**Project 1: Build Decision Tree(DV-"Survived",IDV-"Age,Gender and Fare") and Prediction**

In [5]:

**import** **pandas** **as** **pd**

In [10]:

dataset=pd.read\_excel(r'E:\Vivek\Data Science\Artificial Intelligence & Machine Learning\Notes\Day\_24\Dataset\Bank\_Personal\_Loan\_Modelling.xlsx', sheet\_name=1)

In [11]:

dataset.columns

Out[11]:

Index(['ID', 'Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg',

'Education', 'Mortgage', 'Personal Loan', 'Securities Account',

'CD Account', 'Online', 'CreditCard'],

dtype='object')

In [12]:

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

In [13]:

dataset2=dataset1.dropna()

In [14]:

dataset3=dataset2.drop\_duplicates()

In [15]:

**from** **sklearn.ensemble** **import** RandomForestClassifier

In [16]:

**import** **numpy** **as** **np**

In [17]:

dataset3['CCAvg']=np.round(dataset3['CCAvg'])

C:\Users\Admin\anaconda3\lib\site-packages\ipykernel\_launcher.py:1: 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

"""Entry point for launching an IPython kernel.

In [18]:

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

In [31]:

features=['Age','Experience','Income','Family','CCAvg','Education','Mortgage','Securities Account','CD Account','Online','CreditCard']

In [32]:

rf\_model.fit(X=dataset3[features],y=dataset3['Personal Loan'])

Out[32]:

RandomForestClassifier(bootstrap=True, ccp\_alpha=0.0, class\_weight=None,

criterion='gini', max\_depth=None, max\_features=2,

max\_leaf\_nodes=None, max\_samples=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=1000,

n\_jobs=None, oob\_score=True, random\_state=None,

verbose=0, warm\_start=False)

In [33]:

print('OOB Accuracy:')

print(rf\_model.oob\_score\_);

OOB Accuracy:

0.986164026468819

In [34]:

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

print(features,imp)

Age 0.05065905138887221

Experience 0.05074421916964165

Income 0.3613433877189802

Family 0.09775727906877126

CCAvg 0.14627349354583427

Education 0.16294822825931485

Mortgage 0.047944228475813054

Securities Account 0.006403458933543363

CD Account 0.055437509445271205

Online 0.00956351909225657

CreditCard 0.01092562490170145

In [35]:

**from** **sklearn** **import** tree

In [36]:

tree\_model=tree.DecisionTreeClassifier()

In [37]:

tree\_model=tree.DecisionTreeClassifier(max\_depth=6,max\_leaf\_nodes=10)

In [39]:

predictor=pd.DataFrame([dataset3['Education'],dataset3['CCAvg'],dataset3['Income']]).T

In [40]:

tree\_model.fit(X=predictor,y=dataset3['Personal Loan'])

Out[40]:

DecisionTreeClassifier(ccp\_alpha=0.0, class\_weight=None, criterion='gini',

max\_depth=6, max\_features=None, max\_leaf\_nodes=10,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, presort='deprecated',

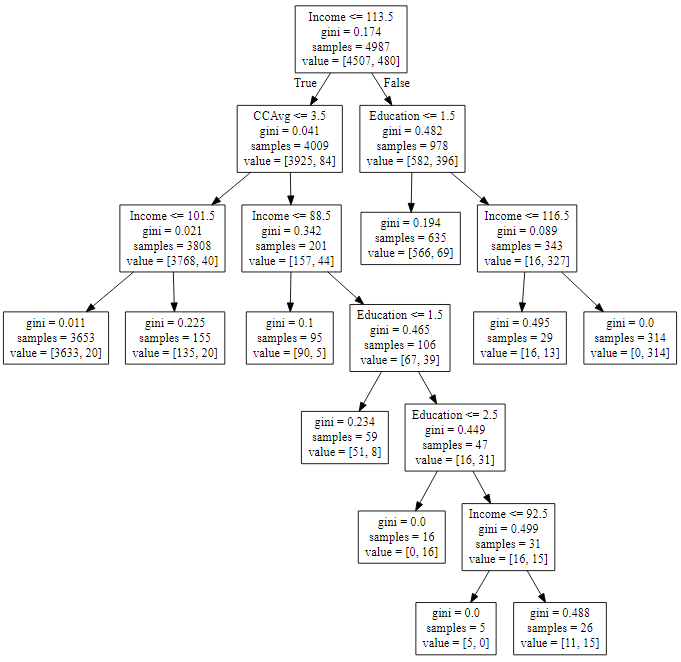
random\_state=None, splitter='best')

In [41]:

**with** open('Dtree.dot','w') **as** f:

f=tree.export\_graphviz(tree\_model,feature\_names=['Education','CCAvg','Income'],out\_file=f);

**Decision Tree :**

****