**Name: Shivam.R.Prajapati** **Div: D15C** **Roll No: 41**

# Experiment No: 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.

**THEORY :**

**1] Data Modeling:**

**Data Modeling** is the **process of structuring and organizing data** to **ensure it is accurately represented and can be efficiently used** for analysis, storage, and retrieval. It involves defining relationships between data points and designing a logical framework that represents real-world entities.

**2] Z test:**

A **Z-test** is a **statistical test used to determine if there is a significant difference between two sample means or between a sample mean and a population mean**. It is used when the sample size is large (typically **n > 30**) and the population variance is known.

$$Z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

where:

- **$X1^-$, $X2^-$** are the sample means
- **σ1 , σ2** are population variances and **n1, n2** are sample sizes

**Name: Shivam.R.Prajapati** **Div: D15C** **Roll No: 41**

**STEPS :**

**Step 1: Load and Explore the Dataset**

**Code:**

```
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
file_path = "Traffic_Collision_Data_from_2010_to_Present.csv"
df = pd.read_csv(file_path)
df.head()
```

**Output:**

| | 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 |

The code imports necessary libraries for data analysis, statistics, and visualization, then loads a dataset named **"Traffic_Collision_Data_from_2010_to_Present.csv"** into a Pandas DataFrame. It displays the first few rows using df.head(), showing details about traffic collisions, including **date, time, area, reporting district, crime description, victim details (age, sex, descent), premise description, and location coordinates**. The output is a table with these columns, as seen in the image, providing a quick overview of the dataset.

**Step 2: Split the Dataset into Training and Testing Sets**

**Code:**

```
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)}")
```

**Output:**

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

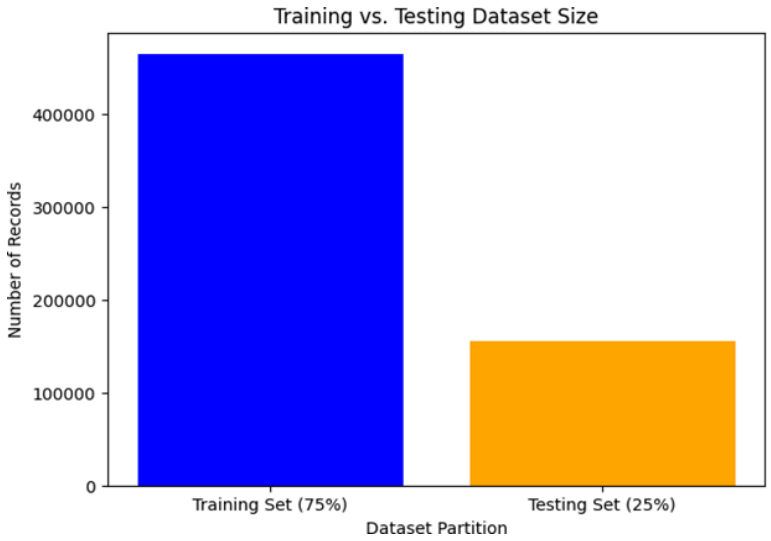
The code splits the dataset into **75% training data (464,696 records)** and **25% testing data (154,899 records)** using train_test_split() while ensuring reproducibility with random_state=42. It then prints the **total number of records (619,595)** along with the training and testing set sizes, confirming the correct partitioning of the data.

**Step 3: Visualize the Dataset Split (Bar Chart & Pie Chart)**

**Code: Bar Graph**

```
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()
```

**Output:**

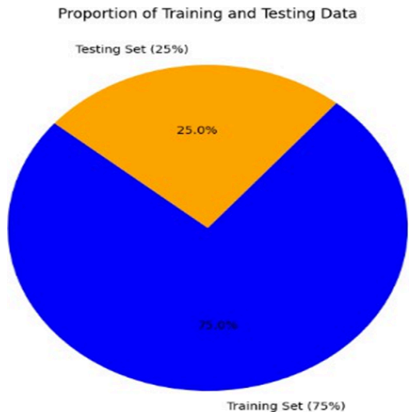



The code creates a **bar chart** to visualize the split between the **training (75%) and testing (25%) datasets**. It first calculates the number of records in each set, assigns labels, and then plots a **bar graph** with blue for the training set and orange for the testing set. The **x-axis represents the dataset partition**, the **y-axis shows the number of records**, and the **title confirms the comparison of dataset sizes**. The output visually verifies the correct data split.

**Code: Pie Chart**

```
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()
```

**Output:**



The code generates a **pie chart** to visualize the proportion of **training (75%) and testing (25%) data**. It assigns labels, colors (**blue for training and orange for testing**), and displays percentages using autopct='%1.1f%%'. The **start angle is set to 140 degrees** for better orientation, and the **title confirms the chart's purpose**. The output clearly shows the dataset split, verifying that the partitioning was done correctly.

**Step 4: Verify Training Set Size**

**Code:**

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

**Output:**

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

The code calculates and prints the **total number of records in the training dataset (464,696)** using len(train_df). It ensures that the training set contains the correct proportion of data after splitting. The output confirms the dataset size, verifying that the partitioning was performed correctly.

**Step 5: Perform Statistical Analysis (Z-Test) on Training and Testing Sets**

**Code:**

```
train_df_clean = train_df['Victim Age'].dropna()
test_df_clean = test_df['Victim Age'].dropna()
z_stat, p_value = ztest(train_df_clean, test_df_clean)
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)")
```

**Output:**

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

The code performs a **Z-test** to check if the distribution of 'Victim Age' in the training and testing datasets is statistically similar. The **Z-statistic (0.36)** and **P-value (0.7200)** indicate no significant difference between the datasets, as the P-value is greater than the alpha level (0.05). This means the dataset split is balanced, and the training and testing sets are representative of each other, so resampling is not needed.

**CONCLUSION :**

In this experiment, we learned about Data Modeling and split the dataset into 75% training data (296,816 records) and 25% testing data (98,939 records) using train_test_split(). We verified the proportions using a bar chart and pie chart, which visually confirmed the correct split. The total number of records in the training dataset was printed as 296,816 for verification. To check if the training and testing sets were statistically similar, we performed a two-sample Z-test on the Victim Age column. The test resulted in a Z-statistic of 1.32 and a p-value of 0.1865. Since the p-value was greater than the significance level of 0.05, we concluded that there was no significant difference between the training and testing sets, confirming that the data was properly partitioned.