**Project 2: Build Decision Tree for Attrition Rate Analysis**

In [1]:

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

In [2]:

dataset1=pd.read\_csv(r'E:\Vivek\Data Science\Artificial Intelligence & Machine Learning\Notes\Day\_24\Dataset\general\_data.csv')

In [3]:

dataset1.columns

Out[3]:

Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager'],

dtype='object')

In [4]:

**from** **sklearn** **import** preprocessing

In [5]:

le=preprocessing.LabelEncoder()

In [6]:

dataset1['Attrition']=le.fit\_transform(dataset1['Attrition'])

In [7]:

dataset1['BusinessTravel']=le.fit\_transform(dataset1['BusinessTravel'])

In [8]:

dataset1['Department']=le.fit\_transform(dataset1['Department'])

In [9]:

dataset1['EducationField']=le.fit\_transform(dataset1['EducationField'])

In [10]:

dataset1['Gender']=le.fit\_transform(dataset1['Gender'])

In [11]:

dataset1['JobRole']=le.fit\_transform(dataset1['JobRole'])

In [12]:

dataset1['MaritalStatus']=le.fit\_transform(dataset1['MaritalStatus'])

In [13]:

dataset2=dataset1.drop(['EmployeeCount','EmployeeID','Over18','StandardHours'],axis=1)

In [14]:

dataset2.columns

Out[14]:

Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

'Education', 'EducationField', 'Gender', 'JobLevel', 'JobRole',

'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked',

'PercentSalaryHike', 'StockOptionLevel', 'TotalWorkingYears',

'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',

'YearsWithCurrManager'],

dtype='object')

In [15]:

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

In [16]:

dataset3=dataset2.dropna()

In [17]:

dataset4=dataset3.drop\_duplicates()

In [18]:

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

In [21]:

features=['Age','BusinessTravel','Department','DistanceFromHome','Education','EducationField','Gender','JobLevel','JobRole','MaritalStatus','MonthlyIncome','NumCompaniesWorked','PercentSalaryHike','StockOptionLevel','TotalWorkingYears','TrainingTimesLastYear','YearsAtCompany','YearsSinceLastPromotion','YearsWithCurrManager','JobRole']

In [22]:

rf\_model.fit(X=dataset4[features],y=dataset4['Attrition'])

Out[22]:

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 [23]:

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

0.8421768707482993

In [24]:

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

print(features,imp)

Age 0.09510429365470176

BusinessTravel 0.026276717508389773

Department 0.025487548255861532

DistanceFromHome 0.0668646830012191

Education 0.03944691915388556

EducationField 0.04021078543770036

Gender 0.018339853786024842

JobLevel 0.03542710639905045

JobRole 0.043691504203490994

MaritalStatus 0.03885042808146334

MonthlyIncome 0.08887609126550973

NumCompaniesWorked 0.053624869343934345

PercentSalaryHike 0.06289602450298613

StockOptionLevel 0.03351889890033075

TotalWorkingYears 0.08252854369257148

TrainingTimesLastYear 0.041176761815817885

YearsAtCompany 0.06765934916661773

YearsSinceLastPromotion 0.04180376471748034

YearsWithCurrManager 0.05357959014191083

JobRole 0.04463626697105315

In [41]:

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

In [42]:

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

In [47]:

predictors=pd.DataFrame([dataset4['Age'],dataset4['MonthlyIncome'],dataset4['TotalWorkingYears']]).T

In [48]:

tree\_model.fit(X=predictors,y=dataset4['Attrition'])

Out[48]:

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')

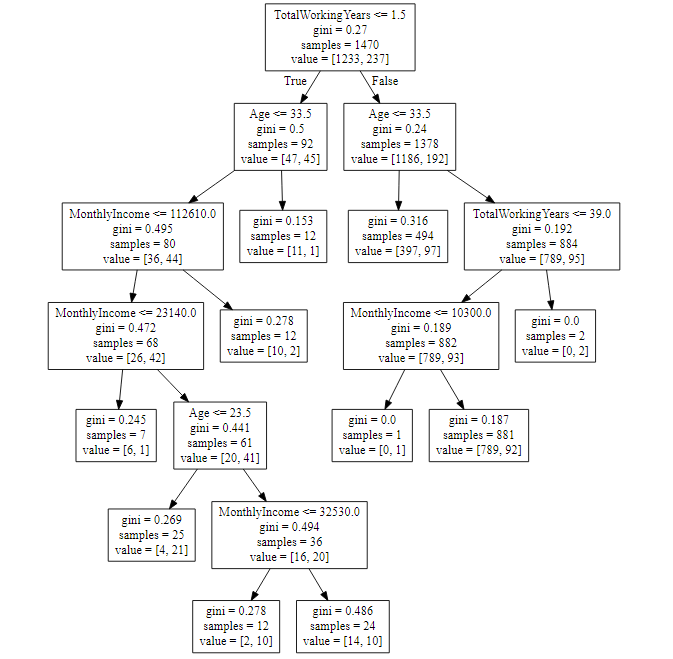
In [49]:

**with** open('Dtree2.dot','w') **as** f:

f=tree.export\_graphviz(tree\_model,feature\_names=['Age','MonthlyIncome','TotalWorkingYears'],out\_file=f);

In [ ]:

**Decision Tree:**

****