

EXPERIMENT-9:

AIM:

Write a program to demonstrate the working of the decision tree algorithm. Use an appropriate dataset for building the decision tree and apply this knowledge to classify a newsample.

REQUIREMENTS:

1. Install Python in your PC.
2. Import scikit-learn, a popular machine language library in python.
Scikit-learn is important because it provides a comprehensive and user-friendly machine learning library in Python. It offers a wide range of algorithms and tools for tasks such as classification, regression, clustering, and model evaluation, making it accessible for researchers, practitioners, and enthusiasts to implement and experiment with machine learning algorithms.
3. NumPy is used in machine learning for efficient numerical computations and array operations.
4. Import NumPy, scikit-learn depends on this library for numerical computations.
5. Load 'Iris' Dataset.

DESCRIPTION:

The Iris dataset includes the following components:

1. Sepallength:Length of sepal
2. Sepalwidth:Width of sepal
3. Petallength:Length of petal
4. Petalwidth:Width of petal Target variable: The Iris dataset also includes a target variable or class label for each sample.

The target variable represents the species of the Iris flower.

There are three possible classes:

- A. Setosa refers to a species of Iris flowers characterized by short sepals, small petals, and vibrant colors, such as white, pink, or deep red.
- B. Versicolor is a species of Iris flower characterized by medium-sized flowers with bluish-purple or violet petals, typically found in the Iris dataset.
- C. Virginica is a species of Iris flower characterized by larger flowers with longer sepals and petals, often exhibiting shades of deep purple to burgundy colors.

PROCEDURE:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
# Load the Iris dataset
iris = load_iris()
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.2,
random_state=42)
```

```

# Create a decision tree classifier
clf = DecisionTreeClassifier()
# Train the classifier on the training data
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
# Replace with your own sample
predicted_class = clf.predict(new_sample)
# Print the predicted class
print("Predicted class:", iris.target_names[predicted_class[0]])
accuracy = accuracy_score(y_test, y_pred)
sepalength=float(input("ENTER SEPAL LENGTH OF FLOWER:"))
sepalwidth=float(input("ENTER SEPAL WIDTH OF FLOWER:"))
petallength=float(input("ENTER PETAL LENGTH OF FLOWER:"))
petalwidth=float(input("ENTER PETAL WIDTH OF FLOWER:"))
print("Accuracy:", accuracy)
new_sample = [[sepalength,sepalwidth,petallength,petalwidth]]
predicted_class = clf.predict(new_sample)
predicted_species = iris.target_names[predicted_class]
print("Predicted species:", predicted_species)

```

OUTPUT:

```

ENTER SEPAL LENGTH OF FLOWER:4
ENTER SEPAL WIDTH OF FLOWER:5
ENTER PETAL LENGTH OF FLOWER:65
ENTER PETAL WIDTH OF FLOWER:5
Accuracy: 1.0
Predicted species: ['virginica']

```

RESULT:

The Result will be seen as Predicted class label for the new sample printed as the output. The predicted class label corresponds to one of the three classes in the iris dataset:

‘setosa’, ‘versicolor’, or ‘virginica’. The accuracy is also included of the decision tree classifier on the testing set, indicating how well the model performs on unseen data.

