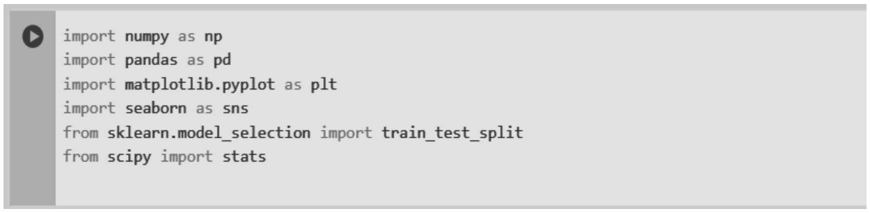
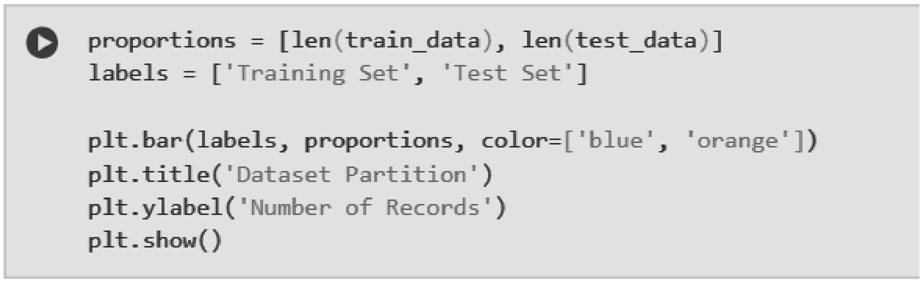
Aim: Perform Data Modeling on the dataset

Output :

1. 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.The total records



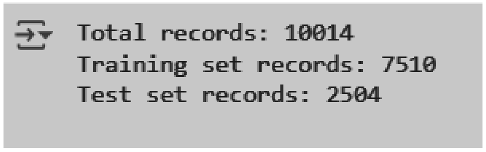
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from scipy import stats
```



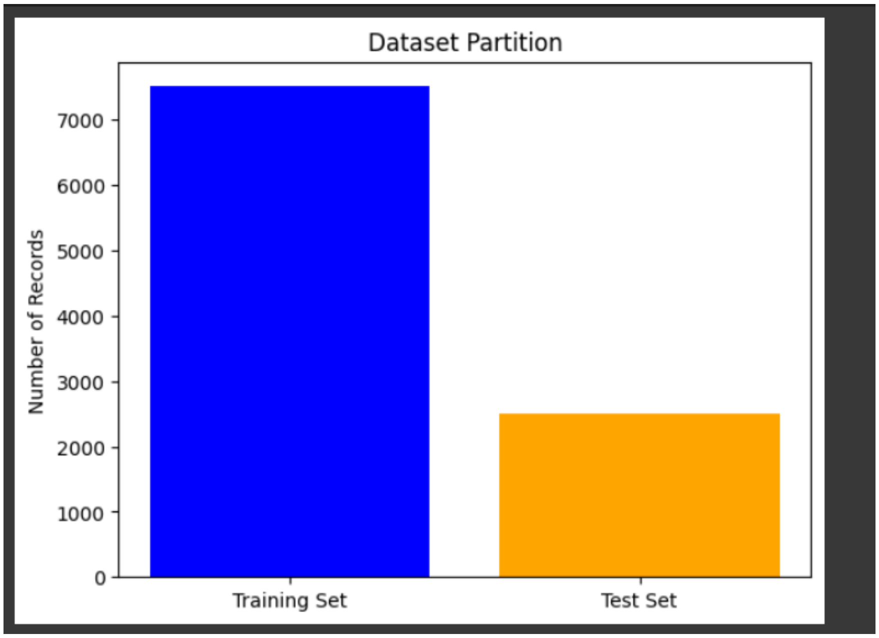
```
proportions = [len(train_data), len(test_data)]
labels = ['Training Set', 'Test Set']

plt.bar(labels, proportions, color=['blue', 'orange'])
plt.title('Dataset Partition')
plt.ylabel('Number of Records')
plt.show()
```

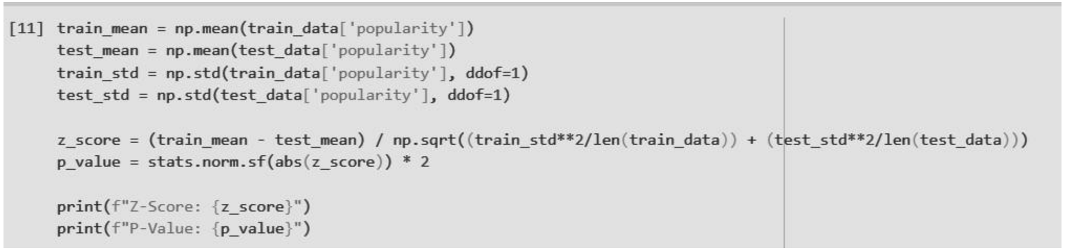
2. Visualization using a bar graph to confirm the proportions of the data split into training and test sets and checking the records that are split up



```
Total records: 10014
Training set records: 7510
Test set records: 2504
```



3. Using a **two-sample Z-test**, we evaluated whether the data split introduced any bias in the popularity distribution

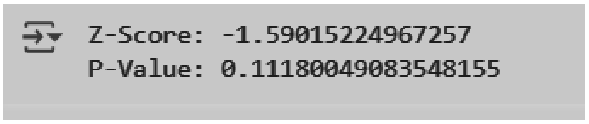


```
[11] train_mean = np.mean(train_data['popularity'])
     test_mean = np.mean(test_data['popularity'])
     train_std = np.std(train_data['popularity'], ddof=1)
     test_std = np.std(test_data['popularity'], ddof=1)

     z_score = (train_mean - test_mean) / np.sqrt((train_std**2/len(train_data)) + (test_std**2/len(test_data)))
     p_value = stats.norm.sf(abs(z_score)) * 2

     print(f"Z-Score: {z_score}")
     print(f"P-Value: {p_value}")
```

4. Now this test is performed for the comparison of **Populari**ty column in training and testing dataset. The Z-statistic was calculated based on the means, standard deviations, and sizes of both samples.



```
Z-Score: -1.59015224967257
P-Value: 0.11180049083548155
```

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