Deep Panchal
TY - 9 - B - 38

Experiment no : 3

Aim: To implement and analyze the Decision Tree and Naïve Bayes classification algorithms using Python.

Introduction:

Classification is a fundamental machine learning technique used for predicting categorical outcomes. Two widely used classifiers are:

- **Decision Tree:** A tree-like structure where decisions are made based on feature conditions.
- **Naïve Bayes:** A probabilistic classifier based on Bayes' Theorem with an assumption of feature independence.

Procedure:

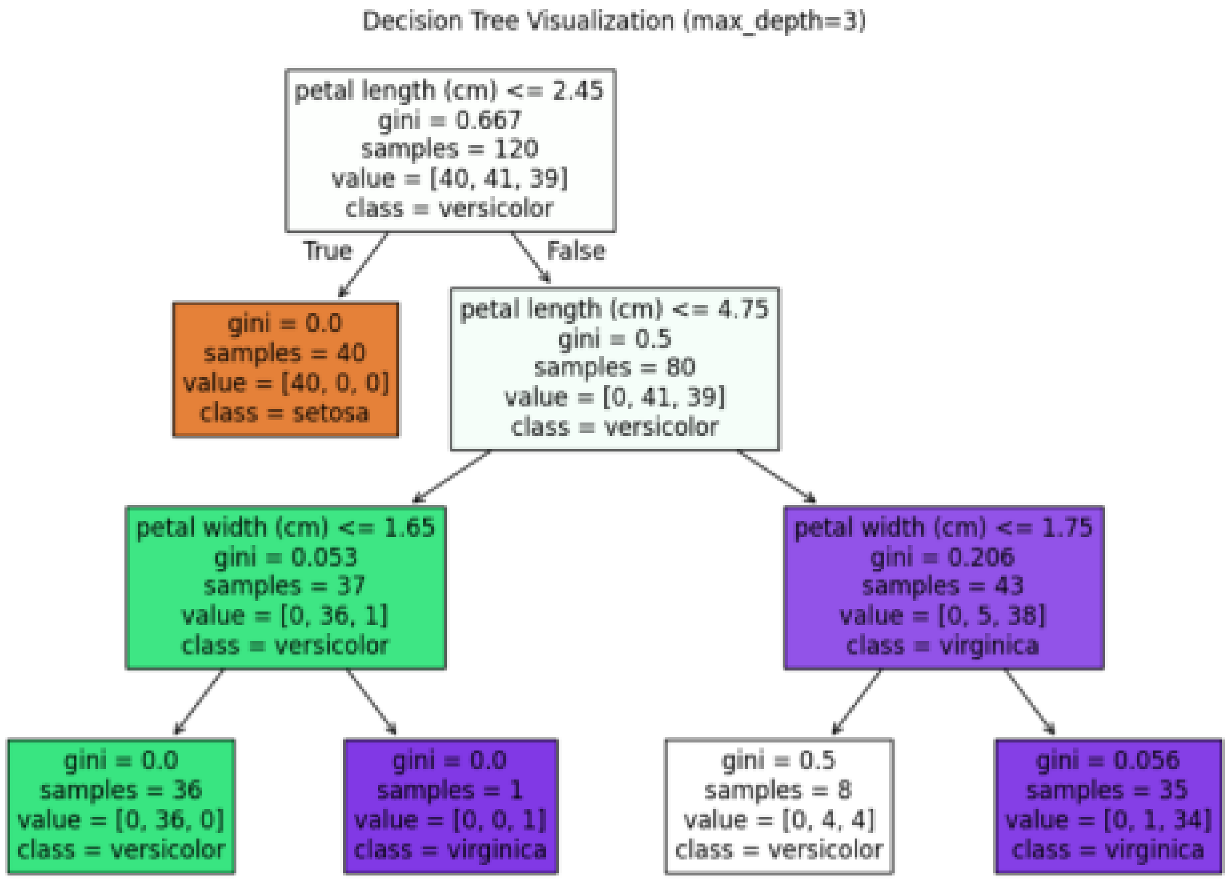
1. **Import Libraries: Load necessary Python libraries (numpy, matplotlib.pyplot, sklearn).**
2. **Load Dataset: Use the Iris dataset for classification.**
3. **Split Dataset: Divide data into training (80%) and testing (20%) sets.**
4. **Train Decision Tree Classifier:**
5. **Iterate over different max_depth values (1 to 10).**
6. **Train the model and record training/testing accuracy.**
7. **Plot Accuracy vs. Model Complexity: Compare train and test accuracy for different tree depths.**
8. **Visualize Best Decision Tree: Identify the depth with the highest test accuracy and plot the decision tree.**
9. **Train Naïve Bayes Classifier: Fit the Gaussian Naïve Bayes model and evaluate accuracy.**

Program Codes:

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

```
# Load dataset
iris = datasets.load_iris()
X, y = iris.data, iris.target

# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42) # Vary max_depth and record accuracies
depths = range(1, 11) # Testing max_depth from 1 to 10
train_accuracies = []
test_accuracies = []
models = {}
for depth in depths:
dt = DecisionTreeClassifier(max_depth=depth, random_state=42)
dt.fit(X_train, y_train)

# Store model for visualization later
models[depth] = dt

# Train and test accuracy
train_accuracies.append(accuracy_score(y_train, dt.predict(X_train)))
test_accuracies.append(accuracy_score(y_test, dt.predict(X_test)))

# Plot accuracy vs model complexity (max_depth)
plt.figure(figsize=(8, 6))
plt.plot(depths, train_accuracies, marker='o', linestyle='-', color='blue', label='Train Accuracy')
plt.plot(depths, test_accuracies, marker='s', linestyle='--', color='red', label='Test Accuracy')
plt.xlabel("Tree Depth (max_depth)")
plt.ylabel("Accuracy")
plt.title("Decision Tree Accuracy vs. Model Complexity")
plt.legend()
plt.grid(True)
plt.show()
best_depth = depths[test_accuracies.index(max(test_accuracies))]
best_tree = models[best_depth]

# Plot Decision Tree
plt.figure(figsize=(12, 8))
plot_tree(best_tree, filled=True, feature_names=iris.feature_names, class_names=iris.target_names)
plt.title(f"Decision Tree Visualization (max_depth={best_depth})")
plt.show()
```

Implementation/Output snap shot:



Conclusion:

Decision Trees work well with hierarchical data representation and can handle both numerical and categorical data.Naïve Bayes assumes independence between features and performs well with probabilistic models and text classification tasks.Decision Trees are prone to overfitting if depth is not controlled, while Naïve Bayes assumes conditional independence, which may not hold in some real-world cases.

Review Questions:

1. What is a Decision Tree classifier, and how does it work?

A **Decision Tree** is a classification model that splits data based on feature conditions to form a tree-like structure. It works by:

• Choosing the best feature to split at each node.
• Recursively splitting data until leaf nodes are reached.
• Assigning class labels based on majority voting at the leaf nodes.

2. Explain the Naïve Bayes algorithm and its underlying assumptions.

The **Naïve Bayes classifier** is a probabilistic model based on **Bayes' Theorem**:

P(A∣B)=P(B∣A)P(A)P(B)P(A|B) = \frac{P(B|A)

P(A)}{P(B)}P(A∣B)=P(B)P(B∣A)P(A)

Assumptions:

- All features are independent (which is rarely true in real life).
- The probability distribution of features follows a Gaussian (Normal) distribution in Gaussian Naïve Bayes.

3. Compare the working principles of Decision Tree and Naïve Bayes classifiers.

Feature	**Decision Tree**	**Naïve Bayes**
Type	Rule-based classifier	Probabilistic classifier
Working	Creates a tree-like structure using feature splits	Uses Bayes' theorem to calculate class probabilities
Interpretability	Easy to visualize and interpret	Less interpretable
Data Dependency	Works well with both numerical and categorical data	Works best with independent features

4. What are the different types of Decision Tree splitting criteria?

1. **Gini Index** – Measures impurity based on class probabilities.
2. **Entropy (Information Gain)** – Measures uncertainty in data.
3. **Chi-square** – Used for categorical variables.
4. **Reduction in Variance** – Used for regression tasks.

Github link:
https://github.com/panchaldeep1123/dwm.gits