“The Guardian Files: Exploring the Washington Post’s Data on Deadly Police Shootings in the United States”

A Punchline Report

Authors:

Suraj Basavaraj Rajolad – 02131154

Deeksha Mallampet – 02120800

Pavan Kumar Gummuluru – 02121263

Vinayak Gururaj Sonter - 02136799

ISSUES:

The data provided contains the information about police shootings across different states in the United States from the year 2015 – 2022.

The data-base includes factors like Id, name, date, manner of death, armed, age, gender, race, city, state, signs of mental illness, threat level, flea, body camera, longitude, latitude and is geocoding exact.

We are going to perform descriptive statistical analysis and perform logistic regression on the basis of gender.

We address the following the questions based on the collected data- base.

- What are the places in Texas where highest number of shootouts have been taken place?
- What is the probability of Black female that has been shot and tasered in the United States?
- Can we prepare a machine learning model which can predict the gender of people that have been shot using other factors provided in the dataset?

FINDINGS:

- After preparing clusters in the Texas state, we found the places in each cluster which are:
  1. Vicinity of Midland, TX (32.8623, -102.5753).
  2. Near the city of San Antonio, TX (29.2542, -98.1550).
  3. Points to location in Dallas, TX (32.7511, -96.6161).
  4. Points to a location in Houston, TX (29.7847, -95.2728).
- The number of female been encountered is less when compared to man. So, the probability of Black Female been ‘shot and tasered’ is approximately 0.0872.
- The short answer for third issue is “Yes”. We worked on logistic regression model, which uses factors like ‘manner_of_death’, ‘age’, ‘race’, ‘state’, ‘sign_of_mental_illness’, ‘threat_level’ and geographical coordinates to predict the gender of a person.

DISCUSSION:

Firstly, by looking at the dataset, we thought of working on gender so we plot bar graph and a pie chart, which shows the number of male and female encountered in the United States of America. We selected Texas state because the population of people who belong to black race is more as compared to the other states in the US. (googled)

Secondly, we prepared clusters using hierarchical clustering technique on the basis of geographical coordinates as we wanted to know the places in Texas where commonly shootouts have been taken place. From each cluster we calculated the mean, longitude and latitude which refers to the place where commonly shootouts take place.

We then implemented the logistic regression to predict the gender using other factors in the data frame and also checked if the model was perfectly working using cross-validation technique.

APPENDIX A METHOD:

We got the data from ‘the Washington post’, this is a website which keep the track of people who are encountered by the police department in the US. This website maintains the record of people being killed by the police in shootouts. The record contains the information such as name, id, manner of death, state, gender, date, armed, flee, age, race, city, sign of mental illness, threat levels, body camera, longitude, latitude and is geocode exact.

We got the excel sheet for 7999 people been shot across the United States.

We started by cleaning the data by removing the null values, correcting the data format. After cleaning the data, we got around 5175 records to perform the analysis.

We started with simple analysis and after group discussion we decided to perform clustering on Texas state to find the places where the shootings are commonly taken place. We implemented hierarchical clustering on the basis of geographical coordinates. We made 4 clusters in Texas. We then performed descriptive analysis on each cluster to identify the mean and median of longitude and latitude. Mean longitude and latitude are references to the place in the map where there are most police shootouts have taken place.

According to the records the number of male were most encountered than the number of female. We also considered race as a variable. We used Bayes theorem to predict the probability of black female been shot in Texas and the probability obtained is 0.0872.

We created a new empty data frame ‘df_3’ where we copied the data frame ‘df’ to df_3 to create a machine learning model which is used to predict the gender by using few factors from the dataset like manner of death, age, race, state, sign of mental illness, threat level, longitude and latitude. We removed the unwanted columns in the df_3 to use only required columns as variables. There is a function called ‘get_dummies’ which is used to handle the categorical data, because the model cannot work on characters or string to perform statistics. After converting the categorical data into 1 or 0 we got 67 columns. We used label encoder which contains the predicted values.

APPENDIX B RESULTS:

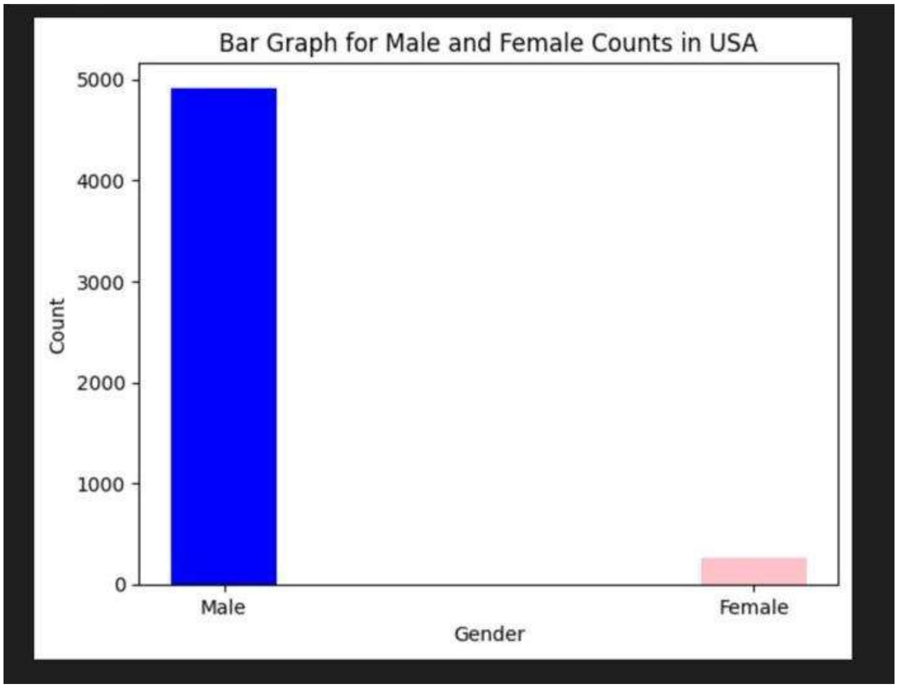


figure:1The figure above shows the count of male and female in united states of America. The number of Male is nearly 4900 and the count of female is less than 1000.

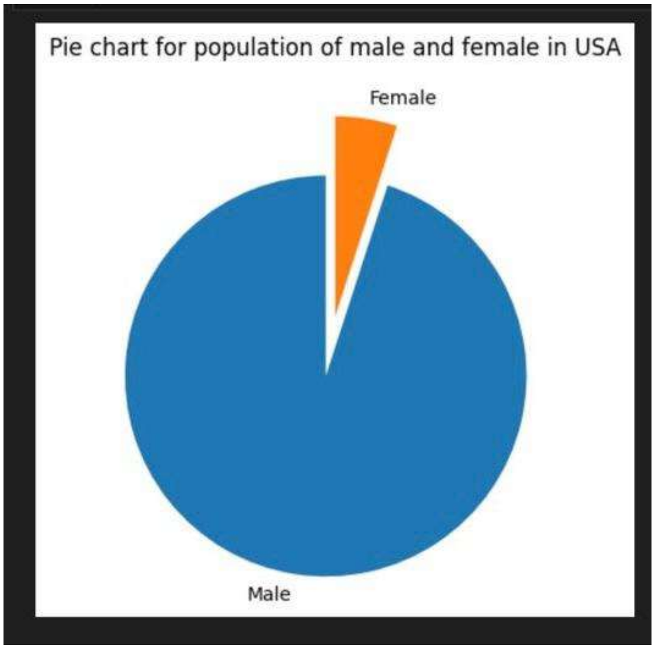


figure: 2 This is a pie chart that is shows the percentage of male and female. For male it is 95.01% and for female it is 4.98

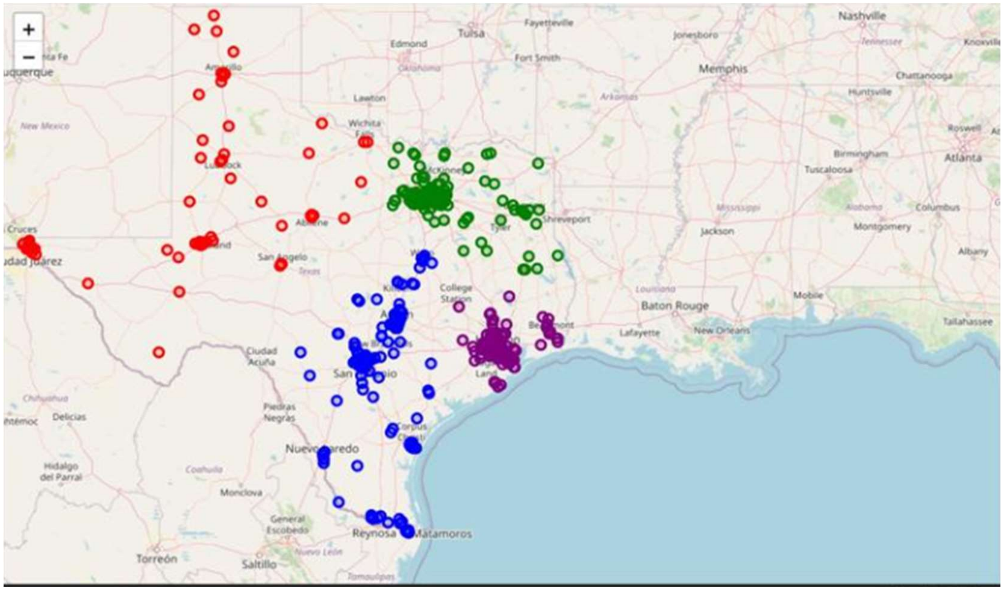


figure: 3 This is the plot of clusters on the map of Texas state, each cluster is given different colours to differentiate between each other. We have used folium library to create the map in the background. Each point in its respective clusters is having some characteristics in common. On the bases of those characteristics the clusters are formed.

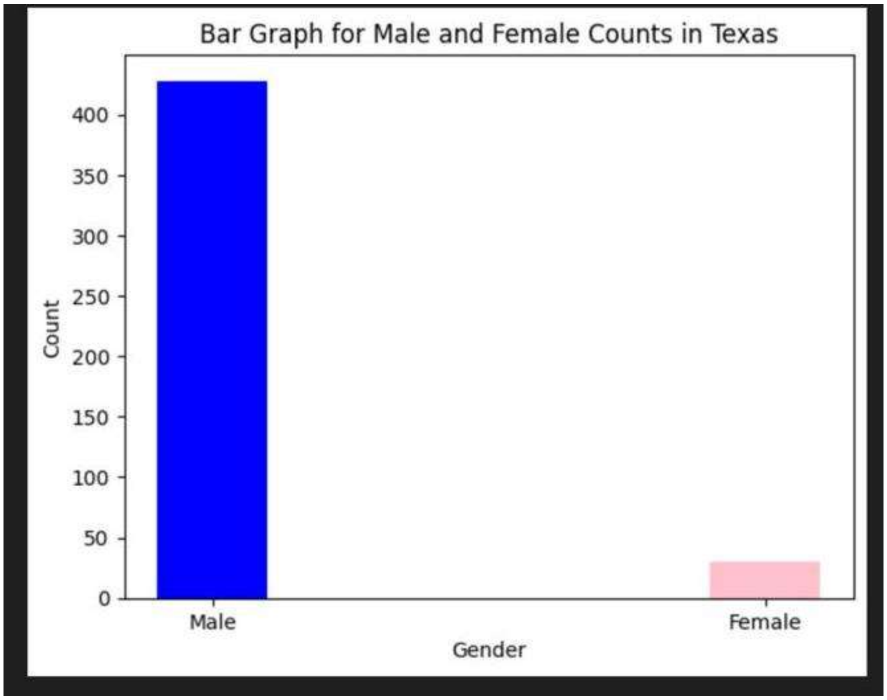


figure: 4 This is a bar graph which shows the count of Male and Female in the records. The bar graph is plotted for Texas state.

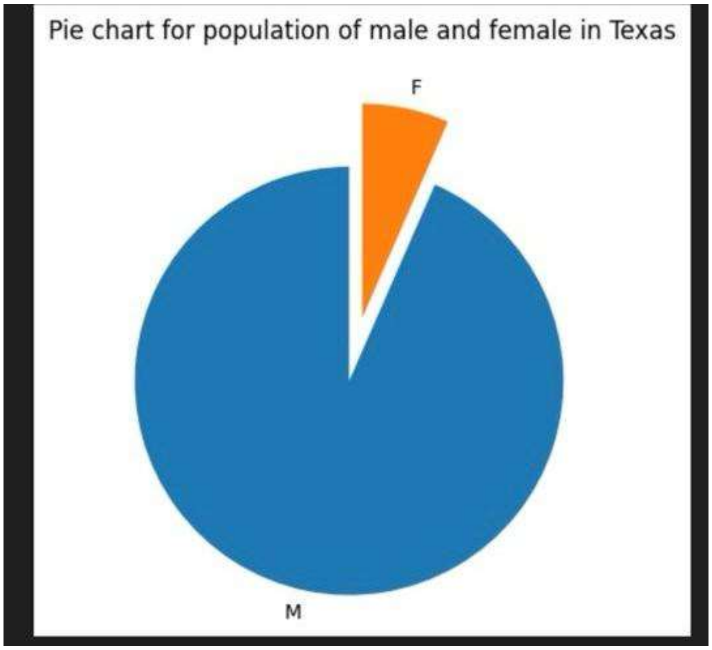


figure: 5 This is a pie chart that is shows the percentage of male and female. For male it is %93.44 and for female it is %6.55

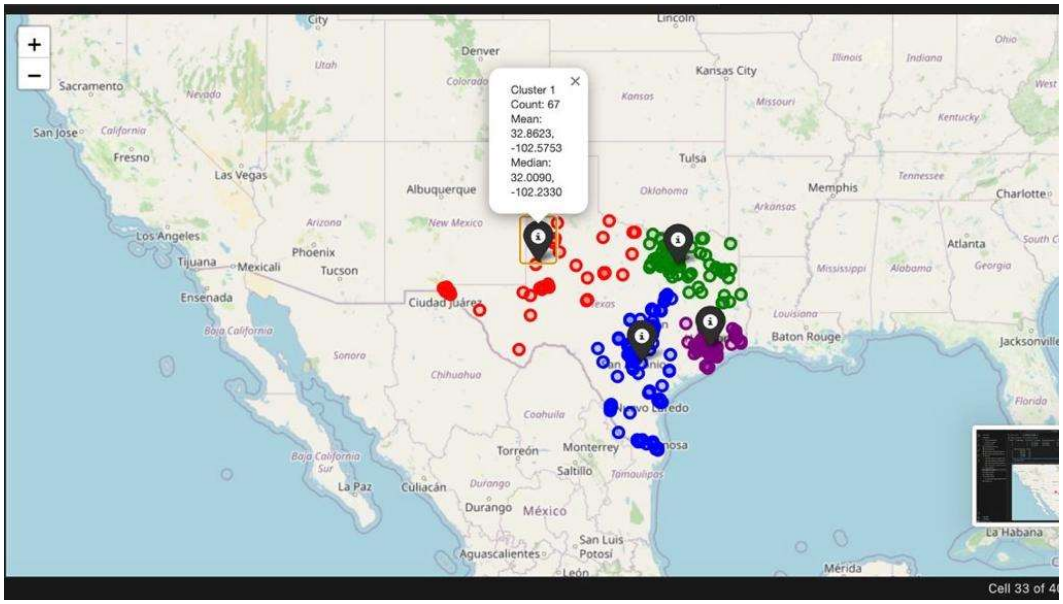
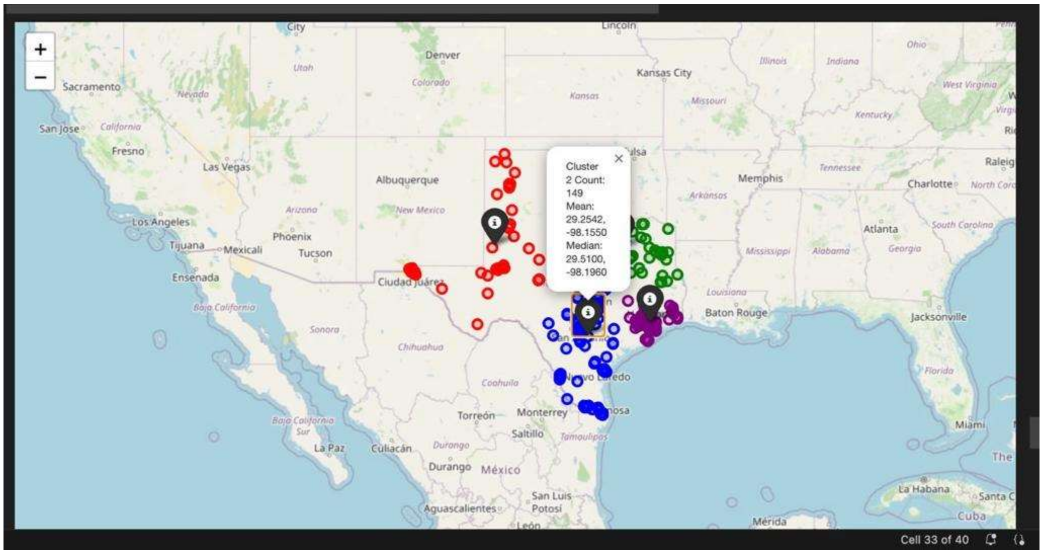
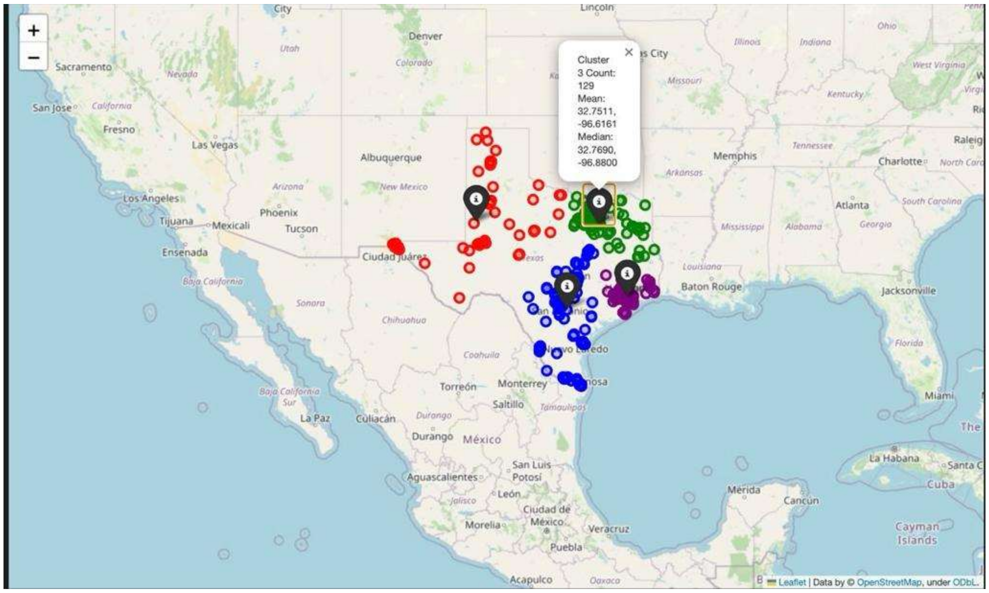

Cluster 1
Count: 67
Mean:
32.8623,
-102.5753
Median:
32.0090,
-102.2330

figure: 6 Vicinity of Midland, TX (32.8623, -102.5753). Even we are counting the number of points in the cluster here there are 67 data points



Cluster 2 Count: 149 Mean: 29.2542, -98.1550 Median: 29.5100, -98.1960

figure:7 Near the city of San Antonio, TX (29.2542, -98.1550). Even we are counting the number of points in the cluster here there are 149 data points



Cluster 3 Count: 129 Mean: 32.7511, -96.6161 Median: 32.7690, -96.8800

figure:8 Points to location in Dallas, TX (32.7511, -96.6161). Even we are counting the number of points in the cluster here there are 129 data points

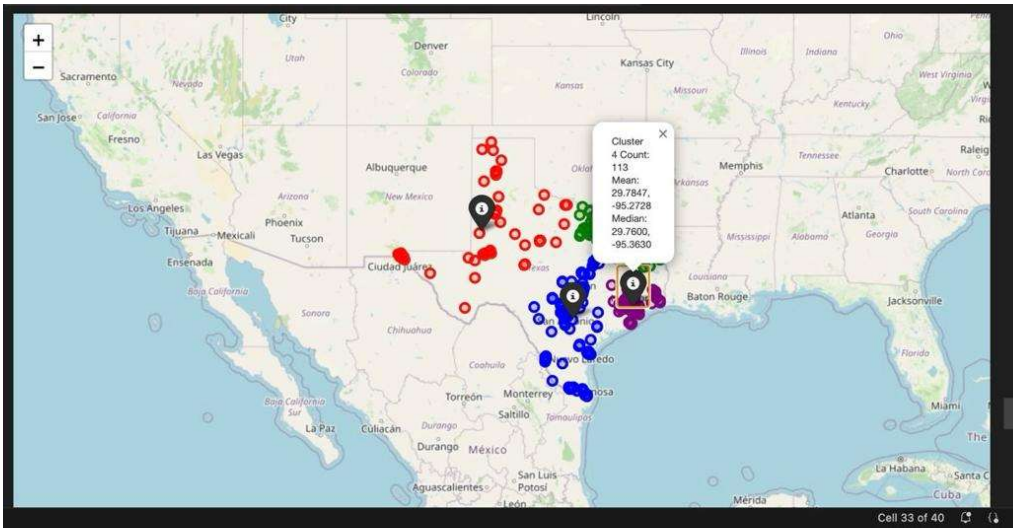
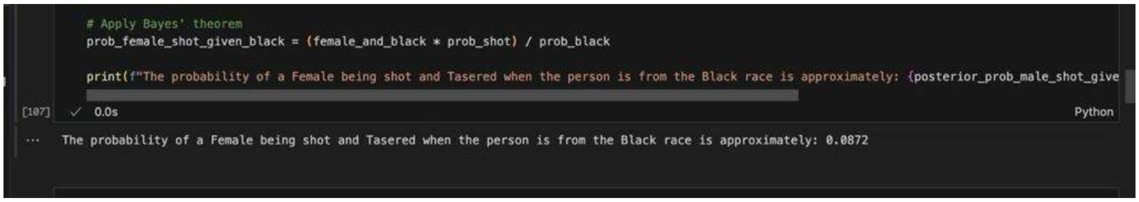


figure:9 Points to a location in Houston, TX (29.7847, -95.2728). Even we are counting the number of points in the cluster here there are 113 data points

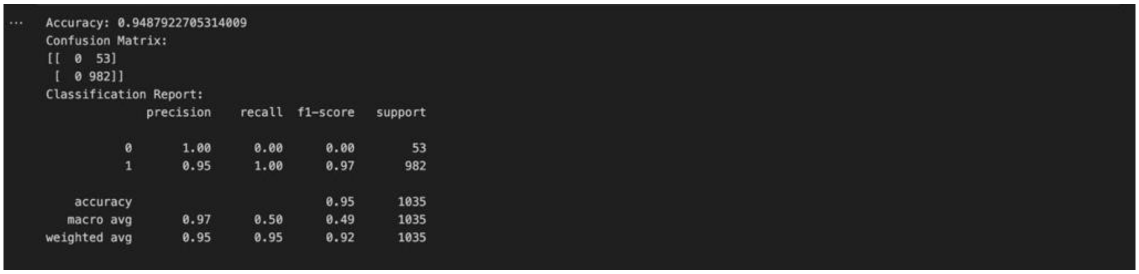


```
# Apply Bayes' theorem
prob_female_shot_given_black = (female_and_black * prob_shot) / prob_black

print(f"The probability of a Female being shot and Tasered when the person is from the Black race is approximately: {posterior_prob_male_shot_give
```

```
[107]  ✓ 0.0s                                                                                     Python
```

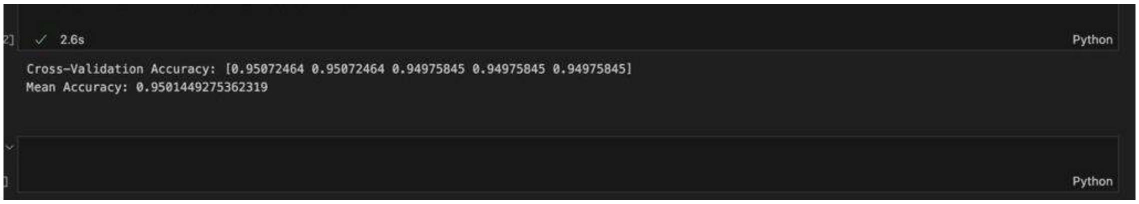
```
··· The probability of a Female being shot and Tasered when the person is from the Black race is approximately: 0.0872
```

figure:10 Probability black female shot



```
···  Accuracy: 0.9487922705314009
     Confusion Matrix:
     [[  0  53]
      [  0 982]]
     Classification Report:
                   precision    recall  f1-score   support

                0       1.00      0.00      0.00        53
                1       0.95      1.00      0.97       982

         accuracy                           0.95      1035
        macro avg       0.97      0.50      0.49      1035
     weighted avg       0.95      0.95      0.92      1035
```

figure:11 Logistic Regression Matrix

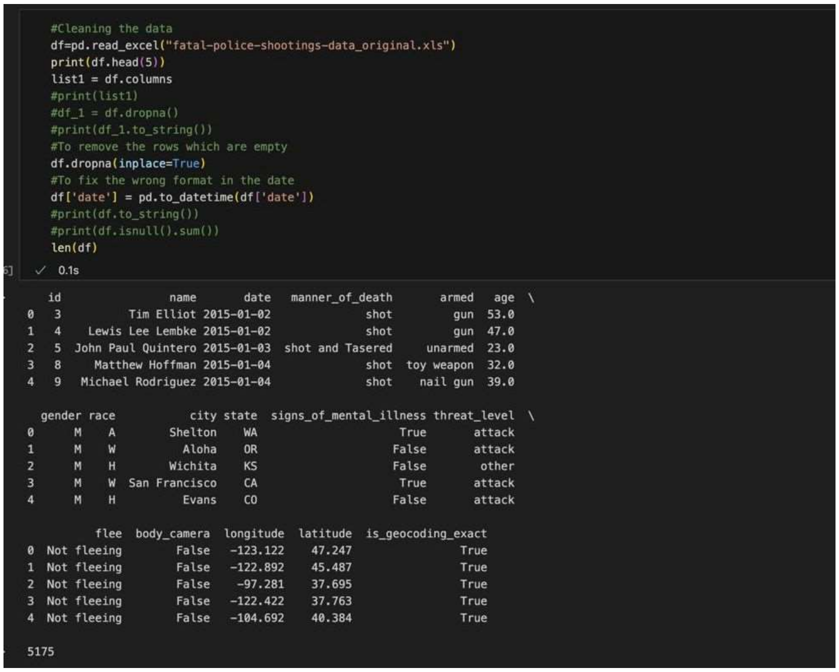


```
Cross-Validation Accuracy: [0.95072464 0.95072464 0.94975845 0.94975845 0.94975845]
Mean Accuracy: 0.9501449275362319
```

figure:12 Cross-validation

APPENDIX C CODE:

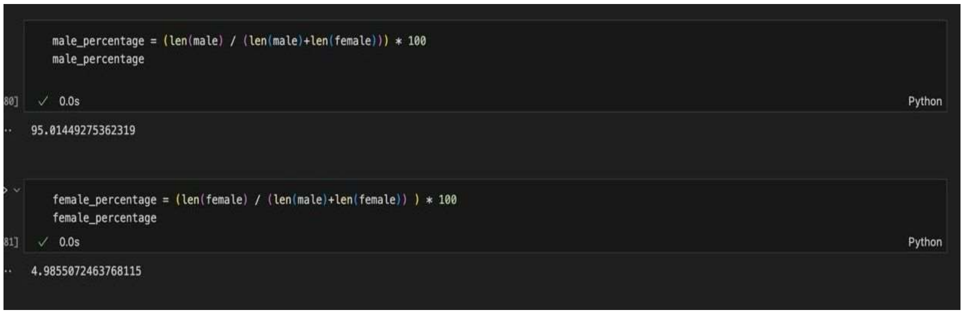
In this code we are cleaning the data for further processing



```
#Cleaning the data
df=pd.read_excel("fatal-police-shootings-data_original.xls")
print(df.head(5))
list1 = df.columns
#print(list1)
#df_1 = df.dropna()
#print(df_1.to_string())
#To remove the rows which are empty
df.dropna(inplace=True)
#To fix the wrong format in the date
df['date'] = pd.to_datetime(df['date'])
#print(df.to_string())
#print(df.isnull().sum())
len(df)
```

```
   id                name       date   manner_of_death       armed   age  \
0   3          Tim Elliot 2015-01-02              shot         gun  53.0
1   4    Lewis Lee Lembke 2015-01-02              shot         gun  47.0
2   5  John Paul Quintero 2015-01-03  shot and Tasered     unarmed  23.0
3   8     Matthew Hoffman 2015-01-04              shot  toy weapon  32.0
4   9   Michael Rodriguez 2015-01-04              shot    nail gun  39.0

  gender race           city state  signs_of_mental_illness threat_level  \
0      M    A        Shelton    WA                     True       attack
1      M    W          Aloha    OR                    False       attack
2      M    H        Wichita    KS                    False        other
3      M    W  San Francisco    CA                     True       attack
4      M    H          Evans    CO                    False       attack

          flee  body_camera  longitude  latitude  is_geocoding_exact
0  Not fleeing        False   -123.122    47.247                True
1  Not fleeing        False   -122.892    45.487                True
2  Not fleeing        False    -97.281    37.695                True
3  Not fleeing        False   -122.422    37.763                True
4  Not fleeing        False   -104.692    40.384                True

5175
```

This is where we calculate the percentage of male and female in USA



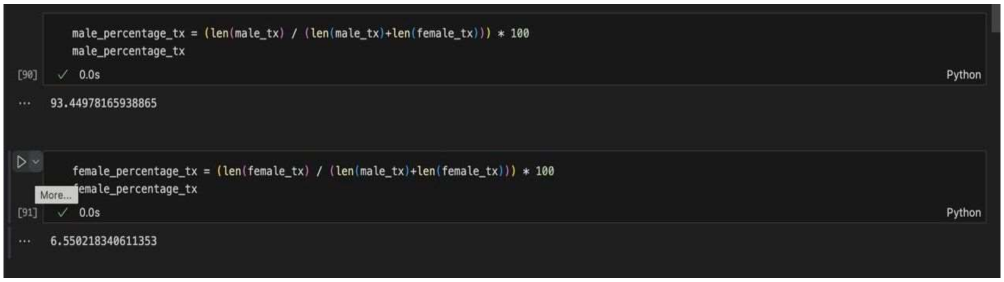
```
male_percentage = (len(male) / (len(male)+len(female))) * 100
male_percentage
```

```
95.01449275362319
```

```
female_percentage = (len(female) / (len(male)+len(female)) ) * 100
female_percentage
```

```
4.9855072463768115
```

Here, we are calculating the percentage of Male and female in Texas state



```
male_percentage_tx = (len(male_tx) / (len(male_tx)+len(female_tx))) * 100
male_percentage_tx
```

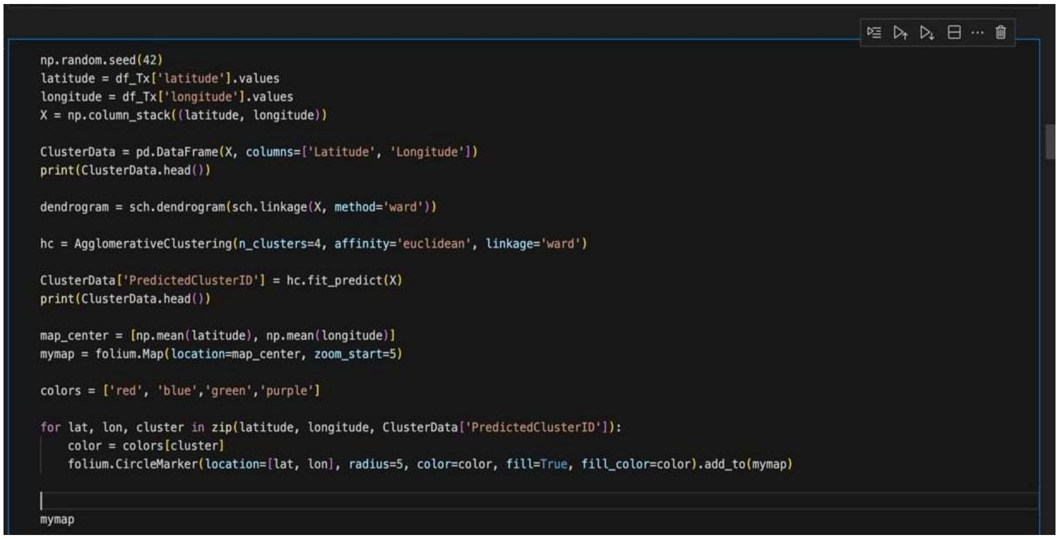
```
93.44978165938865
```

```
female_percentage_tx = (len(female_tx) / (len(male_tx)+len(female_tx))) * 100
female_percentage_tx
```

```
6.550218340611353
```

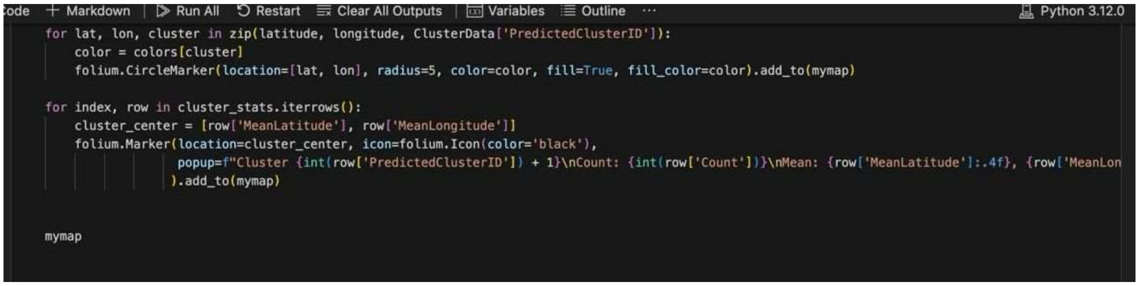
In this part of code, we are preparing clusters we are

Performing descriptive analysis on each cluster



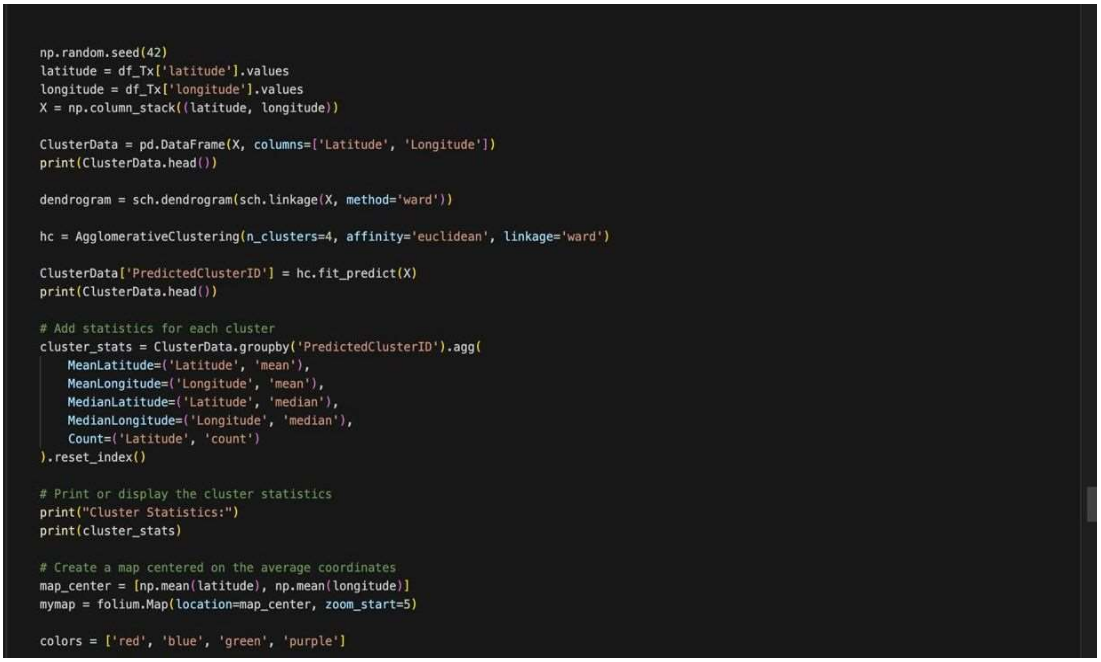
```
np.random.seed(42)
latitude = df_Tx['latitude'].values
longitude = df_Tx['longitude'].values
X = np.column_stack((latitude, longitude))

ClusterData = pd.DataFrame(X, columns=['Latitude', 'Longitude'])
print(ClusterData.head())

dendrogram = sch.dendrogram(sch.linkage(X, method='ward'))

hc = AgglomerativeClustering(n_clusters=4, affinity='euclidean', linkage='ward')

ClusterData['PredictedClusterID'] = hc.fit_predict(X)
print(ClusterData.head())

map_center = [np.mean(latitude), np.mean(longitude)]
mymap = folium.Map(location=map_center, zoom_start=5)

colors = ['red', 'blue','green','purple']

for lat, lon, cluster in zip(latitude, longitude, ClusterData['PredictedClusterID']):
    color = colors[cluster]
    folium.CircleMarker(location=[lat, lon], radius=5, color=color, fill=True, fill_color=color).add_to(mymap)

mymap
```

Here is the code below we are going to plot the map with descriptive analysis

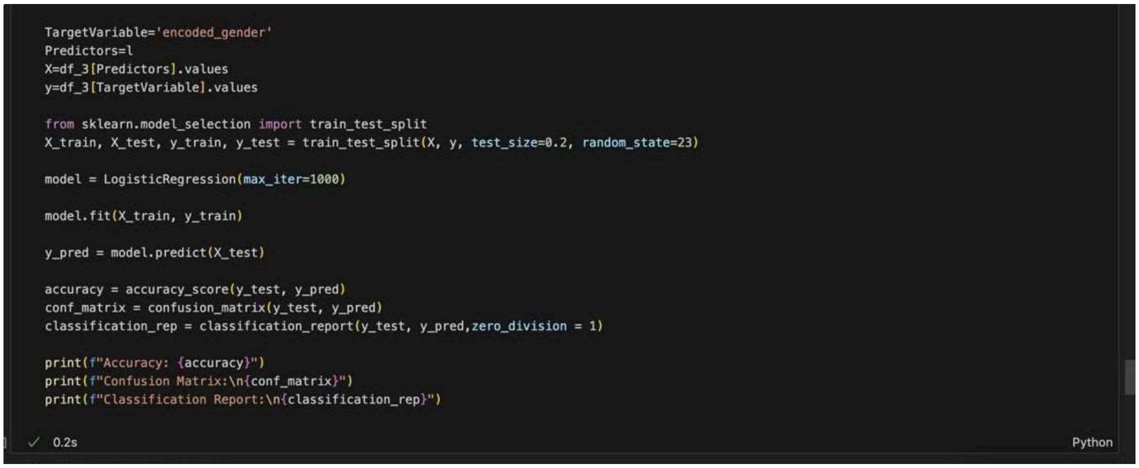


```
for lat, lon, cluster in zip(latitude, longitude, ClusterData['PredictedClusterID']):
    color = colors[cluster]
    folium.CircleMarker(location=[lat, lon], radius=5, color=color, fill=True, fill_color=color).add_to(mymap)

for index, row in cluster_stats.iterrows():
    cluster_center = [row['MeanLatitude'], row['MeanLongitude']]
    folium.Marker(location=cluster_center, icon=folium.Icon(color='black'),
                  popup=f"Cluster {int(row['PredictedClusterID']) + 1}\nCount: {int(row['Count'])}\nMean: {row['MeanLatitude']:.4f}, {row['MeanLon
                 ).add_to(mymap)


mymap
```

This is code for preparing logistic regression model where we got 94% accuracy

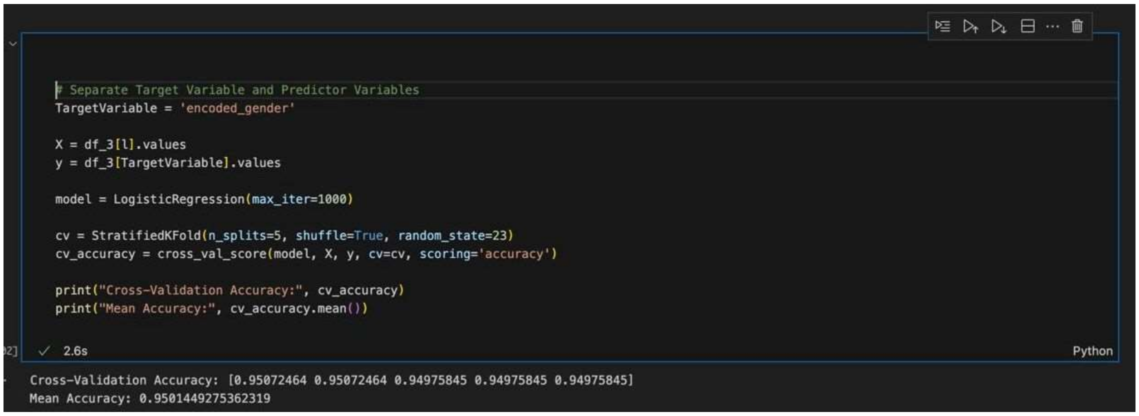


```
np.random.seed(42)
latitude = df_Tx['latitude'].values
longitude = df_Tx['longitude'].values
X = np.column_stack((latitude, longitude))

ClusterData = pd.DataFrame(X, columns=['Latitude', 'Longitude'])
print(ClusterData.head())

dendrogram = sch.dendrogram(sch.linkage(X, method='ward'))

hc = AgglomerativeClustering(n_clusters=4, affinity='euclidean', linkage='ward')

ClusterData['PredictedClusterID'] = hc.fit_predict(X)
print(ClusterData.head())

# Add statistics for each cluster
cluster_stats = ClusterData.groupby('PredictedClusterID').agg(
    MeanLatitude=('Latitude', 'mean'),
    MeanLongitude=('Longitude', 'mean'),
    MedianLatitude=('Latitude', 'median'),
    MedianLongitude=('Longitude', 'median'),
    Count=('Latitude', 'count')
).reset_index()

# Print or display the cluster statistics
print("Cluster Statistics:")
print(cluster_stats)

# Create a map centered on the average coordinates
map_center = [np.mean(latitude), np.mean(longitude)]
mymap = folium.Map(location=map_center, zoom_start=5)

colors = ['red', 'blue', 'green', 'purple']
```



```python
TargetVariable='encoded_gender'
Predictors=l
X=df_3[Predictors].values
y=df_3[TargetVariable].values

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=23)

model = LogisticRegression(max_iter=1000)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred,zero_division = 1)

print(f"Accuracy: {accuracy}")
print(f"Confusion Matrix:\n{conf_matrix}")
print(f"Classification Report:\n{classification_rep}")
```

This is the code to check the accuracy using cross validation.



```python
# Separate Target Variable and Predictor Variables
TargetVariable = 'encoded_gender'

X = df_3[l].values
y = df_3[TargetVariable].values

model = LogisticRegression(max_iter=1000)

cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=23)
cv_accuracy = cross_val_score(model, X, y, cv=cv, scoring='accuracy')

print("Cross-Validation Accuracy:", cv_accuracy)
print("Mean Accuracy:", cv_accuracy.mean())
```

```
Cross-Validation Accuracy: [0.95072464 0.95072464 0.94975845 0.94975845 0.94975845]
Mean Accuracy: 0.9501449275362319
```

Contributions:

All the group members equally contributed to the project