Lab Reort 08

Decision Tree

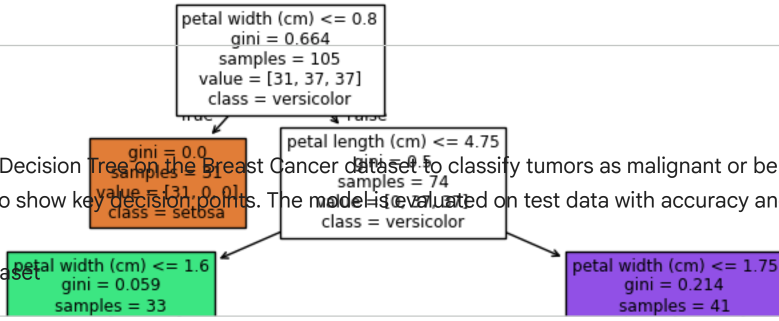
Name MaazAhmad
Reg No B23F722AI170
Section Ai yellow
Submitted to Abdullah Sajid

TASK 1

Basic Decision Tree on Iris & Breast Cancer

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import confusion_matrix, accuracy_score
import matplotlib.pyplot as plt

data = load_iris()
X, y = data.data, data.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

print("Training Accuracy:", clf.score(X_train, y_train))

plt.figure(figsize=(12,6))
plot_tree(clf, feature_names=data.feature_names, class_names=data.target_names, filled=True, max_depth=3)
plt.show()

y_pred = clf.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print("Test Accuracy:", accuracy_score(y_test, y_pred))
```

```
Training Accuracy: 1.0
```



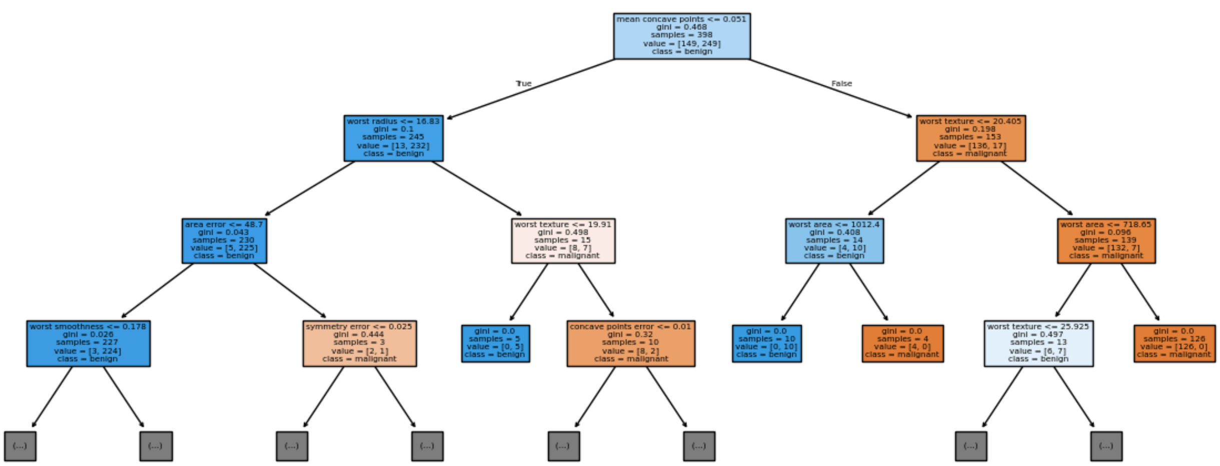
Summary

This code trains a Decision Tree on the Breast Cancer dataset to classify tumors as malignant or benign. It visualizes the first 3 levels of the decision tree to show key decision points. The model is evaluated on test data with accuracy and confusion matrix results.

Breast Cancer Dataset

```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import confusion_matrix, accuracy_score
import matplotlib.pyplot as plt

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

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

print("Training Accuracy:", clf.score(X_train, y_train))

plt.figure(figsize=(15,6))
plot_tree(clf, feature_names=data.feature_names, class_names=data.target_names, filled=True, max_depth=3)
plt.show()

y_pred = clf.predict(X_test)
print(confusion_matrix(y_test, y_pred))
print("Test Accuracy:", accuracy_score(y_test, y_pred))
```

```
Training Accuracy: 1.0
```




```
[[ 60   3]
 [  7 101]]
Test Accuracy: 0.9415204678362573
```

Start coding or generate with AI.

TASK 2

Gini vs Entropy Comparison

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

data = load_iris()
X, y = data.data, data.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

clf1 = DecisionTreeClassifier(criterion='gini')
clf2 = DecisionTreeClassifier(criterion='entropy')

clf1.fit(X_train, y_train)
clf2.fit(X_train, y_train)

pred1 = clf1.predict(X_test)
pred2 = clf2.predict(X_test)

print("Gini Accuracy:", accuracy_score(y_test, pred1))
print("Entropy Accuracy:", accuracy_score(y_test, pred2))
```

Summary

This code compares two Decision Tree models: one using Gini impurity and the other using Entropy. Both models are trained on the same dataset to observe differences in accuracy. The results show which criterion performs better for this specific dataset.

TASK 3

Visual Exploration of Tree

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier, plot_tree
import matplotlib.pyplot as plt

data = load_iris()
X, y = data.data, data.target

clf = DecisionTreeClassifier(max_depth=3)
clf.fit(X, y)

plt.figure(figsize=(12,6))
plot_tree(clf, feature_names=data.feature_names, class_names=data.target_names, filled=True)
plt.show()
```

Summary

This code trains a Decision Tree with a limited depth of 3 to make visualization clear. It plots the decision splits to show how the tree classifies data step-by-step. The visual helps understand which features influence decisions the most.

Summary of Lab

1. The lab introduced Decision Trees and explained how they make splits based on features to classify data more accurately.

2. We learned how impurity measures like Gini and Entropy determine the quality of splits.
3. The lab demonstrated how to train Decision Tree models on real datasets such as Iris and Breast Cancer.
4. Visualization of trees helped understand which features and thresholds are most important for decision-making.
5. We compared the performance of Gini vs Entropy, observing how different criteria affect accuracy.
6. Confusion matrices and accuracy scores were used to evaluate how well the models perform on unseen test data.
7. We observed how increasing tree depth can improve training accuracy but also risks overfitting.