"From Data to Insights: A Data-Driven Exploration of Fatal Police Shootings in the U.S."

Co-authors

Amith Ramaswamy, Chiruvanur Ramesh Babu Sai Ruchitha Babu, Nikhil Premachandra rao, Prajwal Sreeram Vasanth Kumar

A Data-Driven Exploration of Fatal Police Shootings in the U.S.

THE ISSUES:

Within the field of data science and advanced statistics, our attention is drawn to the major issue of fatal police shootings in the United States. By utilizing the information obtained from the Washington Post, our research is ready to discover knowledge. Our goal is to provide detailed, data-driven knowledge of this crucial problem in society that goes beyond traditional news headlines by analyzing complex trends and identifying significant causes.

The dataset in question contains a wide range of features, including important attributes like gender, armed, race, state, body camera, and signs of mental illness.

We address the questions:

- There are several missing values in the data; how to deal with the missing values
- What steps are we taking to address the race column's missing values?
- How should we approach the race column's issue in the future (if new data comes in)?
- What are the trends in police shootings within California?
- How can we understand regional trends contributing to the target?
- The probability of a Male being shot in different races.
- Was there any bias when there was a body cam on any race?

THE FINDINGS:

After analyzing the data collected from Washington Post on fatal police shootings in the U.S.,

- In addressing missing details about individuals, we used an approach called KNN Imputation to estimate missing ages. where we looked at similar cases to estimate and fill in the missing pieces by figuring out what fits best!
- In understanding what factors affect predictions, we simplified ages into groups and used statistical tests to figure out which fits best in predicting someone's race. It's like finding common trends without getting lost in complicated details.
- To determine someone's race, we used an approach that operates like a smart decision-making tree, leading us through data.
- We examined trends in various groups across California, which helped us to grasp patterns and trends. It's similar to detecting trends in different communities in order to understand the bigger picture.

DISCUSSION:

Initially, we took deep insight into the data and found that there were a lot of missing values, we cannot delete all the rows that have empty cells, since data is precious.

In predicting the race of a particular person, there were so many columns, which were in the dataset. so we used some statistical techniques, such as the chi-square test and ANOVA test to determine what and all columns actually affect in predicting the race.

We selected cluster analysis because of California's large population, which is distinguished by its demographic variety and race. This strategy allows for a more detailed investigation of fatal police shootings, revealing patterns and insights among various clusters. The resulting clusters take on considerable significance, providing insights. When examining our decision tree models, it becomes evident that there are three columns that actually contribute to predicting the race of a person: they are latitude, longitude, and age accounting for about 90%.

Application of Bayes theorem to determine the probability of a Male being shot when the person, across all races.

Author Contributions:

	Amith Ramaswamy	Sai Ruchitha Babu	Nikhil Premachandraro	Prajwal S V
Data cleaning	20%	35%	15%	30%
Analyzing	30%	15%	20%	35%
Coding	35%	15%	35%	15%
Visualizing	25%	15%	35%	25%
Report Writing	20%	30%	25%	25%

Appendix A: Method

Data collection:

We accessed the entire police shootings database provided by The Washington Post, available at the following link. This database is useful since it carefully documents fatal police shootings in the United States. The Washington Post carefully selects and updates this material, which comes from a variety of sources, including local news reports, police enforcement websites, and independent databases. Our study acquires a strong foundation of data from which to undertake major analysis and gain insights into the complex dynamics surrounding fatal police shootings by employing this dependable and often updated database.

Variable creation:

Several important variables derived from The Washington Post's police shootings database were examined in our investigation. These variables include a wide range of parameters, and demographic information which includes the age, race, and gender of those involved in fatal police shootings. Geographic information such as location and state were also taken into account in order to comprehend spatial patterns. The use of body cams during events was investigated as an additional factor impacting outcomes. The fact that the shooters were armed offered information on the circumstances underlying the shootings. By carefully integrating these elements, we find trends and understand the complex dynamics present in fatal police shootings in the United States.

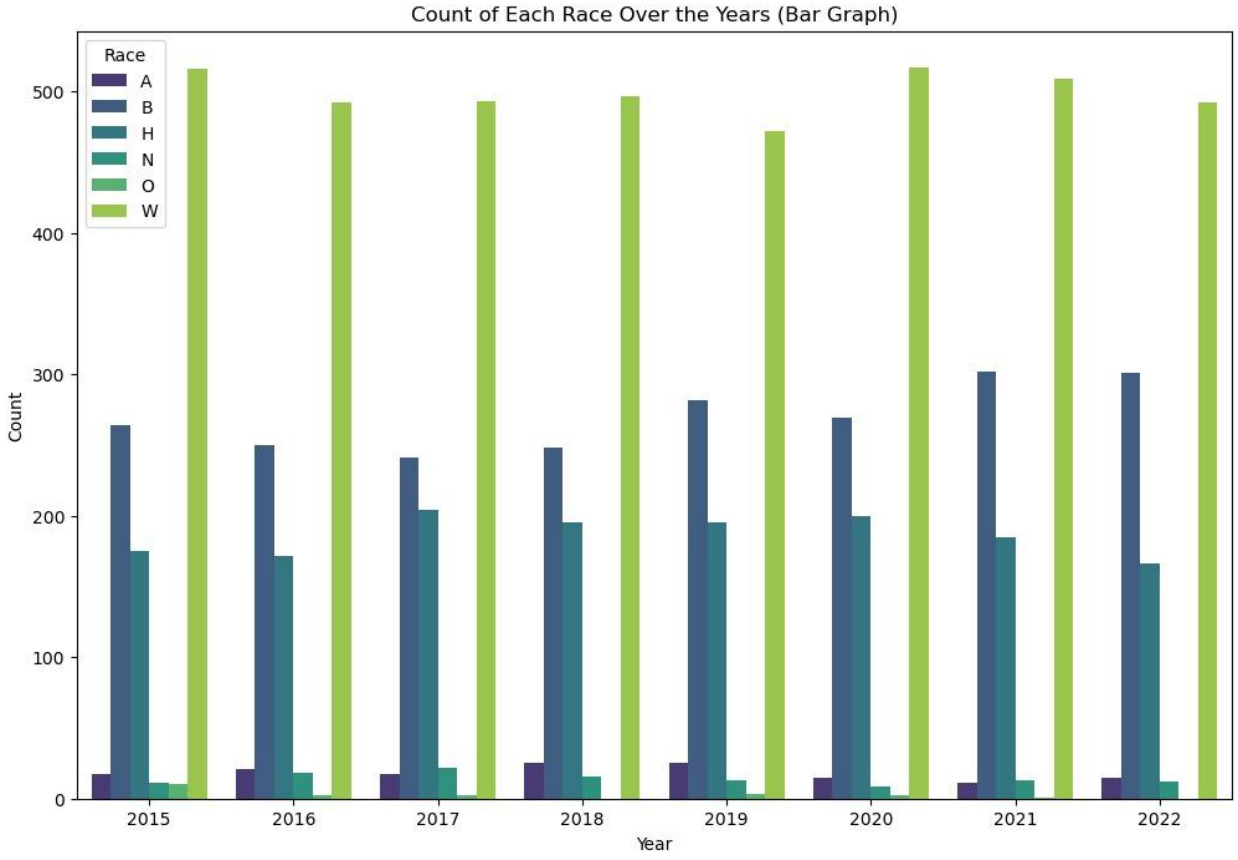
Data Cleaning:

- Missing values were handled using different statistical techniques.
- The age column was handled using the KNN imputation method.
- Analysis by city and state was used to replace the missing data in the race column.
- The missing values in the latitude and longitude columns were successfully handled by grouping them by city and state. However, after doing this, some rows were removed since we were unable to replace the columns in an efficient manner.

Analytic methods:

- Mathematical statistics: Usage of techniques such as KNN Imputation and tests such as (Chi-square, and ANOVA) to determine the columns affecting the creation decision tree.
- Visualizing:
  - Use of a bar graph on race. determining the number of deaths over a period of time on different races.
  - A bar graph to show what and all factors affect the predicting of race.
  - Plotting graph for our decision tree model.
- Bayes rule: The posterior probability of a Male being shot across different races.

Appendix B: Results



Visualizing the trend in the number of deaths of people over each race over the period from 2015 to 2022 using a bar graph: *from the above graph we can conclude that White and Black are the two races which are high in number.*

```python
# Run the anova function for all categorical columns
# Defining a function to find the statistical relationship with all the categorical variables
def FunctionAnova(inpData, TargetVariable, CategoricalPredictorList):
    from scipy.stats import f_oneway

    # Creating an empty list of final selected predictors
    SelectedPredictors=[]

    print('##### ANOVA Results ##### \n')
    for predictor in CategoricalPredictorList:
        CategoryGroupLists=inpData.groupby(predictor)[TargetVariable].apply(list)
        AnovaResults = f_oneway(*CategoryGroupLists)

        # If the ANOVA P-Value is <0.05, that means we reject H0
        if (AnovaResults[1] < 0.05):
            print(predictor, 'is correlated with', TargetVariable, '| P-Value:', AnovaResults[1])
            SelectedPredictors.append(predictor)
        else:
            print(predictor, 'is NOT correlated with', TargetVariable, '| P-Value:', AnovaResults[1])

    return(SelectedPredictors)
```

✓ 0.0s Python

```python
# Calling the function to check which categorical variables are correlated with target
CategoricalPredictorList=['longitude', 'latitude', 'age']
FunctionAnova(inpData=df_anova, TargetVariable='encoded_Race', CategoricalPredictorList=CategoricalPredictorList)
```

✓ 0.1s Python

```
##### ANOVA Results #####

longitude is correlated with encoded_Race | P-Value: 1.3000106498303888e-22
latitude is correlated with encoded_Race | P-Value: 2.1464370891759254e-20
age is correlated with encoded_Race | P-Value: 2.7161890567760563e-60

['longitude', 'latitude', 'age']
```

ANOVA test: *we can see that latitude, longitude, and age are significant to the target column race.*

```python
def chisq(data,predictors,target):
    from scipy.stats import chi2_contingency
    for p in predictors:
        CrossTabResult=pd.crosstab(index=data[p], columns=data[target])
        ChiSqResult = chi2_contingency(CrossTabResult)
        print('The P-Value of the ChiSq Test with '+p+' is:', ChiSqResult[1])


chisq(df,['id', 'manner_of_death', 'armed', 'gender', 'city', 'state', 'signs_of_mental_illness', 'threat_level',
       'flee', 'body_camera', 'is_geocoding_exact',
       'age_group'],'encoded_Race')
```

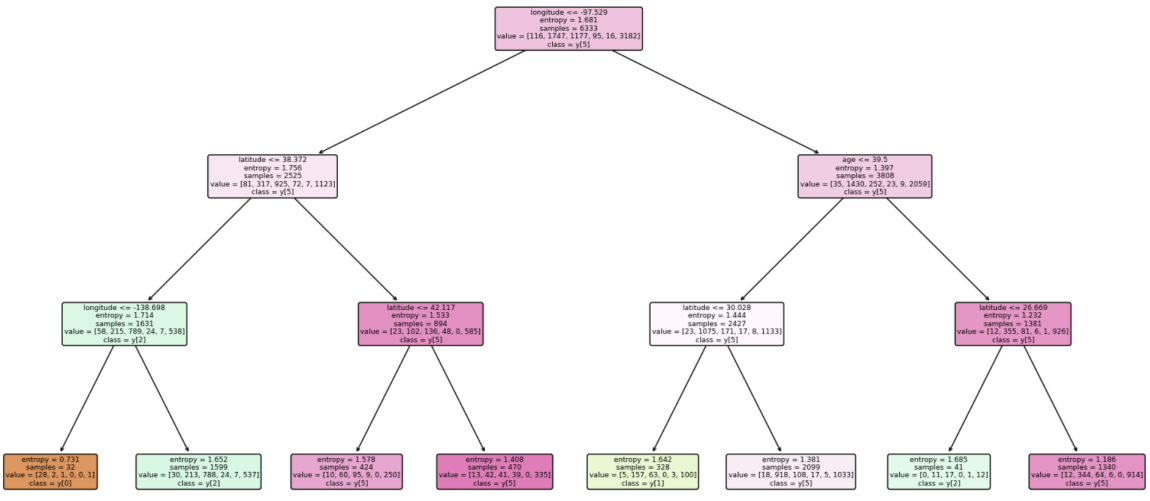
✓ 0.0s Python

```
The P-Value of the ChiSq Test with id is: 0.49196586971810574
The P-Value of the ChiSq Test with manner_of_death is: 0.14469529548927104
The P-Value of the ChiSq Test with armed is: 8.494907970430935e-08
The P-Value of the ChiSq Test with gender is: 0.0005358686513742049
The P-Value of the ChiSq Test with city is: 6.4249835449029695e-127
The P-Value of the ChiSq Test with state is: 0.0
The P-Value of the ChiSq Test with signs_of_mental_illness is: 1.1885669159426051e-26
The P-Value of the ChiSq Test with threat_level is: 8.785046414960335e-08
The P-Value of the ChiSq Test with flee is: 4.100252558936795e-16
The P-Value of the ChiSq Test with body_camera is: 9.300936228851596e-19
The P-Value of the ChiSq Test with is_geocoding_exact is: 0.7293726866201282
The P-Value of the ChiSq Test with age_group is: 8.727229255816087e-15
```

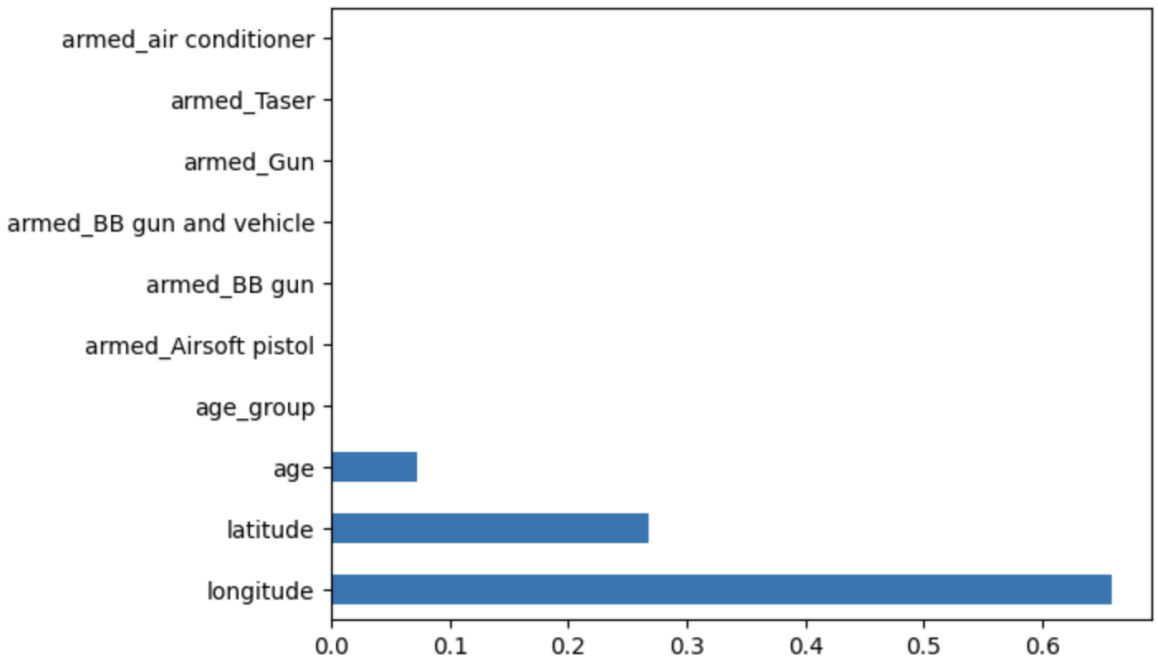
Chi-Square test: *We can consider columns which has a p-value less than significant value of 0.05 on the target value which is "race"*

```
DecisionTreeClassifier(criterion='entropy', max_depth=3)
              precision    recall  f1-score   support

           0       0.50      0.07      0.12        30
           1       0.43      0.07      0.12       410
           2       0.51      0.70      0.59       315
           3       0.00      0.00      0.00        19
           4       0.00      0.00      0.00         4
           5       0.58      0.78      0.67       806

    accuracy                           0.56      1584
   macro avg       0.34      0.27      0.25      1584
weighted avg       0.52      0.56      0.49      1584

[[  2   0  12   0   0  16]
 [  0  29  49   0   0 332]
 [  1  10 220   0   0  84]
 [  0   0   4   0   0  15]
 [  0   0   2   0   0   2]
 [  1  28 148   0   0 629]]
```

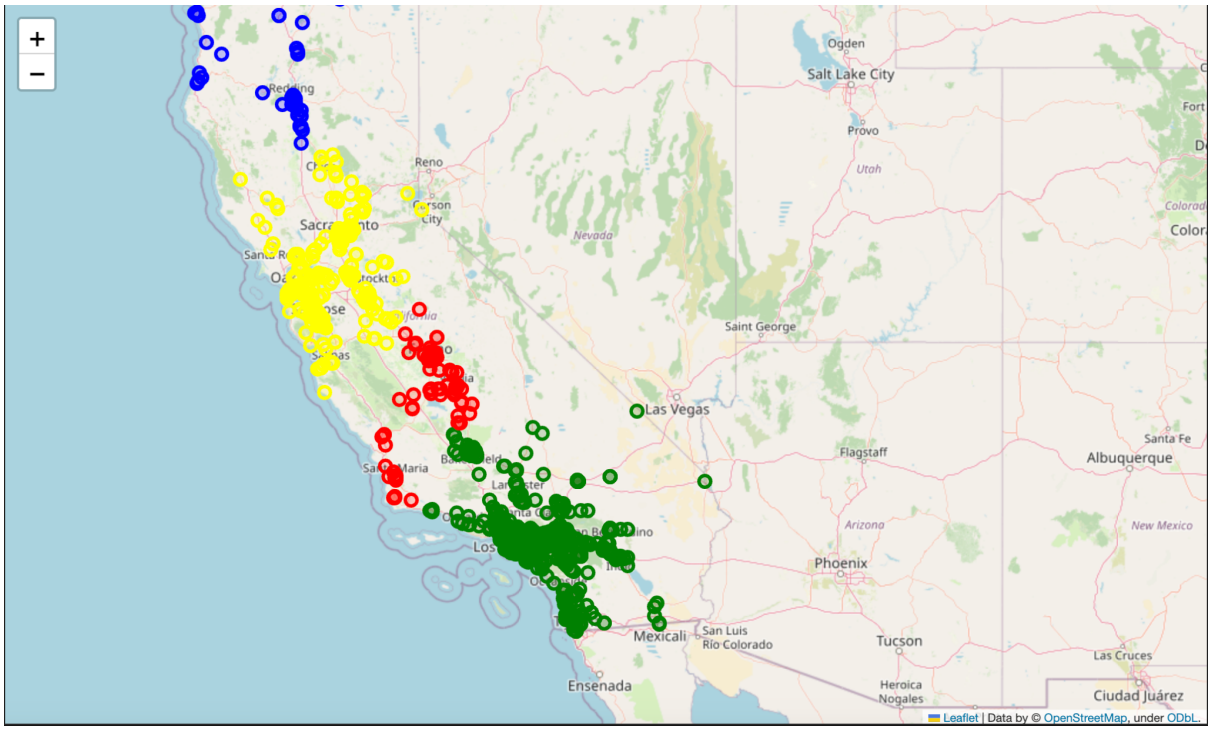
Decision Tree Classifier: *produced an overall accuracy of 56% on the dataset.*



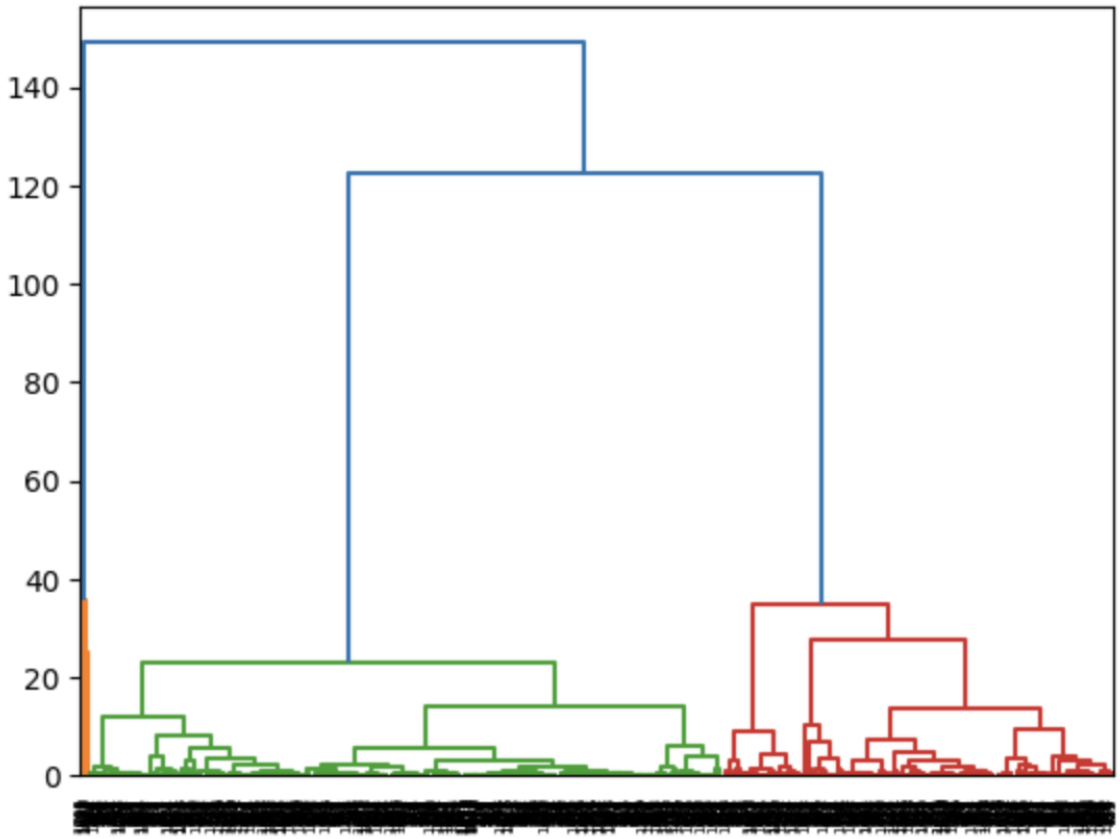
Visualizing a decision tree in the graph: *which breaks down complex choices into clear, understandable branches and outcomes*



Top 10 factors: *that affect predicting the race of a particular person visualized in a bar graph, "latitude, longitude, age" has the lion's share.*



Visualizing clusters: *in the state of California based on latitude and longitude, It is evident from the picture that we see distinct clusters.*

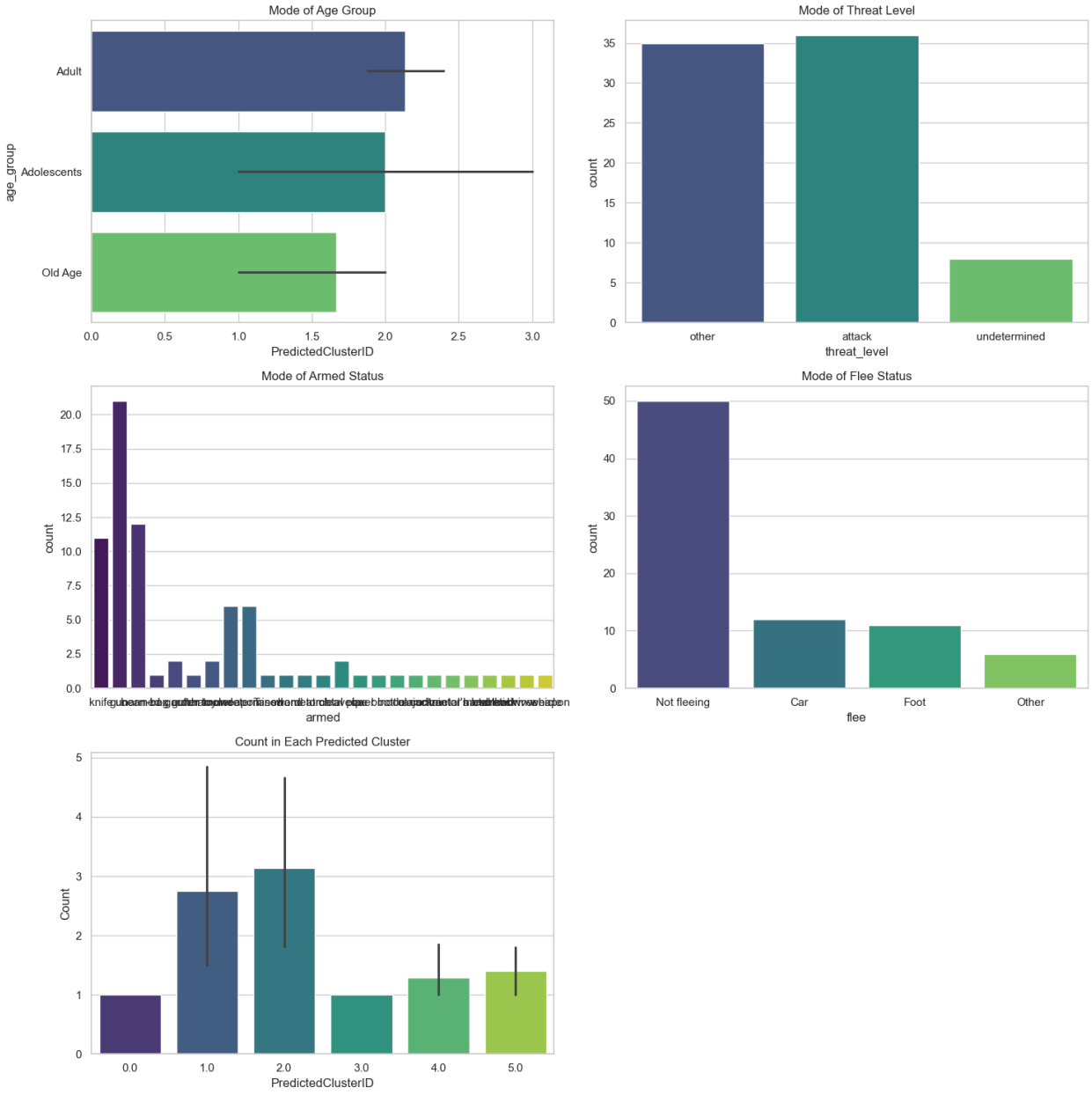


The picture depicts the dendrogram of the hierarchical clustering

```
[173]  df
       ✓ 0.0s                                                                    Python
```

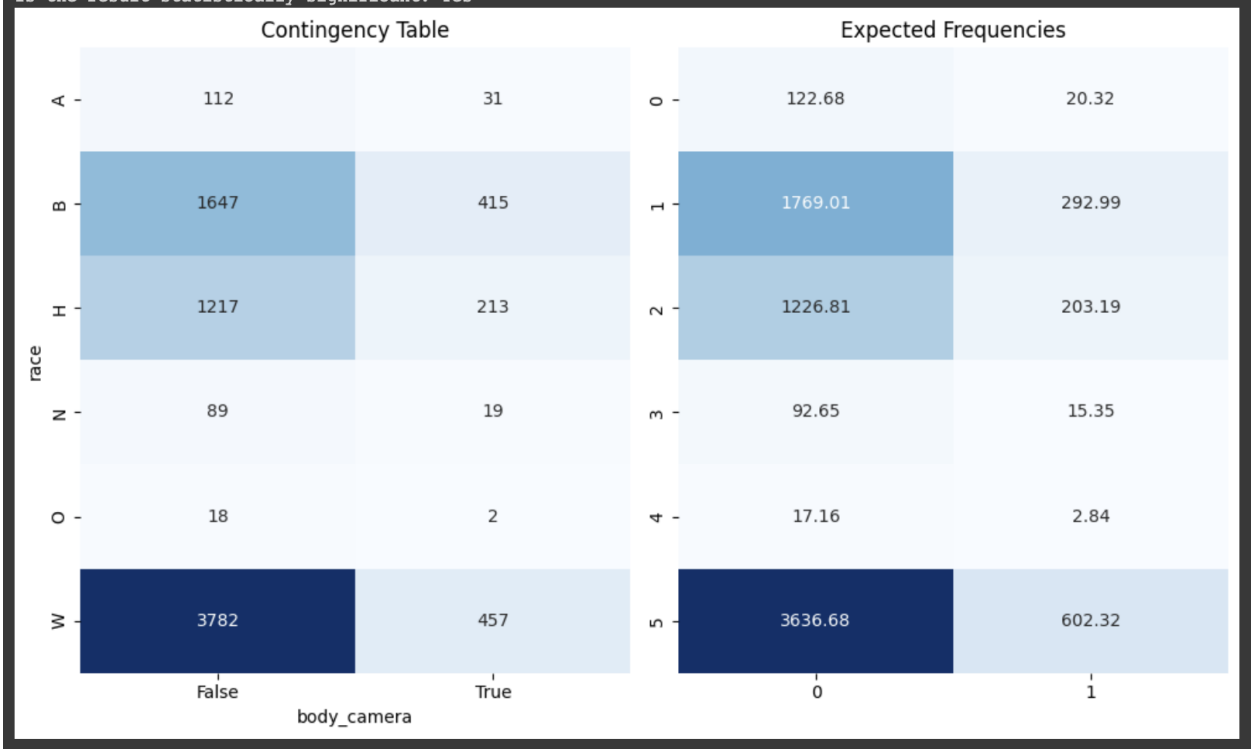
	PredictedClusterID	flee	armed	threat_level	age_group	Count
0	0	car	knife	other	Adult	19
1	1	foot	gun	attack	Adolescents	24
2	2	Notfleeing	unarmed	attack	Adult	10
3	3	foot	gun	other	Adult	13

In this picture: *we can clearly see the trend in the clusters of why the people in these clusters were being shot*



Visualizing the trend of people being shot using the bar graphs

```
Chi-squared statistic: 108.49418618642981
P-value: 8.524478119592931e-22
Degrees of Freedom: 5
Expected Frequencies:
[[1.22681205e+02 2.03187953e+01]
 [1.76901150e+03 2.92988503e+02]
 [1.22681205e+03 2.03187953e+02]
 [9.26543364e+01 1.53456636e+01]
 [1.71582104e+01 2.84178955e+00]
 [3.63668270e+03 6.02317296e+02]]
Is the result statistically significant? Yes
```



The extremely low p-value suggests that there is a significant association between race and body camera status.

Appendix C: Code

The code can be viewed in this GitHub repo
Link: https://github.com/nikhil9066/Police-shooting-MTH_P2

Code 1: Given details of a person, determine the race of a particular person?

```
import pandas as pd
from sklearn import tree

#choose from different tunable hyper parameters
clf = tree.DecisionTreeClassifier(max_depth=3,criterion='entropy')

#Printing all the parameters of Decision Trees
print(clf)

#Creating the model on Training Data
DTree=clf.fit(X_train,y_train)
prediction=DTree.predict(X_test)

#Measuring accuracy on Testing Data
from sklearn import metrics
print(metrics.classification_report(y_test, prediction))
print(metrics.confusion_matrix(y_test, prediction))

#Plotting the feature importance for Top 10 most important columns
feature_importances = pd.Series(DTree.feature_importances_,
index=Predictors)
feature_importances.nlargest(10).plot(kind='barh')

#Printing some sample values of prediction
TestingDataResults=pd.DataFrame(data=X_test, columns=Predictors)
TestingDataResults['TargetColumn']=y_test
TestingDataResults['Prediction']=prediction
TestingDataResults.head()

# Plotting the decision tree
plt.figure(figsize=(20, 10))
tree.plot_tree(DTree, filled=True, feature_names=Predictors,
class_names=True, rounded=True)
plt.show()
```

Code 2: Find the index of the row with the maximum count for each cluster

```
import matplotlib.pyplot as plt
import seaborn as sns

cluster_incident_info = df_cal.groupby(['PredictedClusterID', 'flee',
'armed', 'threat_level','age_group']).size().reset_index(name='Count')
cluster_incident_info.sort_values(by='Count', ascending=False)

# Find the index of the row with the maximum count for each cluster
max_count_index =
cluster_incident_info.groupby('PredictedClusterID')['Count'].idxmax()

max_count_rows = cluster_incident_info.loc[max_count_index]
max_count_rows
sns.set(style="whitegrid")

fig, axes = plt.subplots(3, 2, figsize=(15, 15))

# Plotting for 'age_group'
sns.barplot(x='PredictedClusterID', y='age_group',
data=cluster_incident_info, palette="viridis", ax=axes[0, 0])
axes[0, 0].set_title('Mode of Age Group')

sns.countplot(x='threat_level', data=cluster_incident_info,
palette="viridis", ax=axes[0, 1])
axes[0, 1].set_title('Mode of Threat Level')

sns.countplot(x='armed', data=cluster_incident_info, palette="viridis",
ax=axes[1, 0])
axes[1, 0].set_title('Mode of Armed Status')

sns.countplot(x='flee', data=cluster_incident_info, palette="viridis",
ax=axes[1, 1])
axes[1, 1].set_title('Mode of Flee Status')
sns.barplot(x='PredictedClusterID', y='Count', data=cluster_incident_info,
palette="viridis", ax=axes[2, 0])
axes[2, 0].set_title('Count in Each Predicted Cluster')

fig.delaxes(axes[2, 1])

plt.tight_layout()
plt.show()
```

Code 3: The probability of a Male being shot when the person is from the race

```
#Cleaning the data
df=pd.read_excel("fatal-police-shootings-data_original.xls")

df_cal=df[df['state']=='CA']

# Prior probability of being shot (replace with your actual prior knowledge
or estimate)
prior_prob_shot = df_cal[df_cal['manner_of_death'] == 'shot and
Tasered'].shape[0] / df_cal.shape[0]

# Prior probability of being male (replace with your actual prior knowledge
or estimate)
prior_prob_male = df_cal[df_cal['gender'] == 'M'].shape[0] /
df_cal.shape[0]

# Prior probability of being Black (replace with your actual prior
knowledge or estimate)
prior_prob_black = df_cal[df_cal['race'] == 'H'].shape[0] / df_cal.shape[0]

# Likelihood of being shot given that the person is male and Black
likelihood_shot_given_male_and_black = df_cal[(df_cal['gender'] == 'M') &
(df_cal['race'] == 'H') & (df_cal['manner_of_death'] == 'shot and
Tasered')].shape[0] / df_cal[(df_cal['gender'] == 'M') & (df_cal['race'] ==
'H')].shape[0]

# Apply Bayes' theorem
posterior_prob_male_shot_given_black =
(likelihood_shot_given_male_and_black * prior_prob_shot) / prior_prob_black

print(f"The posterior probability of a Male being shot when the person is
from the 'H' race is approximately:
{posterior_prob_male_shot_given_black:.4f}")
```

Code 4: was there any bias when there was a body cam on any race

```
contingency_table = pd.crosstab(df['race'], df['body_camera'])

# Perform the chi-squared test
chi2_stat, p_val, dof, expected = chi2_contingency(contingency_table)

# Determine if the result is statistically significant (typically using a
significance level of 0.05)
alpha = 0.05
is_significant = p_val < alpha

print(f"Chi-squared statistic: {chi2_stat}")
print(f"P-value: {p_val}")
print(f"Degrees of Freedom: {dof}")
print(f"Expected Frequencies:\n{expected}")
print(f"Is the result statistically significant? {'Yes' if is_significant
else 'No'}")

# Visualization
plt.figure(figsize=(10, 6))

# Contingency Table Heatmap
plt.subplot(1, 2, 1)
sns.heatmap(contingency_table, annot=True, fmt='d', cmap='Blues',
cbar=False)
plt.title('Contingency Table')

# Expected Frequencies Heatmap
plt.subplot(1, 2, 2)
sns.heatmap(expected, annot=True, fmt='.2f', cmap='Blues', cbar=False)
plt.title('Expected Frequencies')

plt.tight_layout()
plt.show()


                                                                    #
Findings
# The extremely low p-value suggests that there is a significant
association between race and body camera status.
```