeles Crime Data using Machine Learning Models

Deepak Bhadouria, Ritu Kukreja, Surendra Pothuri

Drexel University - Winter 2023 Philadelphia, Pennsylvania

Abstract—**This project explores the use of machine learning algorithms such as K-Means Clustering and Random Forest, to analyze crime data from the city of Los Angeles. The goal of our project is to gain a better understanding of the patterns in crime across Los Angeles and to build predictive models that can be used to identify what type of crime could be commited based on certain parameters. The data set consists of crime records from the Los Angeles Police Department from past several years. After performing standard pre-processing and exploratory analysis on the data, K-Means Clustering and Random Forest algorithms were applied to the data. The results of this analysis indicated that K-Means Clustering was able to detect clusters of crime that were associated with certain neighborhoods in Los Angeles. Furthermore, the Random Forest model was able to predict the crime type in a given area, based on the attributes of the area.**

I. Introduction

The Crime in Los Angeles dataset contains information on criminal incidents that occurred within the city of Los Angeles from past several years. The data was obtained from the Los Angeles Police Department's Crime Mapping and Analysis Website, and includes details such as the type of crime, the date and time it occurred, the location (by latitude and longitude), and whether an arrest was made. The dataset consists of over 1.6 million records and can be used for various analyses and applications in the field of criminology, urban planning, and public policy.

II. Dataset

The dataset used was found on Kaggle, but orginally the Los Angeles Police Department's Crime. The data contains 1.6 million records. The following variables are found in our dataset:

1)DR_NO (Division of Records Number: Official file number made up of a 2 digit year, area ID, and 5 digits)

2)DATE OCC (Date at which crime occurred (DD/MM/YYYY))

3)TIME OCC (Time in 24H format)

4)AREA (The LAPD has 21 Community Police Stations referred to as Geographic Areas within the department.)

5)AREA NAME (The 21 Geographic Areas or Patrol Divisions are also given a name designation that references a landmark)

6)Rpt Dist (A four-digit code that represents a sub-area within a Geographic Area)

7)Crm Cd (Crime code(for reference refer UCR-COMPSTAT pdf).

8)Crm Cd Desc (Indicates the crime committed.)

9) Mocodes (Modus Operandi: Activities associated with the suspect in commission of the crime.)

10) Vict Age (Age of the victim (0 - Not known))

11) Vict Sex (Sex of victim (M-male, F-female, X-unknown))

12) Vict Descent (Descent Code: A - Other Asian B - Black C - Chinese D - Cambodian F - Filipino G - Guamanian H - Hispanic/Latin/Mexican I - American Indian/Alaska J - Japanese K - Korean L - Laotian O - Other P - Pacific Islander S - Samoan U - Hawaiian V - Vietnamese W - White X - Unknown Z - Asian Indian.)

13)Premis Cd: The type of structure, vehicle, or location where the crime took place.

14) Premis Desc (The type of structure, vehicle, or location where the crime took place.)

15) Weapon Used Cd (Weapon Code(500- Unknown))

16) Weapon Desc. (The type of weapon used in the crime.)

17) Status (Status of the case. (IC is the default))

18) Status Desc (Description of Status)

19)Crm Cd 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.

20)Crm Cd 2: May contain a code for an additional crime, less serious than Crime Code 1.

21)Crm Cd 3: May contain a code for an additional crime, less serious than Crime Code 1.

22)Crm Cd 4: May contain a code for an additional crime, less serious than Crime Code 1.

23) LOCATION (Street address of crime incident rounded to the nearest hundred block to maintain anonymity.)

24) LAT (latitude)

25) LON (longitude)

26)API Field Name: MM/DD/YYYY.

III. Exploratory Data Analysis

Unfortunately, the dataset had not cleaned when compiled. There were missing data.

1) Dropped these columns ("Mocodes","DR_NO","Rpt Dist No", "LOCATION", "Date Rptd", "Part 1-2", "Crm Cd 1", "Crm Cd 2", "Crm Cd 3", "Crm Cd 4", "Cross Street")

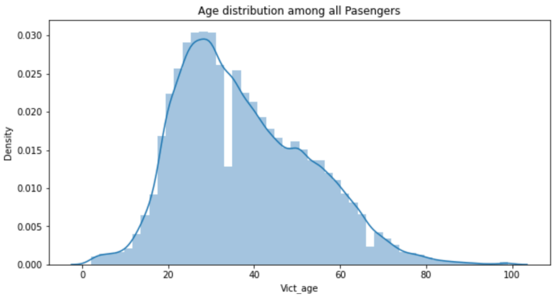
2)Changed some of the columns for easy to understand.

"DATE OCC - Crime_date"
"TIME OCC - Crime_time"
"AREA - Area_cd"
"AREA NAME - Area_name"
"Crm Cd - Crime_cd"
"Crm Cd Desc - Crime_cd_desc"
"Vict Age - Vict_age"
"Vict Sex - Vict_sex"
"Vict Descent - Vict_descent"
"Premis Cd - Premis_cd"
"Premis Desc - Premis_desc"
"Weapon Used Cd - Weapon_cd"
"Weapon Desc - Weapon_desc"
"Status Desc - Status_desc"

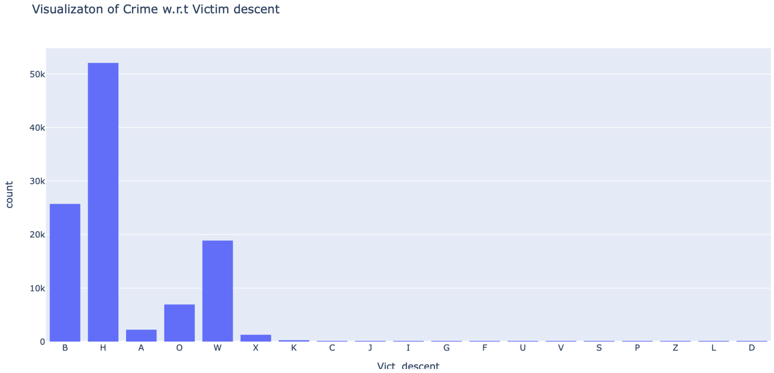
3)Dropped the null value rows using na.drop() function.

4)Filtered the dataset using below conditions.
(df.Vict_age > 0)
(df.Vict_sex != 'null')
(df.Vict_sex != 'H')
(df.Vict_descent != 'null')

Now, the dataset has been cleaned, ready for EDA.



Age Distribution among all passengers.



Visualization of crime w.r.t Victim descent

Crime_cd_desc	count
BATTERY - SIMPLE ASSAULT	25669
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT	18605
INTIMATE PARTNER - SIMPLE ASSAULT	17138
ROBBERY	9244
CRIMINAL THREATS - NO WEAPON DISPLAYED	6564
BRANDISH WEAPON	5049
INTIMATE PARTNER - AGGRAVATED ASSAULT	4672

summary	LAT	LON
count	107448	107448
mean	33.8156735695408	-117.486901329946
stddev	2.88447631304531	10.014661769579
min	0	-118.6652
max	34.3343	0

Vict_sex	count
M	54064
F	52541
X	843

IV. Methodology

A. Feature Engineering and Selection

We used StringIndexer to encode a string column of labels(Victim sex and Victim descent) to a column of label indices, then we used OneHotEncoder, which encodes a column of category indices to a column of binary vectors, with at most a single one-value. It is a way of representing categorical variables as numerical features. The encoder creates a binary column for each category and returns a sparse matrix or dense array.

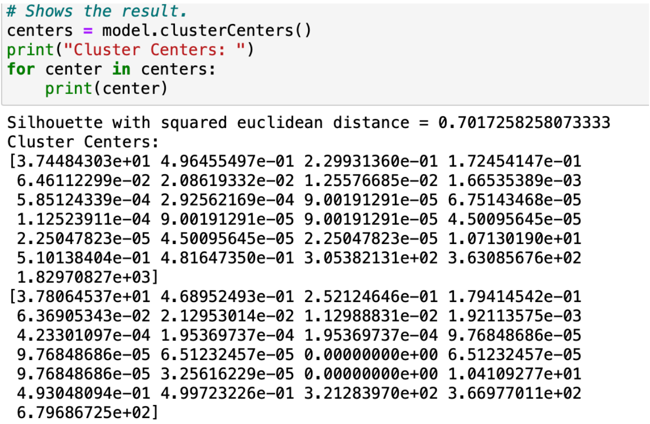
We also used VectorAssembler which is a transformer that combines a given list of columns into a single vector column. It is useful for combining raw features and features generated by different feature transformers into a single feature vector.

B. Machine Learning Modeling

We build two machine learning models in pySpark dataFrame pipeline named :

K-means : We used seven features from our dataset, for which our model will group the data of those features based on the similarity between our features using Euclidean distance between two data points .The smaller the Euclidean distance between two data points, the more similar they are. This measure is then used to assign each data point to its closest cluster and assigns each group a unique label.

EuclideanDistance = sqrt(sum for i to N (v1[i] – v2[i])^2)



```
# Shows the result.
centers = model.clusterCenters()
print("Cluster Centers: ")
for center in centers:
    print(center)
```

```
Silhouette with squared euclidean distance = 0.7017258258073333
Cluster Centers:
[3.74484303e+01 4.96455497e-01 2.29931360e-01 1.72454147e-01
 6.46112299e-02 2.08619332e-02 1.25576685e-02 1.66535389e-03
 5.85124339e-04 2.92562169e-04 9.00191291e-05 6.75143468e-05
 1.12523911e-04 9.00191291e-05 9.00191291e-05 4.50095645e-05
 2.25047823e-05 4.50095645e-05 2.25047823e-05 1.07130190e+01
 5.10138404e-01 4.81647350e-01 3.05382131e+02 3.63085676e+02
 1.82970827e+03]
[3.78064537e+01 4.68952493e-01 2.52124646e-01 1.79414542e-01
 6.36905343e-02 2.12953014e-02 1.12988831e-02 1.92113575e-03
 4.23301097e-04 1.95369737e-04 1.95369737e-04 9.76848686e-05
 9.76848686e-05 6.51232457e-05 0.00000000e+00 6.51232457e-05
 9.76848686e-05 3.25616229e-05 0.00000000e+00 1.04109277e+01
 4.93048094e-01 4.99723226e-01 3.21283970e+02 3.66977011e+02
 6.79686725e+02]
```

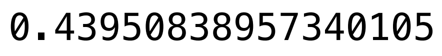
Caption

Random Forest Classifier :

We usedRandomForestClassifier to train our model which uses an algorithm that constructs a large number of decision trees and then combines the results to generate a single output. Each decision tree is built using a random subset(Train data) of the data and a random subset of the features. When we passed our “Test data”to the Random Forest algorithm, each decision tree in the forest makes a prediction and the final output is the average of all the predictions. The final output can be a class label(in the case of classification problems) which is then evaluated by different multi class evaluators.

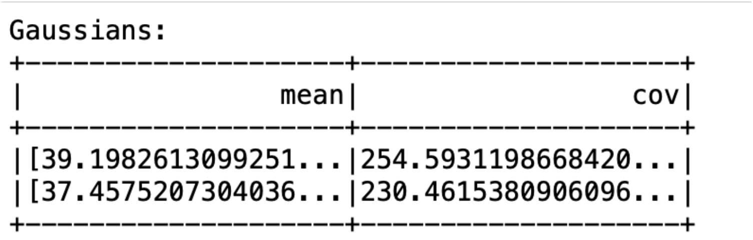
metrics = MulticlassMetrics(prediction_rdd)

metrics.accuracy



```
0.43950838957340105
```

Gaussian Mixture:



```
Gaussians:
+--------------------+--------------------+
|                mean|                 cov|
+--------------------+--------------------+
|[39.1982613099251...|254.5931198668420...|
|[37.4575207304036...|230.4615380906096...|
+--------------------+--------------------+
```

Gaussian Mixture

V. Results

Each model performed fine, with the best performing learners being the K-means, which gave us the Euclidean distance of “0.71” and our multi class evaluators for random forest classifier gave us the accuracy of 50%.

VI. Conclusion and Future Work

The K-means algorithm was out good model producing more similarity between two data points with measuring Euclidean distance. Our random forest model was our average performing model. It was able to achieve Accuracy nearly 50%. For our future work we would be work more on cleaning datasets and more on feature extraction. Our Gaussian mixture works well in measuring distance between two points. This is important for many ML algorithms as it provides a way to compare the similarity of two points in a multi-dimensional space. It also provides an efficient way to calculate distances between points, which can be useful for clustering or other analysis tasks.