Experiment 4

**Student Name**: Rahul Saxena

**UID:** 24MCI10204

**Branch:** MCA (AI-ML)

**Section/Group:** MAM - 3(B)

**Semester:** II

**Subject Name:** Machine Learning Lab [24CAP-672]

---

**Aim:** Decision Tree Implementation Using Pandas and Graphviz in Jupyter Notebook

**Steps:**

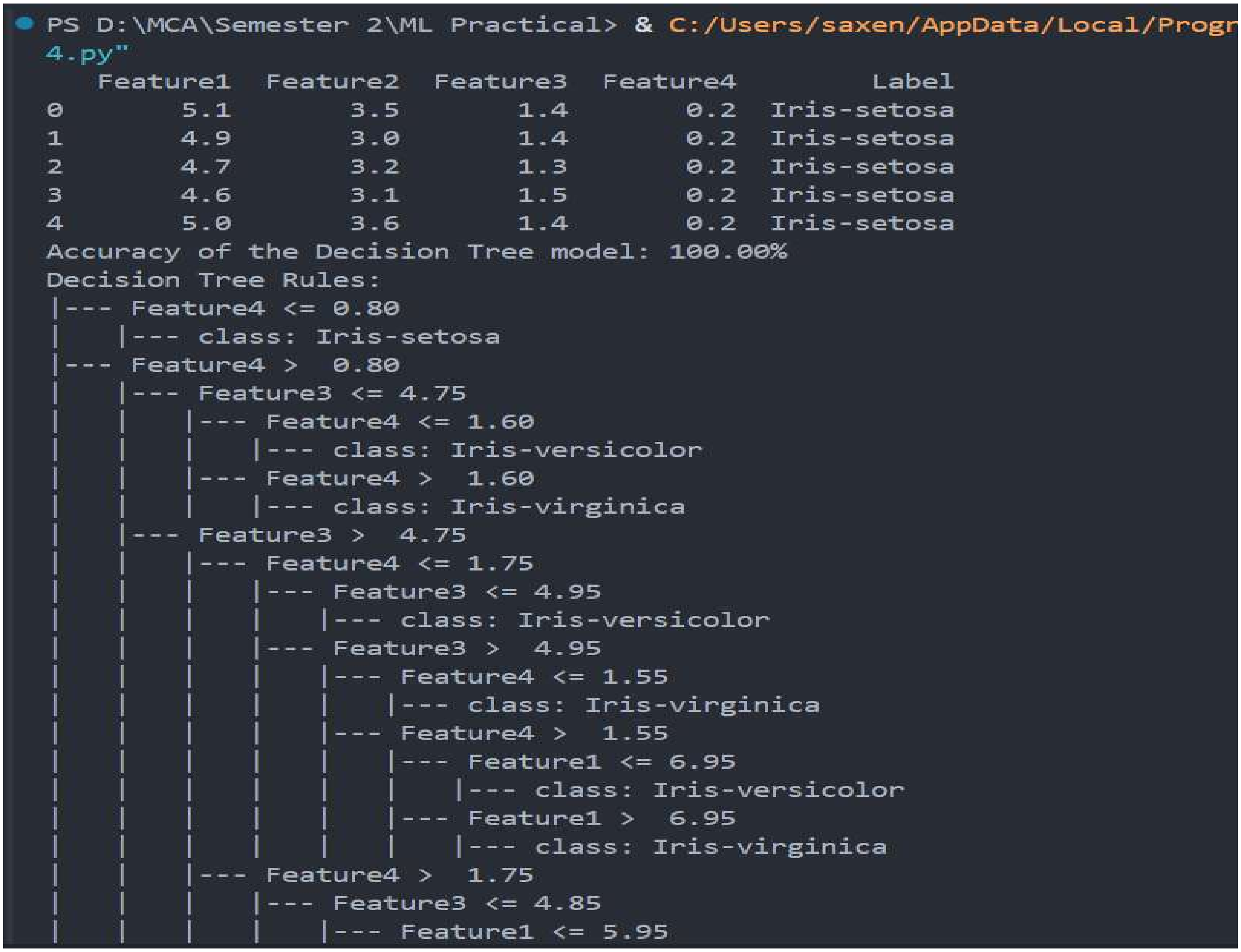
1. Import Necessary Libraries
   - Import pandas for handling datasets.
   - Import DecisionTreeClassifier, train_test_split, accuracy_score, export_graphviz, and export_text from sklearn.
   - Import graphviz to visualize the decision tree.
2. Prepare the Dataset
   - Create a dictionary containing features (Feature1, Feature2) and a target label (Label).
   - Convert the dictionary into a Pandas DataFrame.
   - Split the dataset into independent variables X and dependent variable y.
3. Split Data into Training and Testing Sets
   - Use train_test_split with a 70-30 ratio to divide the dataset into training and testing sets.
4. Initialize and Train the Decision Tree Model
   - Create an instance of DecisionTreeClassifier using the Gini impurity criterion.
   - Train the classifier using fit() with training data.
5. Make Predictions and Evaluate Accuracy
   - Predict labels on the test data using predict().
   - Calculate model accuracy using accuracy_score().
   - Print the accuracy percentage.

6. Visualize the Decision Tree
   - Generate a Graphviz representation of the trained decision tree using export_graphviz().
   - Render and display the decision tree as a PNG file.
7. Extract and Display Decision Rules
   - Extract human-readable decision rules using export_text().
   - Print the extracted rules for better understanding of the model's decision process.

## Input Code:

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier, export_text
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import export_graphviz
from graphviz import Source
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/iris.csv"
columns = ['Feature1', 'Feature2', 'Feature3', 'Feature4', 'Label']
df = pd.read_csv(url, names=columns)
print(df.head())
X = df.iloc[:, :-1]
y = df.iloc[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
clf = DecisionTreeClassifier(criterion='gini')
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy of the Decision Tree model: {accuracy * 100:.2f}%")
dot_data = export_graphviz(clf, out_file=None, feature_names=X.columns, class_names=clf.classes_,
                           filled=True, rounded=True, special_characters=True)
graph = Source(dot_data)
graph.render("decision_tree", format="png", view=True)
print("Decision Tree Rules:")
tree_rules = export_text(clf, feature_names=list(X.columns))
print(tree_rules)
```

## Output:



```
PS D:\MCA\Semester 2\ML Practical> & C:/Users/saxen/AppData/Local/Progr
4.py"
   Feature1  Feature2  Feature3  Feature4            Label
0       5.1       3.5       1.4       0.2  Iris-setosa
1       4.9       3.0       1.4       0.2  Iris-setosa
2       4.7       3.2       1.3       0.2  Iris-setosa
3       4.6       3.1       1.5       0.2  Iris-setosa
4       5.0       3.6       1.4       0.2  Iris-setosa
Accuracy of the Decision Tree model: 100.00%
Decision Tree Rules:
|--- Feature4 <= 0.80
|   |--- class: Iris-setosa
|--- Feature4 >  0.80
|   |--- Feature3 <= 4.75
|   |   |--- Feature4 <= 1.60
|   |   |   |--- class: Iris-versicolor
|   |   |--- Feature4 >  1.60
|   |   |   |--- class: Iris-virginica
|   |--- Feature3 >  4.75
|   |   |--- Feature4 <= 1.75
|   |   |   |--- Feature3 <= 4.95
|   |   |   |   |--- class: Iris-versicolor
|   |   |   |--- Feature3 >  4.95
|   |   |   |   |--- Feature4 <= 1.55
|   |   |   |   |   |--- class: Iris-virginica
|   |   |   |   |--- Feature4 >  1.55
|   |   |   |   |   |--- Feature1 <= 6.95
|   |   |   |   |   |   |--- class: Iris-versicolor
|   |   |   |   |   |--- Feature1 >  6.95
|   |   |   |   |   |   |--- class: Iris-virginica
|   |   |--- Feature4 >  1.75
|   |   |   |--- Feature3 <= 4.85
|   |   |   |   |--- Feature1 <= 5.95
```

## Learning Outcome:

- Understanding how to create and manipulate a dataset using Pandas.
- Learning how to train a Decision Tree classifier in scikit-learn.
- Gaining knowledge on splitting data for training and testing.
- Evaluating model accuracy and performance.
- Visualizing a decision tree using Graphviz.
- Extracting and interpreting decision rules from the trained model.