PRODIGY INFOTECH INTERNSHIP

TASK 3: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.

DONE BY: Krupa J Shetty

Email:krupajshetty@gmail.com

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier,plot_tree
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import LabelEncoder
```

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

	age	job	marital	education	default	balance	housing	loan	contact
0	58	management	married	tertiary	no	2143	yes	no	unknowr
1	44	technician	single	secondary	no	29	yes	no	unknowr
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknowr
3	47	blue-collar	married	unknown	no	1506	yes	no	unknowr
4	33	unknown	single	unknown	no	1	no	no	unknowr
...	...	...	...	...	...	...	...	...	..
45206	51	technician	married	tertiary	no	825	no	no	cellulaı
45207	71	retired	divorced	primary	no	1729	no	no	cellulaı
45208	72	retired	married	secondary	no	5715	no	no	cellulaı
45209	57	blue-collar	married	secondary	no	668	no	no	telephone
45210	37	entrepreneur	married	secondary	no	2971	no	no	cellulaı

45211 rows × 17 columns

```
df.head(10)
```

	age	job	marital	education	default	balance	housing	loan	contact	day
0	58	management	married	tertiary	no	2143	yes	no	unknown	5
1	44	technician	single	secondary	no	29	yes	no	unknown	5
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5
4	33	unknown	single	unknown	no	1	no	no	unknown	5
5	35	management	married	tertiary	no	231	yes	no	unknown	5
6	28	management	single	tertiary	no	447	yes	yes	unknown	5
7	42	entrepreneur	divorced	tertiary	yes	2	yes	no	unknown	5
8	58	retired	married	primary	no	121	yes	no	unknown	5
9	43	technician	single	secondary	no	593	yes	no	unknown	5

```
df.isna().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
```

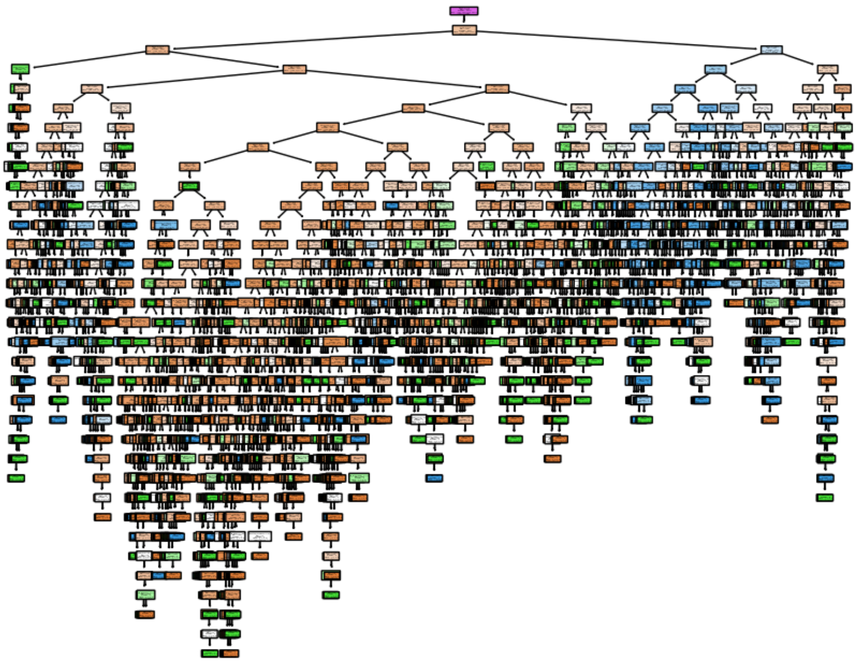
```
df.dropna(inplace=True)
df_1=df.drop_duplicates()
df_1.info
```

```
<bound method DataFrame.info of        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
...    ...           ...       ...        ...     ...      ...     ...  ...
45206   51    technician   married   tertiary      no      825      no   no
45207   71       retired  divorced    primary      no     1729      no   no
45208   72       retired   married  secondary      no     5715      no   no
45209   57   blue-collar   married  secondary      no      668      no   no
45210   37  entrepreneur   married  secondary      no     2971      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
...          ...  ...   ...       ...       ...    ...       ...      ...  ...
45206   cellular   17   nov       977         3     -1         0  unknown  yes
45207   cellular   17   nov       456         2     -1         0  unknown  yes
45208   cellular   17   nov      1127         5    184         3  success  yes
45209  telephone   17   nov       508         4     -1         0  unknown   no
45210   cellular   17   nov       361         2    188        11    other   no

[45211 rows x 17 columns]>
```

```
#Preprocess the data
X=df.drop('poutcome', axis=1)
y=df['poutcome']
X=pd.get_dummies(X)
X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.2, random_state=42)

#Create the classifier
clf=DecisionTreeClassifier()

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

#Make predictions
y_pred=clf.predict(X_test)

#Calculate accuracy
accuracy=accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9183899148512662
```

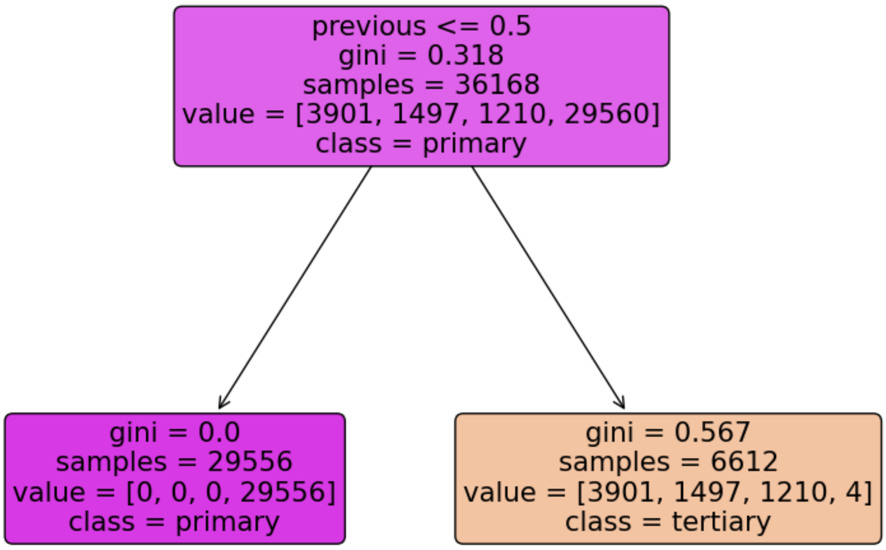
```
#Visualize the decision tree
plt.figure(figsize=(10,8))
plot_tree(clf,feature_names=list(X.columns), class_names=df['education'].unique().tolist(),filled=True,rounded=True)
plt.show()
```



```
#Create the classsifier with pruning enabled
clf=DecisionTreeClassifier(ccp_alpha=0.01)
#Train the classifier
clf.fit(X_train,y_train)
#Make predictions
y_pred=clf.predict(X_test)
#Calculate accuracy
accuracy=accuracy_score(y_test,y_pred)
print("Accuracy:",accuracy)
```

```
Accuracy: 0.928674112573261
```

```
#Visualize the pruned decision tree
plt.figure(figsize=(10,8))
plot_tree(clf,feature_names=list(X.columns),class_names=df['education'].unique().tolist(),filled=True,rounded=True)
plt.show()
```


previous <= 0.5
gini = 0.318
samples = 36168
value = [3901, 1497, 1210, 29560]
class = primary
gini = 0.0
samples = 29556
value = [0, 0, 0, 29556]
class = primary
gini = 0.567
samples = 6612
value = [3901, 1497, 1210, 4]
class = tertiary