NAME: Atharva Shinde CLASS: D15C ROLL_NO.: 54

Aim: **Perform Data Modelling – Partitioning the dataset.**

Theory:

Importance of data Partitioning.

Partitioning data into **train** and **test** splits is a fundamental practice in machine learning and statistical modeling. This division is crucial for ensuring that models generalize well to unseen data and do not overfit to the training dataset. Below is a detailed explanation of why this partitioning is important:

1. Evaluation of Model Generalization

- **Purpose**: The primary goal of machine learning is to build models that perform well on **unseen data**, not just the data they were trained on. Partitioning the data into train and test sets allows us to simulate this scenario.
- **Mechanism**: The **training set** is used to train the model, while the **test set** acts as a proxy for unseen data. By evaluating the model on the test set, we can estimate how well the model is likely to perform on new, real-world data.
- **Risk of Not Partitioning**: Without a separate test set, we risk overestimating the model's performance because the model may simply memorize the training data (overfitting) rather than learning generalizable patterns.

2. Avoiding Optimistic Bias

- **Optimistic Bias**: If the same data is used for both training and evaluation, the model's performance metrics (e.g., accuracy, precision, recall) will be overly optimistic. This is because the model has already "seen" the data and may have memorized it.
- **Test Set as a Safeguard**: The test set acts as a safeguard against this bias, providing a more realistic measure of the model's performance.

3. Detection of Overfitting

- **Overfitting Definition**: Overfitting occurs when a model learns the noise or specific details of the training data, leading to poor performance on new data.

- **Role of Test Set**: The test set provides an independent evaluation of the model. If the model performs well on the training set but poorly on the test set, it is a clear indication of overfitting.
- **Example**: A model achieving 99% accuracy on the training set but only 60% on the test set suggests that it has overfitted to the training data.

Visual Representation

Using a bar graph to visualize a 75:25 train-test split is an effective way to clearly communicate the distribution of the dataset. The graph provides an immediate visual representation of the proportions, making it easy to see that 75% of the data is allocated for training and 25% for testing. This clarity ensures that the split is transparent and well-understood, which is crucial for validating the model's development process.

Additionally, the bar graph highlights whether the split is balanced and appropriate for the task at hand. A 75:25 ratio is a common and practical division, and visualizing it helps confirm that the test set is large enough to provide a reliable evaluation of the model's performance. This visual justification reinforces the credibility of our data preparation and modeling approach.

Z-Testing:

Key Idea: **Fair Evaluation, Partitioning Issues.**

The two-sample Z-test is a statistical hypothesis test used to determine whether the means of two independent samples are significantly different from each other. It assumes that the data follows a normal distribution and that the population variances are known (or the sample sizes are large enough for the Central Limit Theorem to apply). The test calculates a Z-score, which measures how many standard deviations the difference between the sample means lies from zero. This score is then compared to a critical value or used to compute a p-value to determine statistical significance.

The primary use case of the Z-test is to compare the means of two groups and assess whether any observed difference is due to random chance or a true underlying difference. In the context of dataset partitioning, the Z-test can be used to validate whether the train and test splits are statistically similar. For example, by comparing the means of a key feature (e.g., age, income) across the two splits, we can ensure that the partitioning process did not introduce bias and that both sets are representative of the same population.

The significance of the Z-test lies in its ability to provide a quantitative measure of similarity between datasets. If the p-value is greater than the chosen significance level (e.g., 0.05), we can conclude that the splits are statistically similar, ensuring a fair and reliable evaluation of the model. This step is crucial for maintaining the integrity of the machine learning workflow and ensuring that the model's performance metrics are trustworthy.

Steps:

Imported train_test_split **from** sklearn.model_selection**:**

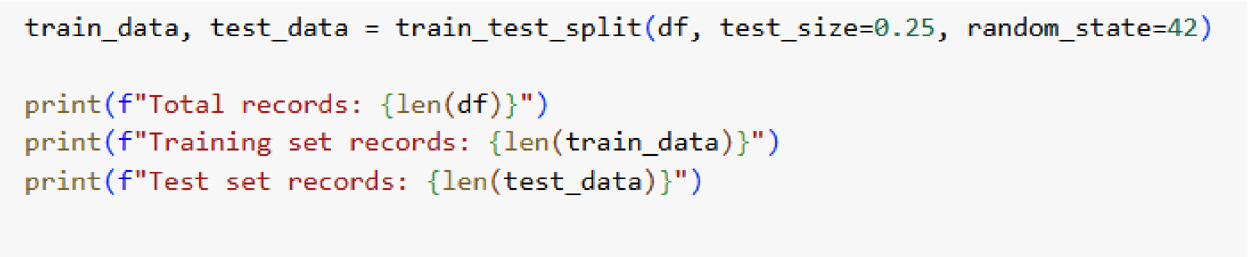
- This function is used to split arrays or matrices into random train and test subsets.

Split Features and Target Variable:

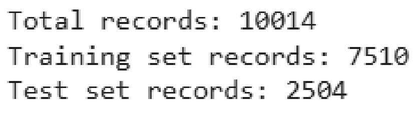
- **Features (**X**):** We created a dataframe X by dropping the 'Total' column from df. This dataframe contains all the feature variables except the target.
- **Target (**y**):** We created a series y which contains the 'Total' column from df. This series is our target variable.

Partitioned the Data:

- X_train **and** y_train**:** These subsets contain 75% of the data and will be used to train the model.
- X_test **and** y_test**:** These subsets contain the remaining 25% of the data and will be used to test the model's performance.



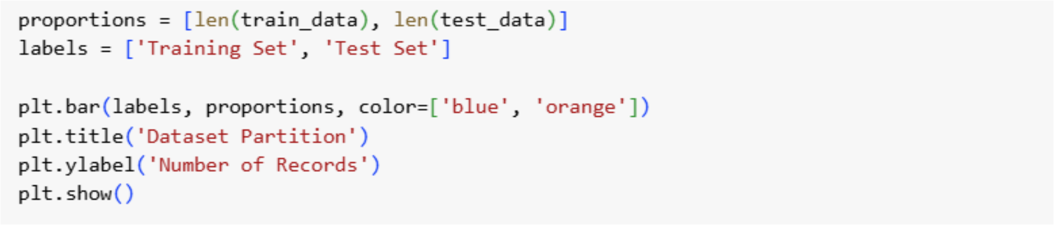
```
train_data, test_data = train_test_split(df, test_size=0.25, random_state=42)

print(f"Total records: {len(df)}")
print(f"Training set records: {len(train_data)}")
print(f"Test set records: {len(test_data)}")
```



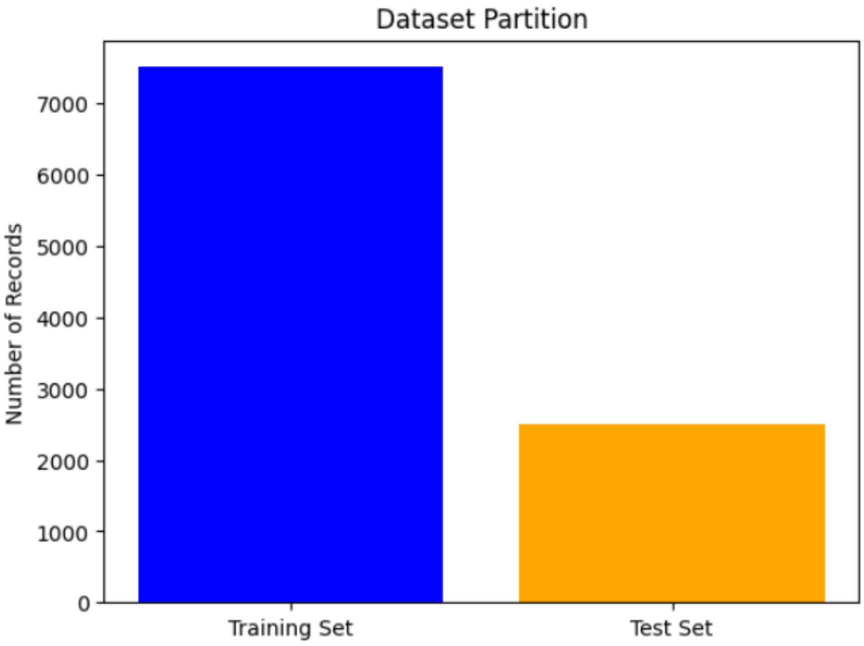
```
Total records: 10014
Training set records: 7510
Test set records: 2504
```

Visualizing the split.

- **plt.bar(labels, sizes, color=['blue', 'orange']): This function creates a bar graph with the specified labels and sizes. The bars are colored blue for training data and orange for test data.**
- **plt.title('Proportion of Training and Test Data (Features & Target)'): This sets the title of the graph.**
- **plt.ylabel('Number of Samples'): This sets the label for the y-axis, indicating the number of samples.**
- **plt.show(): This function displays the graph.**

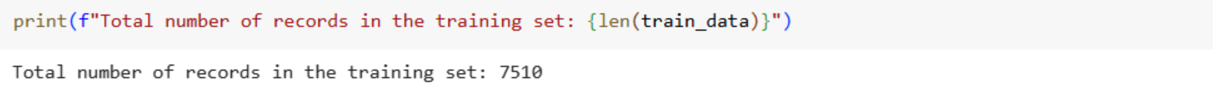


```
proportions = [len(train_data), len(test_data)]
labels = ['Training Set', 'Test Set']

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



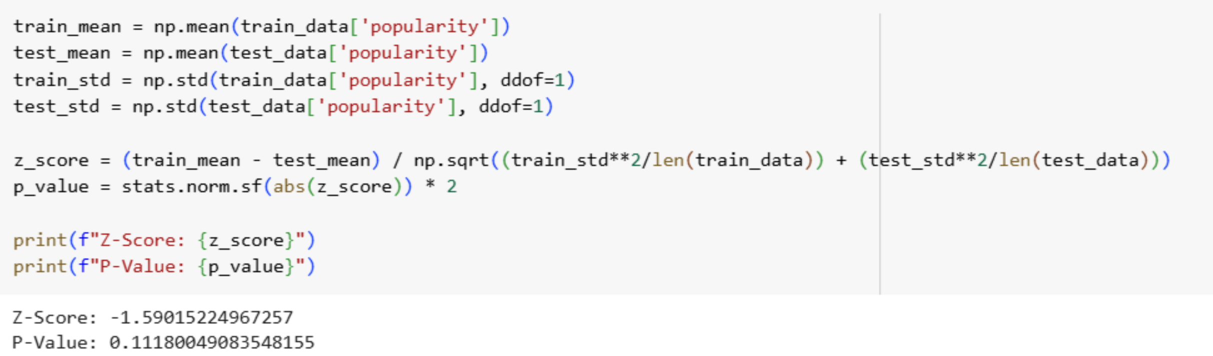
Significance of the Output:

- **Z-Statistic:**
  - Indicates the number of standard deviations by which the mean of the training set differs from the mean of the test set.
- **P-Value:**
  - Helps determine the significance of the Z-statistic. A low P-value (< 0.05) suggests that the difference is statistically significant.



```
print(f"Total number of records in the training set: {len(train_data)}")
```

```
Total number of records in the training set: 7510
```



```
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}")
```

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

Inference from the Output:

- **Interpretation:**

In this analysis, the P-Value obtained is 0.1118, which is greater than the significance level of 0.05. This indicates that there is no statistically significant difference between the training and test sets. Therefore, it can be inferred that both sets are likely drawn from the same distribution, ensuring that the model will be evaluated on data that is representative of what it was trained on.

This is an scenario for building a machine learning model, as it reduces the risk of biased performance metrics and improves the model's generalization ability. The consistent distribution between the training and test sets confirms the effectiveness of the data-splitting strategy used in this study.

Conclusion:

In this experiment, we successfully partitioned the dataset into **training and test sets** using a 75:25 split ratio, ensuring a robust foundation for model development and evaluation. The partitioning was visualized using a bar graph, which clearly illustrated the proportion of data allocated to each set, confirming that the split was appropriately balanced.

The dataset was divided into a training set (75%) and a test set (25%), resulting in 7,510 records for model training and 2,504 for model evaluation. A statistical hypothesis test was performed to compare the means of the popularity attribute in the training and test sets. The resulting Z-Score of -1.59 and P-Value of 0.1118 indicate no statistically significant difference between the two groups, confirming that the split maintained the population distribution's integrity. This suggests that the dataset is well-shuffled and that the training and test sets are representative of the overall data distribution.