

```
import pandas as pd
```

```
# Load the CSV file from your local directory
df = pd.read_csv('bank-full.csv', sep=';')

# Preview the data
print(df.head())

  age         job marital education default balance housing loan \
0  58 management married   tertiary    no     2143    yes   no
1  44 technician single secondary    no      29    yes   no
2  33 entrepreneur married secondary    no      2    yes  yes
3  47 blue-collar married unknown    no    1506    yes   no
4  33        unknown single unknown    no      1    no   no

  contact day month duration campaign pdays previous poutcome y
0 unknown  5   may       261        1     -1      0  unknown  no
1 unknown  5   may       151        1     -1      0  unknown  no
2 unknown  5   may        76        1     -1      0  unknown  no
3 unknown  5   may       92        1     -1      0  unknown  no
4 unknown  5   may      198        1     -1      0  unknown  no
```

```
# Check for missing values
print(df.isnull().sum())
```

```
age      0
job      0
marital  0
education 0
default  0
balance  0
housing  0
loan      0
contact  0
day      0
month    0
duration 0
campaign 0
pdays    0
previous 0
poutcome 0
y        0
dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder

df_encoded = df.copy()
label_encoders = {}

# Encode all object (categorical) columns
for col in df_encoded.select_dtypes(include='object').columns:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df_encoded[col])
    label_encoders[col] = le

X = df_encoded.drop('y', axis=1) # Features
y = df_encoded['y']             # Target (0 = no, 1 = yes)
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier(criterion='entropy', max_depth=5, random_state=42)
clf.fit(X_train, y_train)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
y_pred = clf.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

Accuracy: 0.8848833351763795
Confusion Matrix:
[[7749  203]
 [ 838  253]]
Classification Report:
precision    recall    f1-score   support
          0       0.90      0.97      0.94      7952
          1       0.55      0.23      0.33     1091

   accuracy                           0.88      9043
  macro avg       0.73      0.60      0.63      9043
weighted avg       0.86      0.88      0.86      9043
```

```
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt

plt.figure(figsize=(20,10))
plot_tree(clf, feature_names=X.columns, class_names=['No', 'Yes'], filled=True)
plt.show()
```

