Decision tree - Attrition Analysis

In [1]:

import pandas as pd  
import numpy as np  
dataset = pd.read\_csv('Dataset/general\_data.csv')

In [2]:

from sklearn import preprocessing as pp  
  
df = dataset  
df['Attrition'] = pp.LabelEncoder().fit\_transform(df['Attrition'])  
df['BusinessTravel'] = pp.LabelEncoder().fit\_transform(df['BusinessTravel'])  
df['Department'] = pp.LabelEncoder().fit\_transform(df['Department'])  
df['EducationField'] = pp.LabelEncoder().fit\_transform(df['EducationField'])  
df['Gender'] = pp.LabelEncoder().fit\_transform(df['Gender'])  
df['JobRole'] = pp.LabelEncoder().fit\_transform(df['JobRole'])  
df['MaritalStatus'] = pp.LabelEncoder().fit\_transform(df['MaritalStatus'])

In [3]:

df1 = df.drop(['EmployeeCount','EmployeeID','Over18','StandardHours'], axis=1)  
  
df1 = df1.dropna()  
df2 = df1.drop\_duplicates()  
  
df2['TotalWorkingYears'] = np.round(df1['TotalWorkingYears'])  
df2['MonthlyIncome'] = np.round(df1['MonthlyIncome'])  
  
df2.columns

C:\Users\Joy\anaconda3\lib\site-packages\ipykernel\_launcher.py:6: 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  
   
C:\Users\Joy\anaconda3\lib\site-packages\ipykernel\_launcher.py:7: 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  
 import sys

In [4]:

from sklearn.ensemble import RandomForestClassifier  
  
rf\_model = RandomForestClassifier(n\_estimators=1000,max\_features=2,oob\_score=True)  
  
features = ['Age','BusinessTravel','Department','DistanceFromHome','Education','EducationField','Gender','JobLevel','JobRole','MaritalStatus','MonthlyIncome','NumCompaniesWorked','PercentSalaryHike','StockOptionLevel','TotalWorkingYears','TrainingTimesLastYear','YearsAtCompany','YearsSinceLastPromotion','YearsWithCurrManager']  
  
rf\_model.fit(X=df2[features],y=df2["Attrition"])

Out[4]:

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

In [5]:

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

OOB Accuracy score: 0.8435374149659864

In [20]:

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

Age : 0.09852488510388213  
BusinessTravel : 0.02770405623412496  
Department : 0.026328463168833662  
DistanceFromHome : 0.06818625439712718  
Education : 0.039904481182891194  
EducationField : 0.041648401211808225  
Gender : 0.0186656353184396  
JobLevel : 0.03755174955068577  
JobRole : 0.054915046948035146  
MaritalStatus : 0.03912653520472101  
MonthlyIncome : 0.09366782127599364  
NumCompaniesWorked : 0.05586304925372595  
PercentSalaryHike : 0.06561988914637865  
StockOptionLevel : 0.03359604522747235  
TotalWorkingYears : 0.08696108257433229  
TrainingTimesLastYear : 0.04418301437360989  
YearsAtCompany : 0.06843071882564983  
YearsSinceLastPromotion : 0.04434878603852514  
YearsWithCurrManager : 0.054774084963763514

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

##### **\_\_**1. Age[¶](#Xbc912a1c65a16005e586af098ab9dd1ced19fc6)

##### **\_\_**2. MonthlyIncome[¶](#Xb1051ffda1952bdfde48fda00ab2aa5837a3f9c)

##### **\_\_**3. TotalWorkingYears[¶](#X749f7d23cd9839306fb52453320acae80b10a24)

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

In [21]:

from sklearn import tree  
  
tree\_model = tree.DecisionTreeClassifier(max\_depth=6, max\_leaf\_nodes=12)  
  
cl\_data = pd.DataFrame([df2["Age"],df2["MonthlyIncome"],df2["TotalWorkingYears"]]).T  
  
tree\_model.fit(X=cl\_data,y=df2["Attrition"])

Out[21]:

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

In [22]:

with open("Dtree\_Attrition.dot","w") as f:  
 f=tree.export\_graphviz(tree\_model,feature\_names=["Age","MonthlyIncome","TotalWorkingYears"],out\_file=f)

### Causes for attrition occurances[¶](#Causes-for-attrition-occurances)

##### 1. Person having experience less than 1.5 years and age is less than 33.5 then attrition occurance is high.[¶](#X4a9f45665440dafe6feabe114805d39c5d32c32)

##### 2. Person having experience less than 1.5 years and age is less than 33.5 and Monthly Income < 112610 then attrition occurance is high.[¶](#X539b7a2938783d096e7f391ee8b17a8d4730beb)

##### 3. Person having experience less than 1.5 years and age is less than 23.5 and Monthly Income < 23140 then attrition occurance is high.[¶](#X90dd5725810f2d861efbb51d9146da329a0d9b8)

##### 4. Person having experience less than 1.5 years and age is less than 18.5 and Monthly Income < 32530 then attrition occurance is high.[¶](#X60c175c353ed615eba69e3a231a55eb8bff8200)

##### 5. Person having experience more than 1.5 years and age is more than 33.5 then attrition occurance is low.[¶](#Xf8d351cd5dfc66f9fcc42fabbd3fe7f22bf383c)

##### 6. Person having experience more than 39 years and and Monthly Income < 10300 then attrition occurance is high.[¶](#X6c93defb8cd53fbc0391aa2eeb4c88d37464514)

##### 7. Person having experience more than 5.5 years and and Monthly Income > 10300 then attrition occurance is low.[¶](#Xa6a0fd1d9cf6dd5bc0a5d26d07e5522137caae8)