Aim: Perform data Data Modeling

Theory:

1. **Data Partitioning**

Data partitioning divides large datasets into smaller parts (partitions) to improve performance and scalability. Common types include:

1. List Partitioning: Divides data based on specific column values (e.g., customers from different countries).
2. Hash Partitioning: Uses a hash function on a column to distribute data evenly, useful when no natural partition exists.
3. Hybrid Partitioning: Uses both horizontal and vertical partitioning to split data efficiently

The choice depends on data size, access patterns, and system requirements.

2. **Hypothesis Testing**

Hypothesis testing is a statistical method used to make inferences or draw conclusions about the population based on a sample of data. It is a way to test the validity of a claim or idea, often referred to as a hypothesis, about a population parameter:

- Null Hypothesis (H0): Assumes no difference or effect (e.g., "Men and women are of the same height on average").
- Alternative Hypothesis (Ha): Suggests a difference exists (e.g., "Men are taller than women").
- P-Value: Probability of observing the data if H0 is true. A low p-value (< 0.05) suggests rejecting H0.
- Significance Level (α): The threshold below which the null hypothesis is rejected. Common choices are 0.01, 0.05, and 0.1. If the p-value is less than the significance level, the null hypothesis is rejected.
- One-Tailed & Two-Tailed Tests: One-tailed tests are used when the direction of the difference is known, such as when testing if a new drug is better than a placebo. Two-tailed tests are used when the direction of the difference is unknown, such as when testing if a new drug has a different effect than the current standard.
- Formula :

$$z = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\left(\sigma_1^2/n_1 + \sigma_2^2/n_2\right)}}$$

Steps:

1. Define H0 and Ha.

2. Choose α (e.g., 0.05).
3. Calculate test statistics and p-value.
4. Compare p-value with α and decide.
5. Interpret results:

Type I error occurs when the null hypothesis is rejected when it is actually true, leading to a false positive. A Type II error occurs when the null hypothesis is not rejected when it is actually false, leading to a false negative.

Output :

1. Load the dataset

```
import pandas as pd
df = pd.read_csv("/content/market_data.csv")
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Item_Identifier            8523 non-null   object
 1   Item_Weight                8523 non-null   float64
 2   Item_Fat_Content           8523 non-null   object
 3   Item_Visibility            8523 non-null   float64
 4   Item_Type                  8523 non-null   object
 5   Item_MRP                   8523 non-null   float64
 6   Outlet_Identifier          8523 non-null   object
 7   Outlet_Size                8523 non-null   object
 8   Outlet_Location_Type       8523 non-null   object
 9   Outlet_Type                8523 non-null   object
 10  Item_Outlet_Sales          8523 non-null   float64
 11  Sales_Bin                  8523 non-null   object
 12  MRP_Bin                    8523 non-null   object
 13  Item_Outlet_Sales_Capped   8523 non-null   float64
dtypes: float64(5), object(9)
memory usage: 932.3+ KB
```

2. For the given feature Item_MRP calculating the counts

```
df['Item_MRP'].value_counts()
```

Item_MRP	count
142.0154	6
172.0422	6
113.2834	5
188.1872	5
261.2910	5
...	...
219.5482	1
57.3930	1
105.8622	1
101.2990	1
75.4670	1

4694 rows × 1 columns

dtype: int64

3. Partition of the dataset that is dividing into training and testing of data in 75-25 where 75% of the records are included in the training data and rest 25% are included in the test data

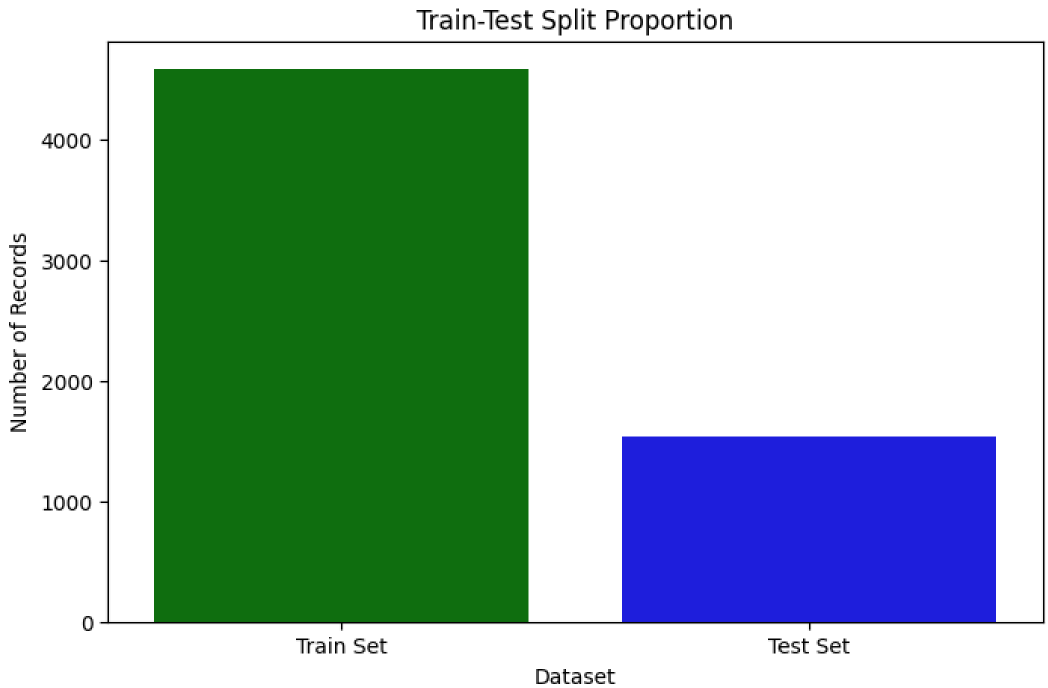
```
[5] import matplotlib.pyplot as plt
    import seaborn as sns
    import numpy as np
    from scipy.stats import norm
    from sklearn.model_selection import train_test_split
```

```
[6] train_df, test_df = train_test_split(df, test_size=0.25, random_state=50)

    train_size = len(train_df)
    test_size = len(test_df)

    plt.figure(figsize=(8, 5))
    sns.barplot(x=['Train Set', 'Test Set'], y=[train_size, test_size], palette=['green', 'blue'])
    plt.xlabel('Dataset')
    plt.ylabel('Number of Records')
    plt.title('Train-Test Split Proportion')
    plt.show()
```

4. Visualization using a bar graph to confirm the proportions of the data split into training and test sets



5. Total number of records in the split

```
Total records in Training Set: 4584
Total records in Testing Set: 1529
```

6. Now using a two-sample Z-test to validate whether the split was biased or unbiased

```
def sample_test(sample1, sample2):
  mean1, mean2 = np.mean(sample1), np.mean(sample2)
  std1, std2 = np.std(sample1, ddof=1), np.std(sample2, ddof=1)
  n1, n2 = len(sample1), len(sample2)

  z_score = (mean1 - mean2) / np.sqrt((std1**2 / n1) + (std2**2 / n2))

  p_value = 2 * (1 - norm.cdf(abs(z_score)))
  return z_score, p_value
```

7. Now this test is performed for the comparison of **Item_Mrp** column in training and testing dataset. The Z-statistic was calculated based on the means, standard deviations, and sizes of both samples. If p-value < significance level (α=0.05), we reject the null hypothesis.

```
Z-score: -0.26839427745723915
P-value: 0.7883958439089263
No significant difference found (p > 0.05). The partitioning is valid.
```

This validates that the partitioning process preserves the original distribution, ensuring that both sets are representative of the overall data.

Conclusion:
A two-sample z-test to verify whether the partitioning of the Item_MRP (Maximum Retail Price) column into training and testing sets was statistically balanced. The results showed a p-value greater than 0.05, indicating that there is no significant difference in the mean MRP between the two partitions. This confirms that the data splitting process maintains a consistent distribution, ensuring that the model is trained and evaluated on representative data without any bias due to MRP variations