

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.preprocessing import LabelEncoder
```

```
df=pd.read_csv("bank-additional-full.csv",sep=";")
```

```
df.head()
```



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

5 rows × 21 columns

```
# Preprocessing the data
# Convert categorical variables using Label Encoding
label_encoders = {}
for column in df.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
    label_encoders[column] = le
```

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

```
# Split the dataset 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)
```

```
# Build the decision tree classifier
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)
```



```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

```
# Predict on the test set
y_pred = clf.predict(X_test)
```

```
# Evaluate the model
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

⇒ Confusion Matrix:

```
[[10275  693]
 [ 675  714]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.94	0.94	0.94	10968
1	0.51	0.51	0.51	1389
accuracy			0.89	12357
macro avg	0.72	0.73	0.72	12357
weighted avg	0.89	0.89	0.89	12357

Start coding or [generate](#) with AI.