

A Objective

In []:

B Synopsis & Algorithm

In [1]: *# In k-fold cross validation The dataset is divided into k equal-sized folds.
#The model is trained on k-1 folds and validated on the remaining fold.
#This process is repeated k times, with each fold used once as the validation set.*

C Code and Output

k-fold Cross Validation

In [2]: `from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import KFold, cross_val_score
import numpy as np`

In [3]: `# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target`

In [4]: `# Initialize the Decision Tree classifier
model = DecisionTreeClassifier()`

In [5]: `# Initialize K-Fold Cross-Validation
kf = KFold(n_splits=5, shuffle=True, random_state=1)`

In [6]: `# Perform cross-validation
scores = cross_val_score(model, X, y, cv=kf)`

In [7]: `# Print the results
print("Cross-Validation Scores:", scores)
print("Mean Accuracy:", np.mean(scores))`

Cross-Validation Scores: [0.96666667 0.96666667 0.96666667 0.93333333 0.86666667]
Mean Accuracy: 0.93999999999999998

Leave One Out Cross Validation

In [8]: `from sklearn.datasets import load_iris
from sklearn.model_selection import LeaveOneOut, cross_val_score
from sklearn.tree import DecisionTreeClassifier
import numpy as np`

In [9]: `# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target`

In [10]: `# Initialize the Decision Tree classifier
model = DecisionTreeClassifier()`

In [11]: `# Initialize Leave-One-Out Cross-Validation
looc = LeaveOneOut()`

In [12]: `# Perform LOOCV
scores = cross_val_score(model, X, y, cv=looc)`

```
In [13]: # Print the results
print("Number of splits:", looc.get_n_splits(X))
print("Cross-Validation Scores:", scores)
print("Mean Accuracy:", np.mean(scores))
```

Number of splits: 150
Cross-Validation Scores: [1.
1.
1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 0.
1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
1. 1. 1. 1. 1.]
Mean Accuracy: 0.9466666666666667

Stratified k-fold Cross Validation

```
In [14]: from sklearn.datasets import load_iris
from sklearn.model_selection import StratifiedKFold, cross_val_score
from sklearn.tree import DecisionTreeClassifier
import numpy as np
```

```
In [15]: # Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target
```

```
In [16]: # Initialize the Decision Tree classifier
model = DecisionTreeClassifier()
```

```
In [17]: # Initialize Stratified K-Fold Cross-Validation
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=1)

# Perform Stratified K-Fold Cross-Validation
scores = cross_val_score(model, X, y, cv=skf)
```

```
In [18]: # Print the results
print("Cross-Validation Scores:", scores)
print("Mean Accuracy:", np.mean(scores))
```

```
Cross-Validation Scores: [0.96666667 1.          0.9          1.          0.86666667]
Mean Accuracy: 0.9466666666666667
```

D Conclusions & Discussion

In [19]: *#Discuss what inferences you have made after the successful completion of the Experiment.*

E Viva Voce Questions

```
In [20]: #1 What is cross validation?
```

In [21]: #2 What is meant by Resampling?

In [22]: #3 What is Leave One Out Cross (LOOC) Validation?

```
In [23]: #4 What is Stratified Cross Validation?
```