MET Bhujbal Knowledge City
Institute of Technology- Polytechnic
Adgaon, Nashik- 422 003.

Date :

Practical No : 3

AIM :

To implement train-test split on a given dataset to evaluate the performance of a machine learning model

Input :

A dataset (eg csv file) containg multiple features & target variable

output :

evaluation metrics such as accuracy precision, recall or mean squared error (depending on the model used)

Theory :

In ML data is usually devided into two set.

1) Training set : Used to train the model's. allowing it to learn pattens from the data.

2) Testing set : used to evaluate the model's performance on unseen data.

The train-test split is a technique where the dataset

is divided into training & testing subsets, commonly used on 80-20 or 70-30 ratio. the Scikit-learn library in python provides a function train-test split() to facilate this process.

- Need for train-test split:
- ML models must be evaluated on data they have not seen before to ensure proper generalization.

- Training a model on the entire dataset can lead to overfitting, where the model memorizes the data instead of learning patterns.

- A seperate test set allows us to measure real-world model performance ensuring its ability to handle new inputs effectively.

- Significance of train-test split:

- prevents overfitting: ensures that the model generalizes well to unseen data.

- provides unbiased evaluation: The test data acts as unseen data to assess model accuracy.

- Helps in hyperparameter: Splitting data allows fine tuning of model parameters before final development.

- efficient model selection :
  Helps in comparing different models to select the best performance one.

- Algorithm :

1) Import necessary libraries such as pandas, sklearn, model_selection, & sklearn.metrics.

2) load the dataset using pandas : read_csv().

4) Split the dataset into training & testing sets using train_test_split(x, y, test_size = 0.2, random_state = 42).

5) Train a machine learning model on the training set.

6) predict the ~~target~~ variable on the testing set.

7) Evaluate the model using performance metrics such as accuracy (for classification) or RMSE (for regression).

8) display results.

Program :

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# load dataset
df = pd.read_csv("dataset.csv")

# define features & target
x = df.drop("target", axis=1)
y = df["target"]

# split the data
x_train, x_test, y_train, y_test = train_test_split(
```

```
# train model
model = LogisticRegression()
model.fit(x_train, y_train)

# Predict & evaluate
y_pred = model.predict(x_test)
```

accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

Conclusion:

The train test split method is an essential step in machine learning to evaluate model performance. By keeping a portion of the data for testing, we can ensure that the model generalized well to unseen data thereby reducing overfitting & improve predictive accuracy.

Refernce:

1) Scikit learn documentation:

https://scikit-learn.org/stable/modules/cross-validation.html

2) machine learning

https://www.coursera.org.

3) Python data science handbook by Jake vanderplas

https://jakevdp.github.io

flowchart