Vidyavardhini's College of Engineering & Technology
Department of Computer Engineering
Academic Year: 2025-26

Experiment No. 5
Implement classification algorithm (Decision Tree)
Date of Performance: 13/08/25
Date of Submission: 20/08/25
Name: Sumit Metkari
Div/Roll no. : 2 / 32

Aim: To implement Decision Tree classification

Objective: Develop a program to implement decision tree classification

Theory:

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

A decision tree is a tree where each -

- Node - a feature(attribute)
- Branch - a decision(rule)

Leaf - an outcome(categorical or continuous

Basic algorithm (a greedy algorithm)

- Tree is constructed in a top-down recursive divide-and-conquer manner
- At start, all the training examples are at the root
- Attributes are categorical (if continuous-valued, they are discretized in advance)
- Examples are partitioned recursively based on selected attributes
- Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)

Conditions for stopping partitioning

- All samples for a given node belong to the same class
- There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
- There are no samples left

Entropy

- In ID3, entropy is calculated for each remaining attribute.
- The attribute with the smallest entropy is used to split the set S on that particular iteration.
- Entropy = 0 implies it is of pure class, that means all are of same category

Information Gain

- Information Gain is calculated for a split by subtracting the weighted entropies of each branch from the original entropy
- Information Gain IG(A) tells us how much uncertainty in S was reduced after splitting set S on attribute A.

ID3 stands for Iterative Dichotomiser 3

- It is a classification algorithm that follows a greedy approach by selecting a best attribute that yields maximum Information Gain(IG) or minimum Entropy(H).

Steps:

1. Calculate entropy for dataset.
2. For each attribute/feature.
    1. Calculate entropy for all its categorical values.
    2. Calculate information gain for the feature.
3. Find the feature with maximum information gain.

Repeat it until we get the desired tree

Code:

Importing Librararies

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
```

Loading the Dataset

[2]:
```
df=pd.read_csv("Iris.csv")
df.head()
```

[2]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

[68]:
```
dt_cv_model.best_params_
```

[68]:
```
{'criterion': 'entropy', 'max_depth': 4, 'max_features': 'log2'}
```

[69]:
```
dt_tuned = DecisionTreeClassifier(criterion = "gini", max_depth = 6, max_features = "sqrt")
```

[70]:
```
dt_tuned.fit(X_train, y_train)
```

[70]:
```
DecisionTreeClassifier(max_depth=6, max_features='sqrt')
```

[71]:
```
y_pred = dt_tuned.predict(X_test)
```

[72]:
```
print("Accuracy Score:", accuracy_score(y_test, y_pred))
```

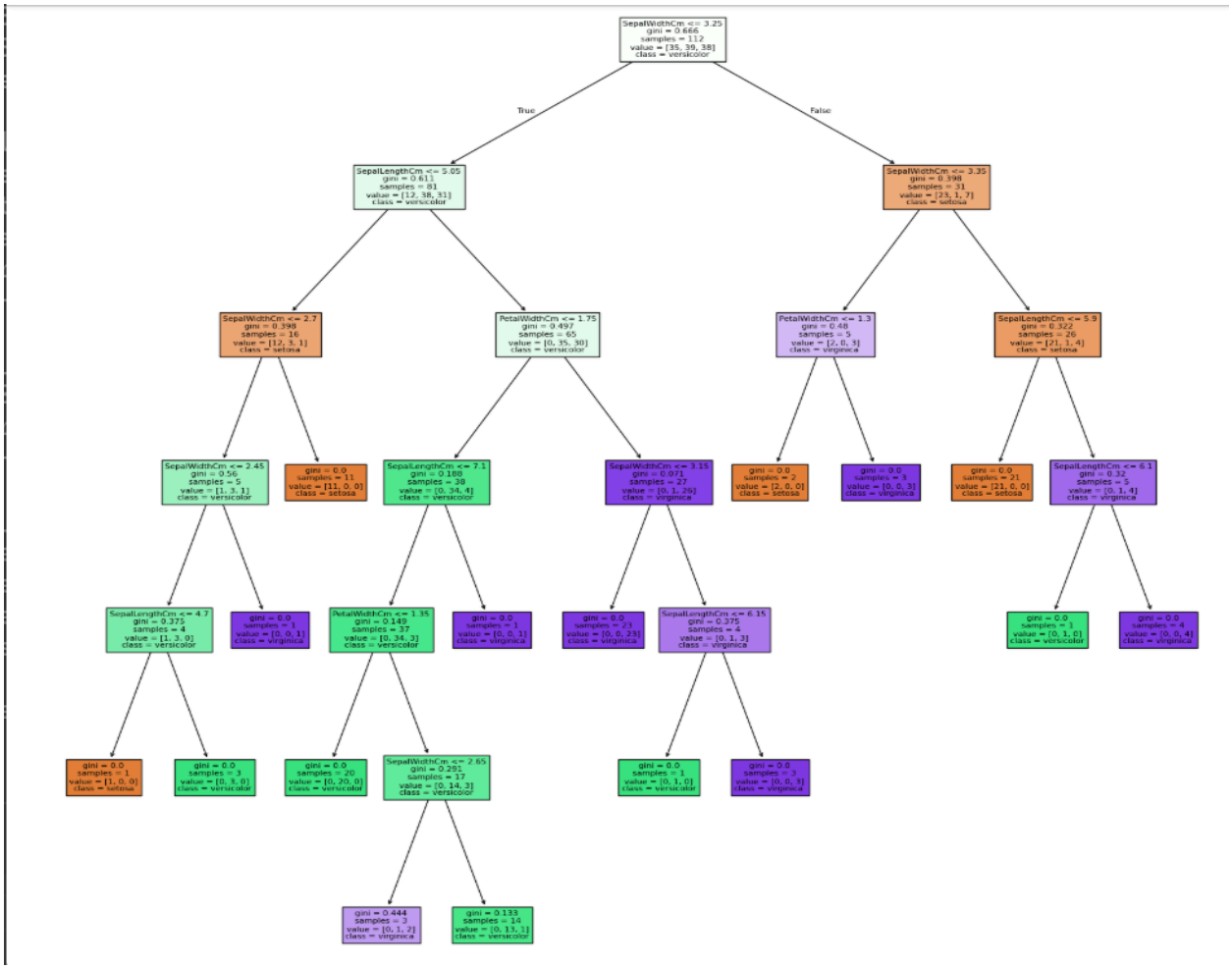
```
Accuracy Score: 0.9473684210526315
```

[73]:
```
cv_scores = cross_val_score(estimator = dt_tuned, X = X_train, y = y_train, cv=10)
cv_scores.mean()
```

[73]:
```
0.8939393939393938
```

[74]:
```
plt.figure(figsize=(20,20))
tree.plot_tree(dt_tuned, filled=True, feature_names=features , class_names = targets)
plt.show()
```

Output:



Conclusion:

The implemented Decision Tree classifier successfully built a model that classifies the given dataset by recursively splitting the data based on attributes with the highest information gain. The output demonstrates how the decision tree partitions the data into distinct classes, leading to clear and interpretable decision rules. The accuracy value obtained indicates the effectiveness of the model in correctly predicting the target classes. Overall, the Decision Tree algorithm proves to be a powerful and intuitive classification technique, capable of handling both categorical and continuous attributes, and providing easily understandable results. Its ability to generate a hierarchical structure of decisions makes it valuable in practical applications where model interpretability is important.