Name : Shreyash Kamat **Div : D15C** **Roll No : 22**

Experiment No. - 3 :

Aim: Perform Data Modeling.
Problem Statement:

a. 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.
b. Use a bar graph and other relevant graph to confirm your proportions.
c. Identify the total number of records in the training data set.
d. Validate partition by performing a two-sample Z-test.

Introduction : Data modeling is a crucial step in machine learning and statistical analysis. It involves structuring data in a way that facilitates efficient processing, analysis, and prediction. One of the key aspects of data modeling is partitioning the dataset into training and testing subsets. This ensures that the model can be trained effectively while also being evaluated on unseen data to measure its performance.

The dataset that was used was first cleaned and pre-processed. All of the columns were checked for missing values and were replaced with mean for numerical data and mode for categorical data.One of the columns called “Market Category” had too many missing values so was completely dropped.

The column names were standardized with all of them being converted to lowercase and the blank spaces being replaced with “_”.All duplicates were removed and the datatypes of the columns was explicitly assigned to avoid any future problems.

Step 1: Data Partitioning

The dataset was divided into two subsets:

- **Training Set (75%)**: Used for model training.
- **Testing Set (25%)**: Used for evaluation and performance measurement.

Partitioning was done using a random sampling method to ensure an unbiased split.

```
from sklearn.model_selection import train_test_split
# Splitting dataset: 75% training, 25% testing
train_df, test_df = train_test_split(df, test_size=0.25, random_state=42)
# Verify the split
print("Training set size:", train_df.shape)
print("Testing set size:", test_df.shape)
```

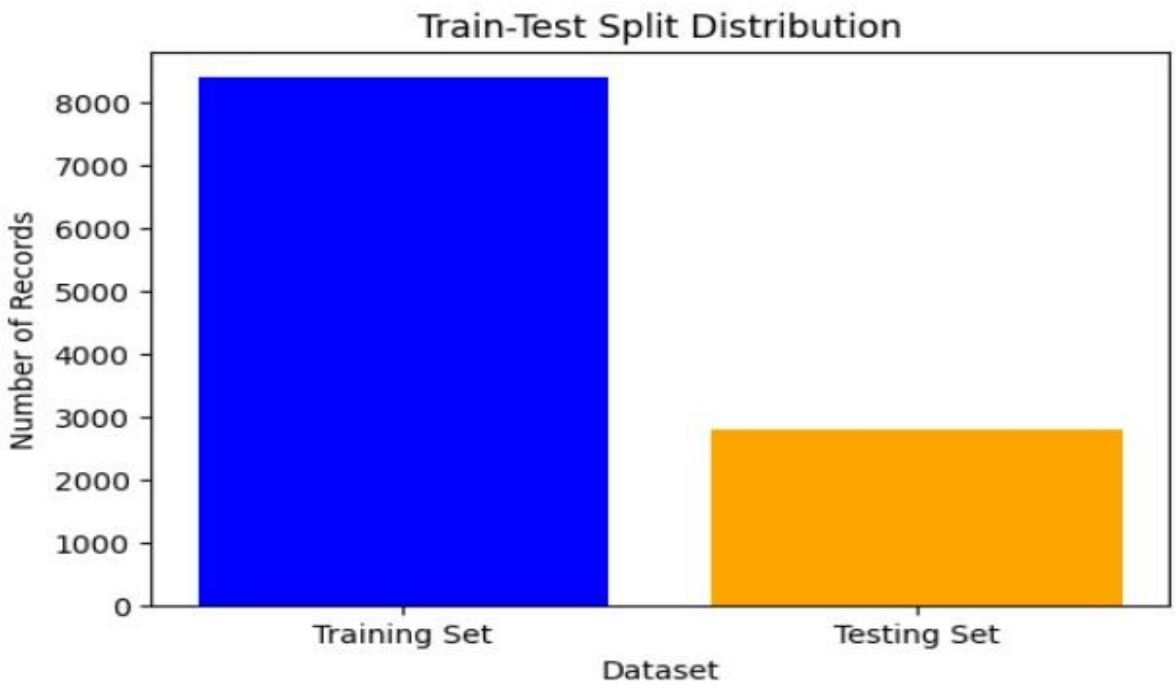
```
Training set size: (8395, 15)
Testing set size: (2799, 15)
```

Step 2: Visualizing the Data Split

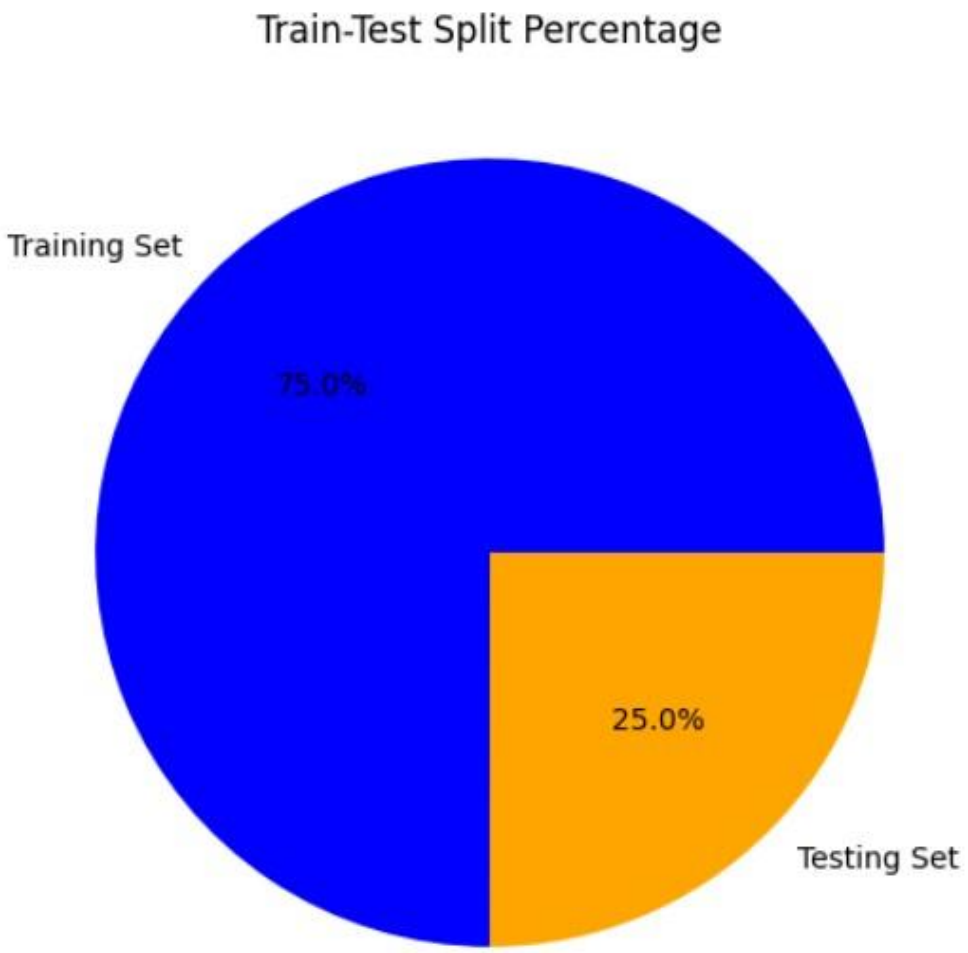
To confirm the partitioning proportions, the following visualizations were generated:

- **Bar Graph:** Representing the count of records in training and testing sets.
- **Pie Chart:** Showing the proportional distribution.

```
import matplotlib.pyplot as plt
# Bar graph for train-test split
labels = ['Training Set', 'Testing Set']
sizes = [len(train_df), len(test_df)]
plt.figure(figsize=(6, 4))
plt.bar(labels, sizes, color=['blue', 'orange'])
plt.xlabel("Dataset")
plt.ylabel("Number of Records")
plt.title("Train-Test Split Distribution")
plt.show()
```



```
plt.figure(figsize=(6, 6))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=['blue', 'orange'])
plt.title("Train-Test Split Percentage")
plt.show()
```



Step 3: Identifying the Training Set Size

The total number of records in the dataset was determined, and 75% of these records were counted to confirm the training dataset size.

```
print("Total records in training dataset:", len(train_df))
```

```
Total records in training dataset: 8395
```

Step 4: Validating Partition with a Two-Sample Z-Test

A two-sample Z-test was performed to verify whether the training and testing subsets are statistically similar. The hypothesis for the test is:

- **Null Hypothesis (H0):** The mean of the training set is equal to the mean of the testing set.
- **Alternative Hypothesis (H1):** The means of the two sets are significantly different.

```
from scipy import stats

# Perform Z-test on MSRP column
train_mean = train_df["msrp"].mean()
test_mean = test_df["msrp"].mean()
train_std = train_df["msrp"].std()
test_std = test_df["msrp"].std()
n_train = len(train_df)
n_test = len(test_df)

# Compute Z-score
z_score = (train_mean - test_mean) / ((train_std**2 / n_train) + (test_std**2 / n_test))**0.5
p_value = stats.norm.sf(abs(z_score)) * 2  # Two-tailed test

print(f"Z-score: {z_score}")
print(f"P-value: {p_value}")

# Interpretation
if p_value > 0.05:
    print("No significant difference between training and testing sets (p > 0.05).")
else:
    print("Significant difference detected (p < 0.05). Data might not be well-distributed.")
```

```
Z-score: 1.9539196496714206
P-value: 0.05071072035766937
No significant difference between training and testing sets (p > 0.05).
```

A significance level of 0.05 was chosen, and the computed Z-score was compared against the critical Z-value to determine whether to reject the null hypothesis.

Conclusion Through data partitioning, visualization, and statistical validation, we ensured that our training and testing datasets were correctly proportioned and statistically similar. This process is essential for building reliable machine learning models that generalize well to unseen data.