ADITYA LALWANI D15C
ROLL NO. 30

Experiment 3

Aim: Perform Data Modeling. Perform following data modeling operations on your selected dataset:-

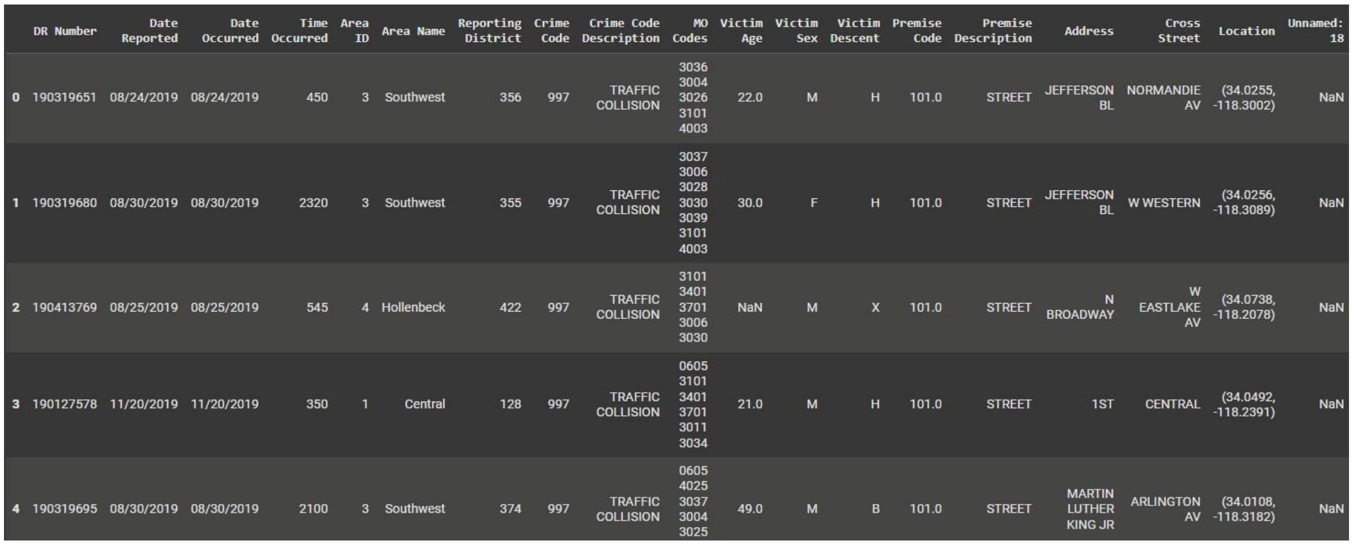
- Partition the data set, for example 75% of the records are included in the training data set and 25% are included in the test data set.
- Use a bar graph and other relevant graphs to confirm your proportions.
- Identify the total number of records in the training data set.
- Validate partition by performing a two-sample Z-test.

Performance:

- Prerequisite: Import all the required libraries: pandas for data manipulation and analysis, numpy for numerical computations, matplotlib.pyplot for basic plotting and data visualization, and seaborn for enhanced statistical graphics. Additionally, use statsmodels' ztest for performing statistical hypothesis testing and sklearn's train_test_split for splitting data into training and testing sets. Also, load data into Pandas:

Command:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.stats.weightstats import ztest
from sklearn.model_selection import train_test_split
df = pd.read_csv('dataset.csv')
df.head()
```



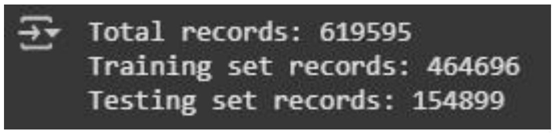
	DR Number	Date Reported	Date Occurred	Time Occurred	Area ID	Area Name	Reporting District	Crime Code	Crime Code Description	MO Codes	Victim Age	Victim Sex	Victim Descent	Premise Code	Premise Description	Address	Cross Street	Location	Unnamed: 18
0	190319651	08/24/2019	08/24/2019	450	3	Southwest	356	997	TRAFFIC COLLISION	3036 3004 3026 3101 4003	22.0	M	H	101.0	STREET	JEFFERSON BL	NORMANDIE AV	(34.0255, -118.3002)	NaN
1	190319680	08/30/2019	08/30/2019	2320	3	Southwest	355	997	TRAFFIC COLLISION	3037 3006 3028 3030 3039 3101 4003	30.0	F	H	101.0	STREET	JEFFERSON BL	W WESTERN	(34.0256, -118.3089)	NaN
2	190413769	08/25/2019	08/25/2019	545	4	Hollenbeck	422	997	TRAFFIC COLLISION	3101 3401 3701 3006 3030	NaN	M	X	101.0	STREET	N BROADWAY	W EASTLAKE AV	(34.0738, -118.2078)	NaN
3	190127578	11/20/2019	11/20/2019	350	1	Central	128	997	TRAFFIC COLLISION	0605 3101 3401 3701 3011 3034	21.0	M	H	101.0	STREET	1ST	CENTRAL	(34.0492, -118.2391)	NaN
4	190319695	08/30/2019	08/30/2019	2100	3	Southwest	374	997	TRAFFIC COLLISION	0605 4025 3037 3004 3025	49.0	M	B	101.0	STREET	MARTIN LUTHER KING JR	ARLINGTON AV	(34.0108, -118.3182)	NaN

ADITYA LALWANI D15C
ROLL NO. 30

- Partition the data set, for example 75% of the records are included in the training data set and 25% are included in the test data set:

Command:
```
train_df, test_df = train_test_split(df, test_size=0.25, random_state=42)
print(f"Total records: {len(df)}")
print(f"Training set records: {len(train_df)}")
print(f"Testing set records: {len(test_df)}")
```

The above code splits the dataset into 75% training data and 25% testing data using train_test_split(), with random_state=42 ensuring the split remains consistent across runs, and then prints the total number of records along with the sizes of the training and testing sets.

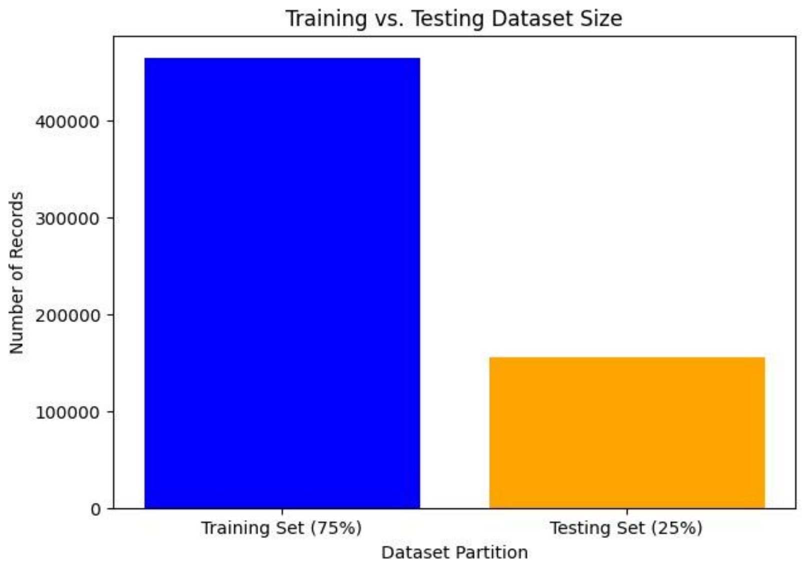


```
Total records: 619595
Training set records: 464696
Testing set records: 154899
```

- Use a bar graph and other relevant graphs to confirm your proportions:

1. Bar graph:-

Command:
```
data_counts = [len(train_df), len(test_df)]
labels = ['Training Set (75%)', 'Testing Set (25%)']
plt.figure(figsize=(7,5))
plt.bar(labels, data_counts, color=['blue', 'orange'])
plt.xlabel("Dataset Partition")
plt.ylabel("Number of Records")
plt.title("Training vs. Testing Dataset Size")
plt.show()
```
The above code generates a bar chart to visualize the dataset split, comparing the number of records in the training set (75%) and testing set (25%), helping to confirm the partitioning.



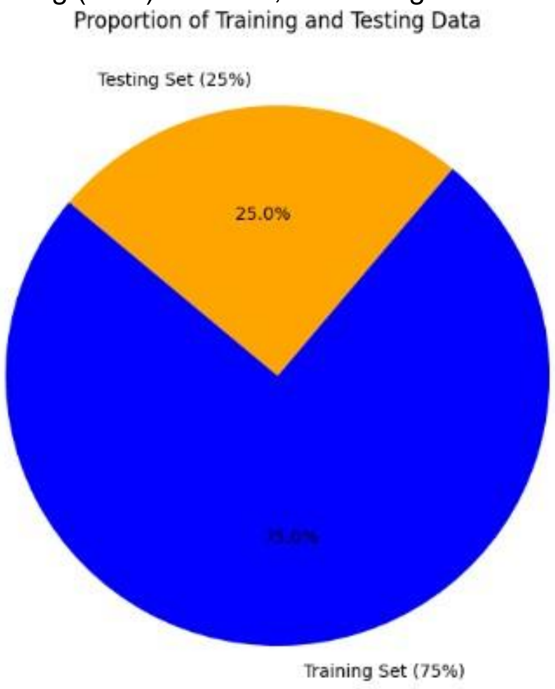
ADITYA LALWANI D15C
ROLL NO. 30

2. Pie Chart:-

<u>Command</u>:

```
plt.figure(figsize=(7,7))
plt.pie(data_counts, labels=labels, autopct='%1.1f%%', colors=['blue', 'orange'], startangle=140)
plt.title("Proportion of Training and Testing Data")
plt.show()
```

The above code generates a pie chart displaying the proportion of records in the training (75%) and testing (25%) datasets, confirming that the data was correctly partitioned.

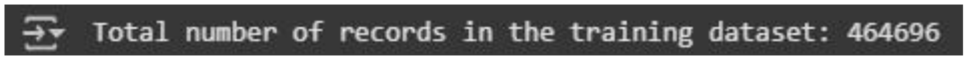


- Identify the total number of records in the training data set:

<u>Command</u>:

```
total_training_records = len(train_df)
print(f"Total number of records in the training dataset: {total_training_records}")
```

The above code calculates and prints the total number of records in the training dataset by determining the length of train_df.



```
Total number of records in the training dataset: 464696
```

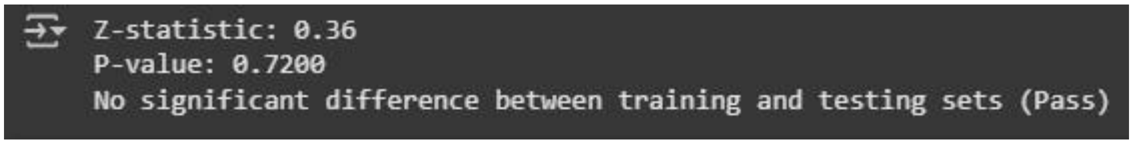
ADITYA LALWANI D15C
ROLL NO. 30

- Validate partition by performing a two-sample Z-test:

Command:

```
train_df_cleaned = train_df['Victim Age'].dropna()
test_df_cleaned = test_df['Victim Age'].dropna()
z_stat, p_value = ztest(train_df_cleaned, test_df_cleaned)
print(f"Z-statistic: {z_stat:.2f}")
print(f"P-value: {p_value:.4f}")
alpha = 0.05
if p_value > alpha:
    print("No significant difference between training and testing sets (Pass)")
else:
    print("Significant difference detected (Fail - Resampling recommended)")
```

The above code performs a two-sample Z-test on the Victim Age column to compare the means of the training and testing datasets, printing the Z-statistic (which measures the difference between the sample means in terms of standard errors) and the p-value (which indicates the probability of observing such a difference by chance). If the p-value is greater than 0.05, the partitioning is considered statistically valid.



```
Z-statistic: 0.36
P-value: 0.7200
No significant difference between training and testing sets (Pass)
```