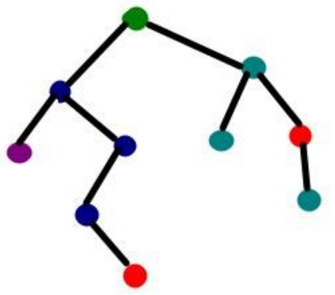
REG NO.220701055

EX.NO : 10 **DATE : 08.11.24**

IMPLEMENTATION OF DECISION TREE CLASSIFICATION TECHNIQUES

Decision Tree is one of the most powerful and popular algorithm. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.



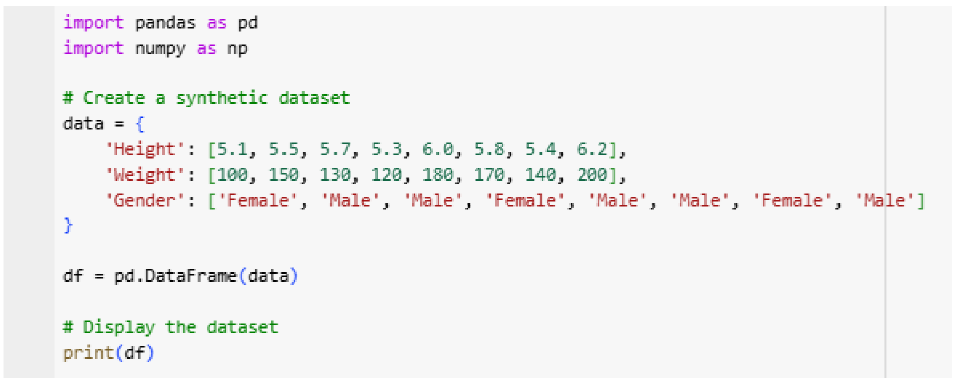
AIM:

To implement a decision tree classification technique for gender classification using python.

EXPLANATION:

- Import tree from sklearn.
- Call the function DecisionTreeClassifier() from tree
- Assign values for X and Y.
- Call the function predict for Predicting on the basis of given random values for each given feature.
- Display the output.

CODE:



```
import pandas as pd
import numpy as np

# Create a synthetic dataset
data = {
    'Height': [5.1, 5.5, 5.7, 5.3, 6.0, 5.8, 5.4, 6.2],
    'Weight': [100, 150, 130, 120, 180, 170, 140, 200],
    'Gender': ['Female', 'Male', 'Male', 'Female', 'Male', 'Male', 'Female', 'Male']
}

df = pd.DataFrame(data)

# Display the dataset
print(df)
```

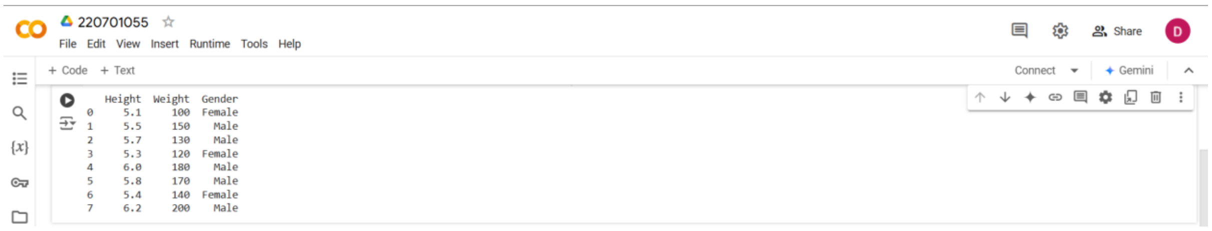
```
[ ] from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import LabelEncoder

    # Encode the target variable (Gender)
    label_encoder = LabelEncoder()
    df['Gender'] = label_encoder.fit_transform(df['Gender'])  # Female: 0, Male: 1

    # Split the dataset into features and target variable
    X = df[['Height', 'Weight']]
    y = df['Gender']

    # Split the data 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)
```

```
[ ] from sklearn.tree import DecisionTreeClassifier

    # Create a Decision Tree Classifier
    classifier = DecisionTreeClassifier()

    # Train the classifier on the training data
    classifier.fit(X_train, y_train)
```

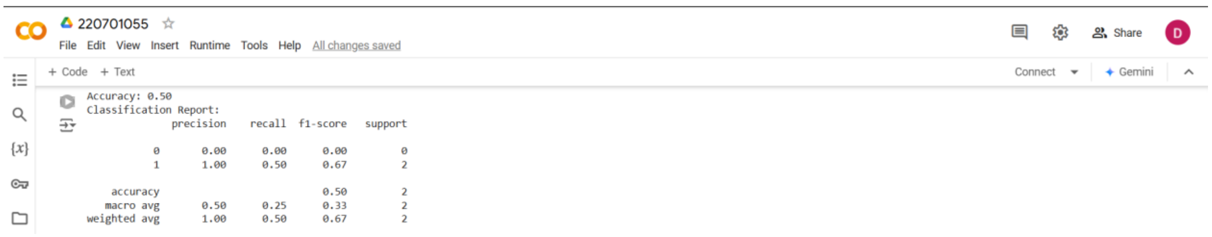
```
[ ] from sklearn.metrics import accuracy_score, classification_report

    # Make predictions on the test set
    y_pred = classifier.predict(X_test)

    # Evaluate the model's performance
    accuracy = accuracy_score(y_test, y_pred)
    report = classification_report(y_test, y_pred)

    print(f"Accuracy: {accuracy:.2f}")
    print("Classification Report:")
    print(report)
```

```
[ ] from sklearn.tree import export_graphviz
    import graphviz

    # Export as dot file
    dot_data = export_graphviz(classifier, out_file=None,
                               feature_names=['Height', 'Weight'],
                               class_names=label_encoder.classes_,
                               filled=True, rounded=True,
                               special_characters=True)

    # Draw graph
    graph = graphviz.Source(dot_data)
    graph.render("gender_classification_tree")
    graph.view()
```

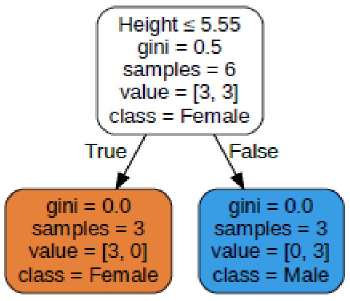
OUTPUT:



```
   Height  Weight  Gender
0     5.1     100  Female
1     5.5     150    Male
2     5.7     130    Male
3     5.3     120  Female
4     6.0     180    Male
5     5.8     170    Male
6     5.4     140  Female
7     6.2     200    Male
```



```
Accuracy: 0.50
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       1.00      0.50      0.67         2

    accuracy                           0.50         2
   macro avg       0.50      0.25      0.33         2
weighted avg       1.00      0.50      0.67         2
```



RESULT: Thus the implementation of decision tree for gender classification is executed successfully.