National University of Computer & Emerging Sciences, Karachi

FALL 2024, LAB MANUAL – 04

DECISION TREE

COURSE CODE :	**AL3002**
INSTRUCTOR :	**Usama Bin Umar**

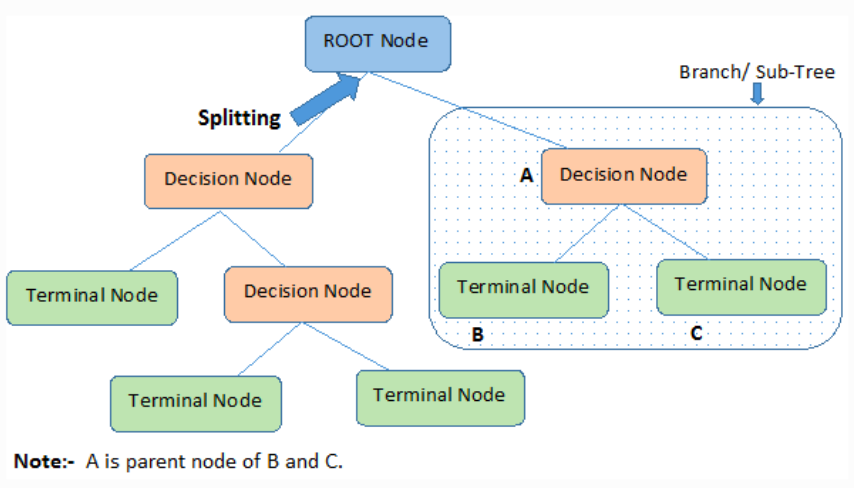
OBJECTIVE

1. Implementation of Decision Tree
2. Different algorithms for decision tree

DECISION TREE CLASSIFIER:

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Decision trees can be used for classification as well as regression problems. The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits. It starts with a root node and ends with a decision made by leaves.



DECISION TREE TERMINOLOGIES

Root Node: The first split which decides the entire population or sample data should further get divided into two or more homogeneous sets

Splitting: It is a process of dividing a node into two or more sub-nodes

Decision Node: This node decides whether/when a sub-node splits into further sub-nodes or not

Leaf Node: Terminal Node that predicts the outcome (categorical or continuous value).

Branch: A subsection of the entire tree is called branch or sub-tree.

Parent Node: A node divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

STEPS TO IMPLEMENT DECISION TREE

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

Step 2: Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**

Step 3: Divide the S into subsets that contain possible values for the best attributes.

Step 4: Generate the decision tree node containing the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node a leaf node.

ATTRIBUTE SELECTION MEASURE

If the dataset consists of **N** attributes, deciding which attribute to place at the root or at different levels of the tree as internal nodes is a complicated step. Just randomly selecting any node to be the root can't solve the issue. If we follow a random approach, it may give us bad results with low accuracy.

- Entropy,
- Information gain
- Gini index
- Gain Ratio
- Reduction in Variance
- Chi-Square

There are three methods to create a decision tree:

- ID3 Algorithm (based on entropy and information gain)
- CART Algorithm (based on Gini index and Gini gain)
- C4.5 Algorithm (based on gain ratio)

Let's study the working of the ID3 Algorithm so that you may know how the decision tree algorithm works from scratch: Steps in making a decision tree using the ID3 algorithm:

Implementing Decision Tree Classifier

Training model using DT

```
from sklearn.tree import DecisionTreeClassifier
DT = DecisionTreeClassifier()
ModelDT = DT.fit(x_train,y_train)
```

Model Testing

```
PredictionDT = DT.predict(x_test)
print(PredictionDT)
```

Model Training Accuracy

```
print("====================DT Training Accuarcy============")
tracDT=DT.score(x_train,y_train)
trainingAccDT=tracDT*100
print(trainingAccDT)
```

Model Testing Accuracy

```
print("====================DT Testing Accuracy============")
teacDT=accuracy_score(y_test,PredictionDT)
testingAccDT=teacDT*100
print(testingAccDT)
```

Decision Tree Classifier Parameters

```
class sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None,
min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None,
random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None,
ccp_alpha=0.0)
```

CRITERION:
The function is to measure the quality of a split. Supported criteria are "Gini" for the Gini impurity and "log_loss" and "entropy" both for the Shannon information gain,

SPLITTER:
The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.

MAXIMUM DEPTH:
The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

Read Sklearn Official Documentation, to know more about DT Parameter.

Decision Tree using Entropy without Pruning Parameter

```
#Decision Tree without Pruning Parameter
dt = DecisionTreeClassifier( criterion='entropy')
dt.fit(X_train, Y_train) # training the model
predictions = dt.predict(X_test) # now testing on test data to get class of test
tracDT=DT.score(x_train,y_train) #Testing on Training Data
print('Training Accuracy : ', tracDT*100)
print('Testing Accuracy : ',(accuracy_score(Y_test, predictions)))
```

Decision Tree using Entropy with Pruning Parameter

```
#Decision Tree with Pruning Parameter
dt = DecisionTreeClassifier( criterion='entropy', ccp_alpha = 0.015)
```

Decision Tree using Gini Index without Pruning Parameter

```
#Decision Tree using Gini Index without Pruning Parameter
dt = DecisionTreeClassifier( criterion='gini')
```

Decision Tree using Gini Index with Pruning Parameter

```
#Decision Tree using Gini Index with Pruning Parameter
dt = DecisionTreeClassifier( criterion='gini', ccp_alpha = 0.015)
```

How to Visualize a Decision Tree

Visualizing a Decision Tree

```
dot_data = tree.export_graphviz(dt,
                      feature_names=fn ,
                     filled=True, rounded=True,
                       special_characters=True)
graph = graphviz.Source(dot_data)
graph
```

TASKS

TASK 1: Complete the assignment that is attached to this lab as "Assignment - Decision tree". Use the "decision tree library" and fill in the table given in that assignment file.

TASK 2: Find out the root node of the decision tree from scratch on the below dataset (Age, Job_Status , Own_House , Credit Rating) using the ID3 algorithm. You can create this dataset in an Excelfile. (IG should be calculated for all the input Attributes).

Our Data: Loan Approval Prediction

ID	AGE	JOB_STATUS	OWNS_HOUSE	CREDIT_RATING	CLASS (Yes or No)
1	Young	False	False	Fair	No
2	Young	False	False	Good	No
3	Young	True	False	Good	Yes
4	Young	True	True	Fair	Yes
5	Young	False	False	Fair	No
6	Middle	False	False	Fair	No
7	Middle	False	False	Good	No
8	Middle	True	True	Good	Yes
9	Middle	False	True	Excellent	Yes
10	Middle	False	True	Excellent	Yes
11	Old	False	True	Excellent	Yes
12	Old	False	True	Good	Yes
13	Old	True	False	Good	Yes
14	Old	True	False	Excellent	Yes
15	Old	False	False	Fair	No

TASK 3 :

Download the dataset

- Perform EDA
- Check whether the dataset is balanced or not (using target variable “Label”)
- Check whether there is any empty records, categorical feature, or duplicate records, yes Then handle this and give a brief explanation why you have chosen this technique in a text cell or “jupyter/colab”
- Check the correlation of your dataset and perform feature selection using Pearson Correlation
- Analyze your dataset and think if feature scaling is required or not. If yes then apply any scaling technique based on your distribution.

- Split your dataset in training, testing, and validation. The train split will be 80% and the test will be 20%. In the validation split your training samples will be 70% and the validation set will be 30%. Briefly describe why we use a validation set in a text cell. Declare Random_state=0
- Apply DT and check model training and testing accuracy.

TASK 4:

Download the dataset and the Research Article. Analyze the article, and what work has been done by this author. Briefly discuss the aim and achievement of this paper in a text cell.
Perform the data analysis as mentioned in this article.
Plot the same graphs that have been used in this article.
Perform Data Analysis to achieve the results that are mentioned in this article.
Make Sure, You have completed all the work that has been done in this Paper.

TASK 5

Find out the root node of the decision tree from scratch on the **below** dataset (**Student, Prior_Experience, Course, Time**) using the **CART algorithm**. You can create this dataset in **an Excel**file.

Student	Prior Experience	Course	Time	Liked
1	Yes	Programming	Day	Yes
2	No	Programming	Day	No
3	Yes	History	Night	No
4	No	Programming	Night	Yes
5	Yes	English	Day	Yes
6	No	Programming	Day	No
7	Yes	Programming	Day	No
8	Yes	Mathematics	Night	Yes
9	Yes	Programming	Night	Yes
10	Yes	Programming	Night	No