

EX.NO :9

DATE :

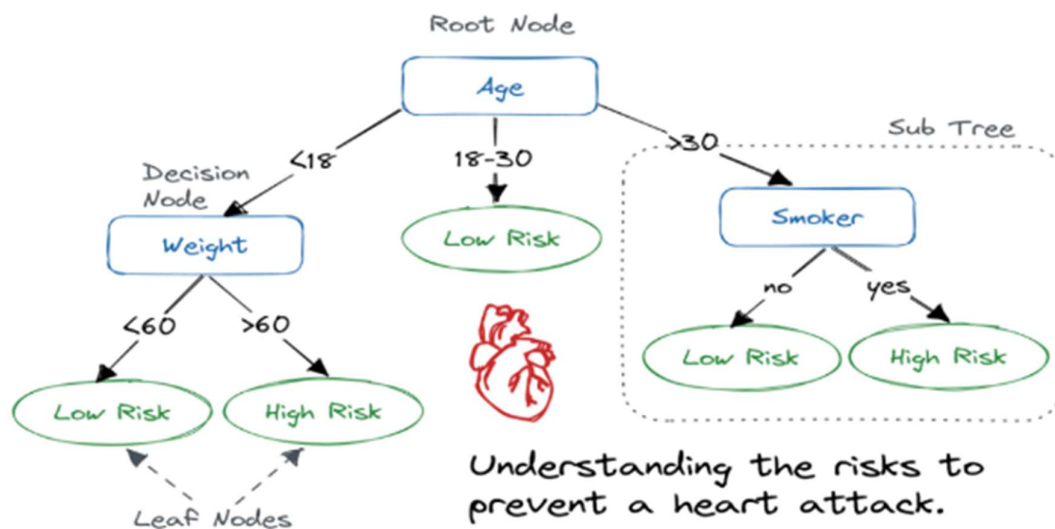
EX NO:220701024

DECISION TREE CLASSIFICATION

Classification is a two-step process; a learning step and a prediction step. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response to given data. A Decision tree is one of the easiest and most popular classification algorithms used to understand and interpret data. It can be utilized for both classification and regression problems.

The Decision Tree Algorithm

A decision tree is a flowchart-like tree structure where an internal node represents a feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in a recursive manner called recursive partitioning. This flowchart-like structure helps you in decision-making. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.



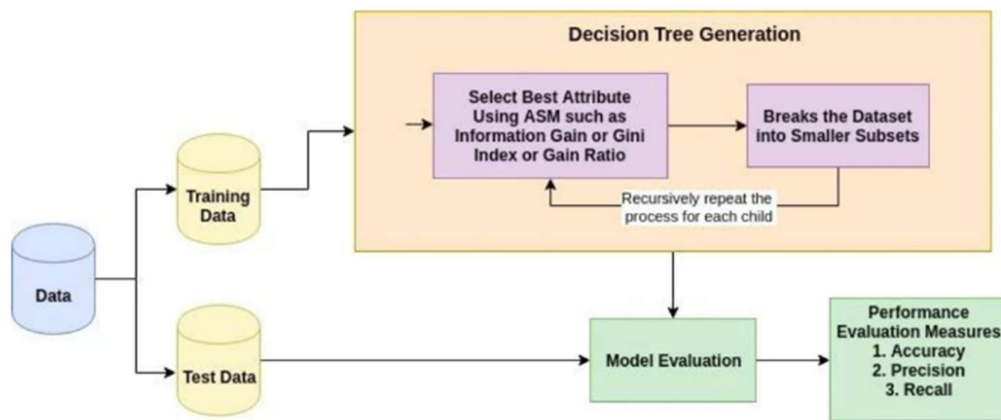
A decision tree is a white box type of ML algorithm. It shares internal decision-making logic, which is not available in the black box type of algorithms such as with a neural network. Its training time is faster compared to the neural network algorithm. The time complexity of decision trees is a function of the number of records and attributes in the given data. The decision tree is a distribution-free or non-parametric method that does

not depend upon probability distribution assumptions. Decision trees can handle high-dimensional data with good accuracy.

How Does the Decision Tree Algorithm Work?

The basic idea behind any decision tree algorithm is as follows:

1. Select the best attribute using Attribute Selection Measures (ASM) to split the records.
2. Make that attribute a decision node and breaks the dataset into smaller subsets.
3. Start tree building by repeating this process recursively for each child until one of the conditions will match:
 - All the tuples belong to the same attribute value.
 - There are no more remaining attributes.
 - There are no more instances.



CODE:

```

import pandas as pd
from sklearn.tree import DecisionTreeClassifier

data = {
    'Height': [152, 155, 172, 185, 167, 180, 157, 180, 164, 177],
    'Weight': [45, 57, 72, 85, 68, 78, 22, 90, 66, 88],
    'Gender': ['Female', 'Female', 'Male', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male']
}

df = pd.DataFrame(data)
X = df[['Height', 'Weight']]
Y = df['Gender']

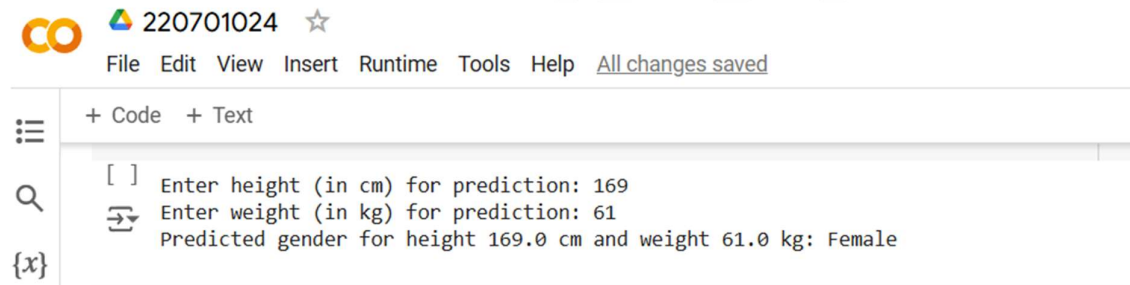
classifier = DecisionTreeClassifier()
classifier.fit(X, Y)

height = float(input("Enter height (in cm) for prediction: "))
weight = float(input("Enter weight (in kg) for prediction: "))
random_values = pd.DataFrame([[height, weight]], columns=['Height', 'Weight'])
predicted_gender = classifier.predict(random_values)

print(f"Predicted gender for height {height} cm and weight {weight} kg: {predicted_gender[0]}")

```

OUTPUT:



RESULT:

The above DECISION TREE CLASSIFICATION program is successfully implemented.