

Task-03

Build a decision tree classifier to predict whether a customer will purchase a product or service based on their demographic and behavioral data. Use a dataset such as the Bank Marketing dataset from the UCI Machine Learning Repository.

In [1]:

```

# Load the required Libraries
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.preprocessing import LabelEncoder

# Load the data
df = pd.read_csv('E:\\prodigy\\bank.csv')
df

# Preprocessing
# Drop unnecessary columns
df = df.drop(['day', 'month'], axis=1)

# Encode categorical variables
label_encoder = LabelEncoder()
categorical_cols = ['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact']
for col in categorical_cols:
    df[col] = label_encoder.fit_transform(df[col])

# Define features and target variable
X = df.drop('deposit', axis=1)
y = df['deposit']

# 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)

# Build the Decision Tree model
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)

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

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

# Print the results
print(f"Accuracy: {accuracy}")
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", classification_rep)

```

Accuracy: 0.7429467084639498

Confusion Matrix:

```
[[883 283]
```

```
[291 776]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.75	0.76	0.75	1166
1	0.73	0.73	0.73	1067
accuracy			0.74	2233
macro avg	0.74	0.74	0.74	2233
weighted avg	0.74	0.74	0.74	2233

In [2]:

```
df
```

Out[2]:

	age	job	marital	education	default	balance	housing	loan	contact	duration	campaign
0	59	0	1	1	0	2343	1	0	2	1042	
1	56	0	1	1	0	45	0	0	2	1467	
2	41	9	1	1	0	1270	1	0	2	1389	
3	55	7	1	1	0	2476	1	0	2	579	
4	54	0	1	2	0	184	0	0	2	673	
...
11157	33	1	2	0	0	1	1	0	0	257	
11158	39	7	1	1	0	733	0	0	2	83	
11159	32	9	2	1	0	29	0	0	0	156	
11160	43	9	1	1	0	0	0	1	0	9	
11161	34	9	1	1	0	0	0	0	0	628	

11162 rows × 15 columns

In []: