

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv(r'bank-additional-full.csv', sep=';')

# Display the first few rows of the dataset
print("First few rows of the dataset:")
print(data.head())

# Check for missing values
print("\nMissing values in each column:")
print(data.isnull().sum())

# Convert categorical columns to numerical using One-Hot Encoding
data = pd.get_dummies(data)

# Features (X) and Target (y)
X = data.drop('y_yes', axis=1) # Drop 'y_yes' to avoid target column
in features
y = data['y_yes'] # Target variable (whether the customer subscribed)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

# Initialize the Decision Tree Classifier
clf = DecisionTreeClassifier(random_state=42)

# Train the classifier on the training set
clf.fit(X_train, y_train)

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

# Evaluate the model
print("\nModel Evaluation:")
print(f"Accuracy: {accuracy_score(y_test, y_pred)}")
print(f"Confusion Matrix:\n{confusion_matrix(y_test, y_pred)}")
print(f"Classification Report:\n{classification_report(y_test,
y_pred)}")

# Plot the Decision Tree
plt.figure(figsize=(12, 8))
plot_tree(clf,
          filled=True,

```

```

        feature_names=X.columns.tolist(), # Convert columns to list
        class_names=['No', 'Yes'],
        rounded=True,
        fontsize=10)
plt.show()

```

First few rows of the dataset:

	age	job	marital	education	default	housing	loan
contact \							
0	56	housemaid	married	basic.4y	no	no	no
telephone							
1	57	services	married	high.school	unknown	no	no
telephone							
2	37	services	married	high.school	no	yes	no
telephone							
3	40	admin.	married	basic.6y	no	no	no
telephone							
4	56	services	married	high.school	no	no	yes
telephone							

	month	day_of_week	...	campaign	pdays	previous	poutcome
emp.var.rate \							
0	may	mon	...	1	999	0	nonexistent
1.1							
1	may	mon	...	1	999	0	nonexistent
1.1							
2	may	mon	...	1	999	0	nonexistent
1.1							
3	may	mon	...	1	999	0	nonexistent
1.1							
4	may	mon	...	1	999	0	nonexistent
1.1							

	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
0	93.994	-36.4	4.857	5191.0	no
1	93.994	-36.4	4.857	5191.0	no
2	93.994	-36.4	4.857	5191.0	no
3	93.994	-36.4	4.857	5191.0	no
4	93.994	-36.4	4.857	5191.0	no

[5 rows x 21 columns]

Missing values in each column:

age	0
job	0
marital	0
education	0
default	0
housing	0
loan	0

```
contact      0
month        0
day_of_week  0
duration     0
campaign     0
pdays       0
previous     0
poutcome     0
emp.var.rate 0
cons.price.idx 0
cons.conf.idx 0
euribor3m    0
nr.employed  0
y            0
dtype: int64
```

Model Evaluation:

Accuracy: 1.0

Confusion Matrix:

```
[[10968    0]
 [    0  1389]]
```

Classification Report:

	precision	recall	f1-score	support
False	1.00	1.00	1.00	10968
True	1.00	1.00	1.00	1389
accuracy			1.00	12357
macro avg	1.00	1.00	1.00	12357
weighted avg	1.00	1.00	1.00	12357

