

zfgn2bf8w

August 2, 2023

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[2]: from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
[4]: df=pd.read_csv("/content/drive/MyDrive/mydatasets/C10_loan1.csv")
df
```

```
[4]:
```

	Home Owner	Marital Status	Annual Income	Defaulted	Borrower
0	Yes	Single	125	No	
1	No	Married	100	No	
2	No	Single	70	No	
3	Yes	Married	120	No	
4	No	Divorced	95	Yes	
5	No	Married	60	No	
6	Yes	Divorced	220	No	
7	No	Single	85	Yes	
8	No	Married	75	No	
9	No	Single	90	Yes	

```
[6]: df['Defaulted Borrower'].value_counts()
```

```
[6]:
```

No	7
Yes	3

Name: Defaulted Borrower, dtype: int64

```
[7]: x=df[['Annual Income','Annual Income']]
y=df['Defaulted Borrower']
```

```
[8]: g1={"Defaulted Borrower":{"Yes":1,"No":2}}
df=df.replace(g1)
df
```

```
[8]:
```

	Home Owner	Marital Status	Annual Income	Defaulted	Borrower
0	Yes	Single	125		No
1	No	Married	100		No
2	No	Single	70		No
3	Yes	Married	120		No
4	No	Divorced	95		Yes
5	No	Married	60		No
6	Yes	Divorced	220		No
7	No	Single	85		Yes
8	No	Married	75		No
9	No	Single	90		Yes

```
[9]: from sklearn.model_selection import train_test_split
```

```
[10]: x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.70)
```

```
[11]: from sklearn.ensemble import RandomForestClassifier
```

```
[12]: rfc=RandomForestClassifier()
      rfc.fit(x_train,y_train)
```

```
[12]: RandomForestClassifier()
```

```
[13]: parameters={'max_depth': [1,2,3,4,5],
                  'min_samples_leaf': [5,10,15,20,25],
                  'n_estimators': [10,20,30,40,50]
                }
```

```
[14]: from sklearn.model_selection import GridSearchCV
      grid_search_
      ↪=GridSearchCV(estimator=rfc,param_grid=parameters,cv=2,scoring="accuracy")
      grid_search.fit(x_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/model_selection/_split.py:700:
UserWarning: The least populated class in y has only 1 members, which is less
than n_splits=2.
  warnings.warn(
```

```
[14]: GridSearchCV(cv=2, estimator=RandomForestClassifier(),
                  param_grid={'max_depth': [1, 2, 3, 4, 5],
                              'min_samples_leaf': [5, 10, 15, 20, 25],
                              'n_estimators': [10, 20, 30, 40, 50]},
                  scoring='accuracy')
```

```
[15]: grid_search.best_score_
```

```
[15]: 0.875
```

```
[16]: rfc_best=grid_search.best_estimator_
```

```
[17]: from sklearn.tree import plot_tree  
  
plt.figure(figsize=(80,40))  
plot_tree(rfc_best.estimators_[5],feature_names=x.  
↪columns,class_names=['Yes','No'],filled=True)
```

```
[17]: [Text(0.5, 0.5, 'gini = 0.245\nsamples = 6\nvalue = [6, 1]\nclass = Yes')]
```

gini = 0.245
samples = 6
value = [6, 1]
class = Yes