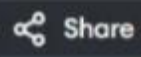


TASK 03

- 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.

[Bank Marketing Dataset](#)

main.py



Share

Run

```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.preprocessing import LabelEncoder
4 from sklearn.tree import DecisionTreeClassifier, plot_tree
5 from sklearn.metrics import accuracy_score, classification_report,
  confusion_matrix
6 import matplotlib.pyplot as plt
7 import seaborn as sns
8
9 #1. tnad dataset
10 ur1 = "https://gist.githubusercontent.com/dim4o/c4a67e5300faaf_cut14df20352601fasf
  /res/bank.csv"
11 df = pd.read_csv(ur1, sep=";")
12
13
14 #2. Encode Categorical Variables
15 label_encoders = {}
16 for col in df.select_dtypes(include = 'object').columns:
17     le = LabelEncoder()
18     df[col] = le.fit_transform(df[col])
19     label_encoders[col] = le
20
21 #3. Prepare features (demographic + behavioral) and target
22 x = df.drop('y', axis=1)      # all columns except the target
23 y = df['y']
24
25 #4. Split into training and testing data
26 x_train, x_test, y_train, y_test = train_test_split(
27     x, y, test_size=0.2, random_state=42
28 )
```

main.py



Share

Run

```

1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3 from sklearn.preprocessing import LabelEncoder
4 from sklearn.tree import DecisionTreeClassifier, plot_tree
5 from sklearn.metrics import accuracy_score, classification_report,
   confusion_matrix
6 import matplotlib.pyplot as plt
7 import seaborn as sns
8
9 #1. load dataset
10 ur1 = "https://gist.githubusercontent.com/dim4o/c4a67e5300faaf_cut14df20352601fasf
   /res/bank.csv"
11 df = pd.read_csv(ur1, sep=";")
12
13
14 #2. Encode Categorical Variables
15 label_encoders = {}
16 for col in df.select_dtypes(include = 'object').columns:
17     le = LabelEncoder()
18     df[col] = le.fit_transform(df[col])
19     label_encoders[col] = le
20
21 #3. Prepare features (demographic + behavioral) and target
22 x = df.drop('y', axis=1)      # all columns except the target
23 y = df['y']
24
25 #4. Split into training and testing data
26 x_train, x_test, y_train, y_test = train_test_split(
27     x, y, test_size=0.2, random_state=42
28 )

```

```
29
30 #5. Train Decision Tree Classifier
31 clf = DecisionTreeClassifier(max_depth=4, random_states=42)
32 clf.fit(x_train, y_train)
33
34 #6. Evaluate the model
35 y_pred = clf.predict(x_test)
36 accuracy = accuracy_score(y_test, y_pred)
37 print(" Decision tree classifier to predict Customer Purchase\n")
```

Confusion Matrix — Customer Purchase Prediction

