Decision tree - Bank Loan

In [1]:

import pandas as pd  
dataset = pd.read\_excel('Dataset/Bank\_Personal\_Loan\_Modelling.xlsx',sheet\_name='Data')

In [2]:

dataset.columns

Out[2]:

Index(['ID', 'Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg',  
 'Education', 'Mortgage', 'Personal Loan', 'Securities Account',  
 'CD Account', 'Online', 'CreditCard'],  
 dtype='object')

In [3]:

dataset1=dataset.drop(['ID', 'ZIP Code'], axis=1)  
dataset1.columns

Out[3]:

Index(['Age', 'Experience', 'Income', 'Family', 'CCAvg', 'Education',  
 'Mortgage', 'Personal Loan', 'Securities Account', 'CD Account',  
 'Online', 'CreditCard'],  
 dtype='object')

In [4]:

dataset2 = dataset1.dropna()  
dataset3 = dataset2.drop\_duplicates()

In [5]:

import numpy as np  
dataset3["CCAvg"] = np.round(dataset3["CCAvg"])

C:\Users\Joy\anaconda3\lib\site-packages\ipykernel\_launcher.py:2: SettingWithCopyWarning:   
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy

In [6]:

from sklearn.ensemble import RandomForestClassifier  
  
rf\_model = RandomForestClassifier(n\_estimators=1000,max\_features=2,oob\_score=True)  
  
features = ['Age','Experience','Income','Family','CCAvg','Education','Mortgage','Securities Account','CD Account','Online','CreditCard']  
  
rf\_model.fit(X=dataset3[features],y=dataset3["Personal Loan"])

Out[6]:

RandomForestClassifier(max\_features=2, n\_estimators=1000, oob\_score=True)

In [7]:

print("OOB Accuracy score: ", rf\_model.oob\_score\_)

OOB Accuracy score: 0.986164026468819

#### Sorting out the Important variables[¶](#Sorting-out-the-Important-variables)

In [8]:

for feature,imp in zip(features,rf\_model.feature\_importances\_):  
 print(feature, " \t:", imp)

Age : 0.05099975756395016  
Experience : 0.05118737974909637  
Income : 0.36065658614213125  
Family : 0.10389075637715041  
CCAvg : 0.1385979019421533  
Education : 0.1651343128333275  
Mortgage : 0.047771058406858885  
Securities Account : 0.006374218620408789  
CD Account : 0.05468560914866897  
Online : 0.009456695136869224  
CreditCard : 0.011245724079385079

#### The variables with higher importance value :[¶](#X71ed5e1789898f2be011c6fb2c0bf3b4bd1abfb)

##### **\_\_**1. income[¶](#Xff4cd1393653dffb2cd7e9ed94c627ced39b77d)

##### **\_\_**2. CCAvg[¶](#X3f9b00bd596962688a0a07661949207bef97520)

##### **\_\_**3. Education[¶](#Xf998006ece6473ece98a81aeff051f1b77f122c)

#### Taking these variables to fit in the model[¶](#X63fda7e59b3fc1edd25c68010c49446ce1d8e93)

In [9]:

from sklearn import tree  
  
tree\_model = tree.DecisionTreeClassifier(max\_depth=6, max\_leaf\_nodes=12)  
  
cl\_data = pd.DataFrame([dataset3["Education"],dataset3["CCAvg"],dataset3["Income"]]).T  
  
tree\_model.fit(X=cl\_data,y=dataset3["Personal Loan"])

Out[9]:

DecisionTreeClassifier(max\_depth=6, max\_leaf\_nodes=12)

In [10]:

with open("Dtree\_bank\_loan.dot","w") as f:  
 f=tree.export\_graphviz(tree\_model,feature\_names=["Education","CCAvg","Income"],out\_file=f)

### Rules for Loan Sanctions[¶](#Rules-for-Loan-Sanctions)

##### 1. Person having income more than 106.5, with CCAvg more than 2.5 have a high chance of getting the loan.[¶](#X83d61e6bad5dee9ba1a5057d60e055e6226f902)

##### 2. Person having income more than 88.5, with Education level 2 or more have a high chance of getting the loan.[¶](#Xb4265af2ddf6b10a6c87bd9dca6ee1f1a80f518)

##### 3. Person having income more than 92.5, with Education level 3 or more have a high chance of getting the loan.[¶](#X82ec7ae55acb6a3dfe1329801173214263f4de0)

##### 4. Person having income more than 116.5, with Education level 2 or more have a high chance of getting the loan.[¶](#X80a9a062e1e490acc81edd02c8c286d074b4c13)

##### 5. Person having income more than 116.5, with CCAvg more than 3.5 have a high chance of getting the loan.[¶](#X6437e8b34b53fc84b86ef9f6db0f722a8675f84)

In [ ]: