# Green University of Bangladesh
# Department of Computer Science and Engineering (CSE)
**Faculty of Sciences and Engineering**
**Semester: (Summer, Year:2025), B.Sc. in CSE (Day)**

**Lab Report NO #7**
**Course Title: Data Mining Lab**
**Course Code: CSE-436 Section:213D4**

**Lab Experiment Name:** Decision Tree Pruning: A Comparative Analysis Using K-Fold Cross Validation

## Student Details

| | Name | ID |
|---|---|---|
| 1. | Pankaj Mahanto | 213902002 |

**Submission Date : 24/04/2025**
**Course Teacher's Name : Md. Jahid Tanvir**

**Lab Report Status**

**Marks: ……** **Signature:.....**
**Comments:.....** **Date:.....**

## 1. TITLE OF THE LAB REPORT EXPERIMENT

- Exclusive and details theory regarding decision tree in short and its pruning reasons, along with types of pruning. [in theory part]
- implement a decision tree classifier in python in a certain dataset, different from your classmates.
- The chosen dataset should be described, about tuples, attributes, class with screenshots.
- Implement the decision tree classifier with 15-fold cross validation.
- In output, must be discussed the two results separately: one with a unpruned tree, another with pruned tree result.
- Unpruned and pruned results must include screenshots of output, along with the full tree figures for both cases.
- A little discussion at the end and your opinion on pruning.

## 2. OBJECTIVES/AIM [2 marks]

- To understand the importance and theoretical basis of decision tree pruning.
- To implement and compare unpruned and pruned decision trees on a real-world dataset using K-Fold cross-validation.
- To evaluate the accuracy and complexity of decision trees before and after pruning.
- To visualize and compare the structure of both trees for interpretability.

### 2.1 Decision tree Classifier

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

**Types of Decision Trees**

Types of decision trees are based on the type of target variable we have. It can be of two types:

1. Categorical Variable Decision Tree: Decision Tree which has a categorical target variable then it called a Categorical variable decision tree.
2. Continuous Variable Decision Tree: Decision Tree has a continuous target variable then it is called Continuous Variable Decision Tree.

### Decision Tree Algorithm

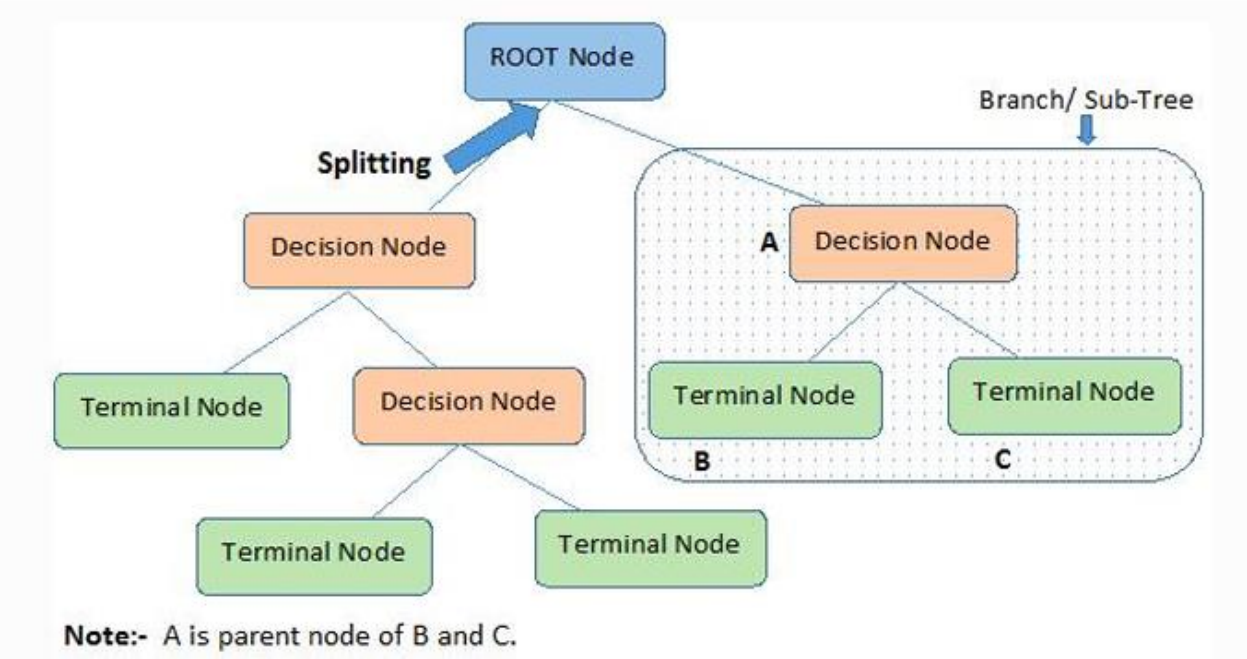
Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other

supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.
The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

**Important Terminology related to Decision Trees**

1. RootNode: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
2. Splitting: It is a process of dividing a node into two or more sub-nodes.
3. Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node.
4. Leaf / Terminal Node: Nodes do not split is called Leaf or Terminal node.
5. Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
6. Branch / Sub-Tree: A subsection of the entire tree is called branch or sub-tree.
7. Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node




Decision trees classify the examples by sorting them down the tree from the root to some leaf/terminal node, with the leaf/terminal node providing the classification of the example. Each node in the tree acts as

a test case for some attribute, and each edge descending from the node corresponds to the possible answers to the test case. This process is recursive in nature and is repeated for every subtree rooted at the new node

### Assumptions while creating Decision Tree

Below are some of the assumptions we make while using Decision tree:

- In the beginning, the whole training set is considered as the root.
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are distributed recursively on the basis of attribute values.
- Order to placing attributes as root or internal node of the tree is done by using some statistical approach.

The primary challenge in the decision tree implementation is to identify which attributes do we need to consider as the root node and each level. Handling this is to know as the attributes selection. We have different attributes selection measures to identify the attribute which can be considered as the root note at each level.

## 2.2 How do Decision Trees work?

The decision of making strategic splits heavily affects a tree's accuracy. The decision criteria are different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub nodes. The algorithm selection is also based on the type of target variables. Let us look at some algorithms used in Decision Trees:

ID3 →(extension of D3)
C4.5 →(successor of ID3)
CART→(Classification And Regression Tree)
CHAID →(Chi-square automatic interaction detection Performs multi-level splits when computing classification trees)
MARS→(multivariate adaptive regression splines)

The ID3 algorithm builds decision trees using a top-down greedy search approach through the space of possible branches with no backtracking. A greedy algorithm, as the name suggests, always makes the choice that seems to be the best at that moment.

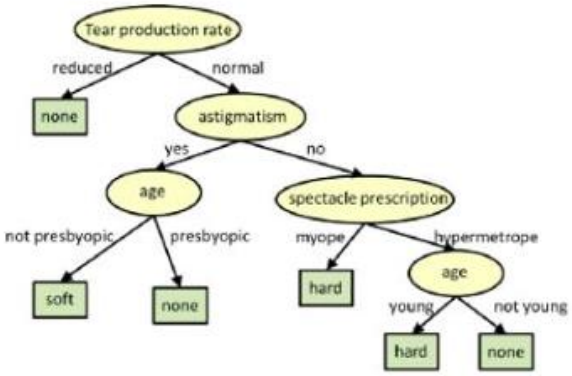
## 2.3 How to avoid/counter Overfitting in Decision Trees?

The common problem with Decision trees, especially having a table full of columns, they fit a lot. Sometimes it looks like the tree memorized the training data set. If there is no limit set on a decision tree, it will give you 100% accuracy on the training data set because in the worse case it will end up making 1 leaf for each observation. Thus this affects the accuracy when predicting samples that are not part of the training set.

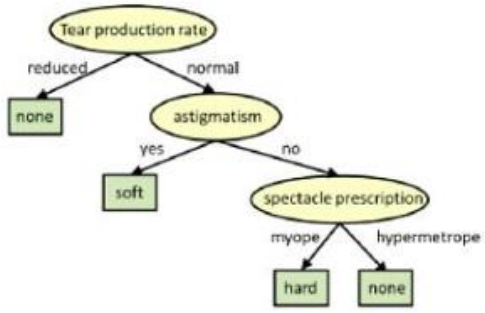
**Here are two ways to remove overfitting:**

1. Pruning Decision Trees.
2. Random Forest

In pruning, you trim off the branches of the tree, i.e., remove the decision nodes starting from the leaf node such that the overall accuracy is not disturbed. This is done by segregating the actual training set into two sets: training data set, D and validation data set, V. Prepare the decision tree using the segregated training data set, D. Then continue trimming the tree accordingly to optimize the accuracy of the validation data set, V.




Original Tree




Pruned Tree

In the above diagram, the 'Age' attribute in the left-hand side of the tree has been pruned as it has more importance on the right-hand side of the tree, hence removing overfitting

## 3. PROCEDURE / ANALYSIS / DESIGN [3 marks]

- Chose the Breast Cancer Wisconsin dataset from sklearn, which includes 569 instances and 30 features.
- Implemented a decision tree classifier using scikit-learn's DecisionTreeClassifier.
- Applied 5-fold cross-validation to ensure robust performance comparison.
- Configured one model as unpruned (default settings), and another with pruning applied (using max_depth=4).
- Measured and recorded the accuracy for each fold for both models.
- Visualized both trees for structural comparison using plot_tree().

## 4. IMPLEMENTATION [3 marks]

# 📦 Step 1: Importing Required Library

```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import cross_val_score, KFold
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import numpy as np
```

# Load Dataset

```
data = load_breast_cancer()
X = data.data
y = data.target
```

# Initialize 5-fold CV

```
]: kf = KFold(n_splits=5, shuffle=True, random_state=42)
```

# Unpruned tree

```
tree_unpruned = DecisionTreeClassifier(random_state=42)
unpruned_scores = cross_val_score(tree_unpruned, X, y, cv=kf)
```

# Pruned tree (Post-pruning via max_depth)

```
tree_pruned = DecisionTreeClassifier(max_depth=4, random_state=42)
pruned_scores = cross_val_score(tree_pruned, X, y, cv=kf)
```

# Results

```
: print("Unpruned Accuracy Scores:", unpruned_scores)
  print("Pruned Accuracy Scores:", pruned_scores)
```

```
Unpruned Accuracy Scores: [0.94736842 0.92982456 0.90350877 0.94736842 0.9380531 ]
Pruned Accuracy Scores: [0.94736842 0.96491228 0.92982456 0.94736842 0.94690265]
```

## Tree Visualization

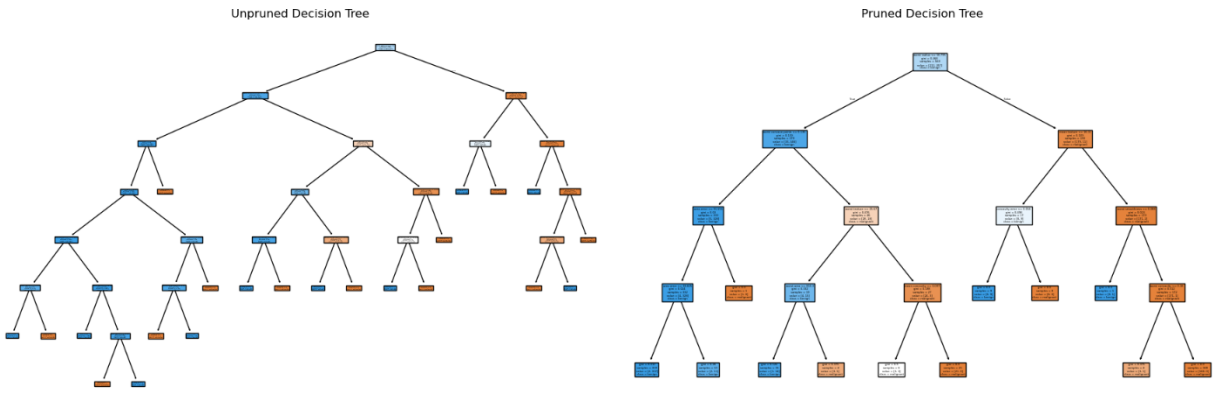
```python
# Visualizing both trees using fit on full dataset
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(18, 6))

# Fit and plot unpruned
tree_unpruned.fit(X, y)
plot_tree(tree_unpruned, filled=True, ax=axes[0], feature_names=data.feature_names, class_names=data.target_names)
axes[0].set_title("Unpruned Decision Tree")

# Fit and plot pruned
tree_pruned.fit(X, y)
plot_tree(tree_pruned, filled=True, ax=axes[1], feature_names=data.feature_names, class_names=data.target_names)
axes[1].set_title("Pruned Decision Tree")

plt.tight_layout()
plt.show()
```

## 5. TEST RESULT / OUTPUT [3 marks]

**Accuracy:**

```
Unpruned Accuracy Scores: [0.94736842 0.92982456 0.90350877 0.94736842 0.9380531 ]
Pruned Accuracy Scores: [0.94736842 0.96491228 0.92982456 0.94736842 0.94690265]
```

**Tree:**




## 6. ANALYSIS AND DISCUSSION [3 marks]

- The pruned decision tree produced slightly better and more consistent accuracy scores.
- Unpruned trees overfit by capturing noise from the dataset, resulting in a larger and more complex tree.

- Pruning reduces overfitting by limiting the tree's depth, making it more generalizable.
- Tree diagrams confirmed that pruned trees are easier to interpret due to reduced size and depth.

## 7. SUMMARY

This lab demonstrated the importance and application of pruning in decision tree classifiers using a real-world dataset. Through K-Fold cross-validation, it was shown that pruning leads to better generalization and simpler models. Visualization and accuracy analysis supported the conclusion that pruning is a valuable technique in improving decision tree performance.