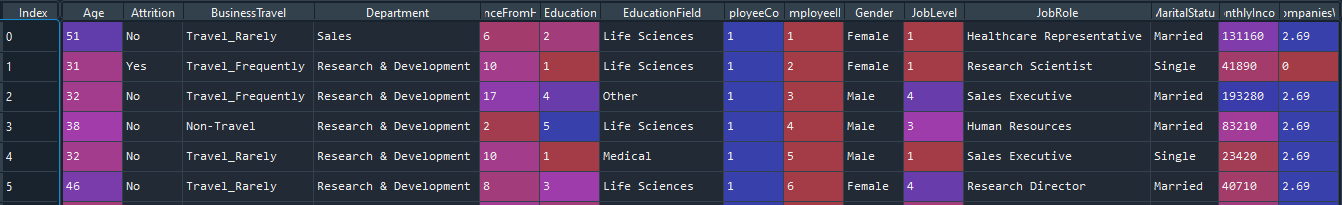
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

dataset = pd.read\_csv("D:\\ML\\dataset\\general\_data.csv")



**### Check null value in dataset**

dataset\_null = dataset.isnull().sum()



**### replace the null values**

new\_numcompaniesworked = np.where(dataset['NumCompaniesWorked'],2.69,dataset['NumCompaniesWorked'])

dataset['NumCompaniesWorked']= new\_numcompaniesworked

new\_totalworkingyears=np.where(dataset['TotalWorkingYears'],11.27,dataset['TotalWorkingYears'])

dataset['TotalWorkingYears']=new\_totalworkingyears

**##### drop other features**

df = dataset.drop(columns=['EmployeeCount','EmployeeID','Over18','StandardHours'])

**##### Label encoding technique used to change values**

from sklearn import preprocessing

le = preprocessing.LabelEncoder()

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

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

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

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

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

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

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

**##### To Model Creation**

from sklearn import tree

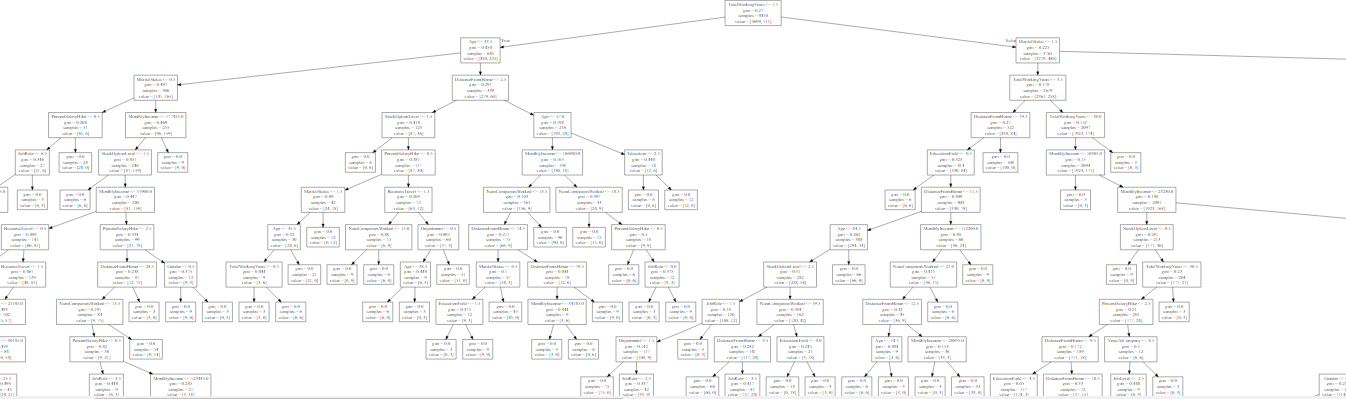
tree\_model = tree.DecisionTreeClassifier(max\_depth=38)

predictors = pd.DataFrame([df['Age'],df['BusinessTravel'],df['Department'],df['DistanceFromHome'],df['Education'],df['EducationField'],df['Gender'],df['JobLevel'],df['JobRole'],df['MaritalStatus'],df['MonthlyIncome'],df['PercentSalaryHike'],df['StockOptionLevel'],df['TrainingTimesLastYear'],df['YearsAtCompany'],df['YearsSinceLastPromotion'],df['YearsWithCurrManager'],df['NumCompaniesWorked'],df['TotalWorkingYears']]).T

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

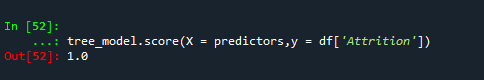
with open("Dattrition",'w') as f:

f = tree.export\_graphviz(tree\_model,feature\_names=['Age','BusinessTravel', 'Department', 'DistanceFromHome','Education', 'EducationField', 'Gender', 'JobLevel', 'JobRole','MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked','PercentSalaryHike', 'StockOptionLevel', 'TotalWorkingYears','TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion','YearsWithCurrManager'],out\_file=f)



**###### Model Accuracy**

tree\_model.score(X = predictors,y = df['Attrition'])



**#### Random Forest Classifier**

from sklearn.ensemble import RandomForestClassifier

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

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

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

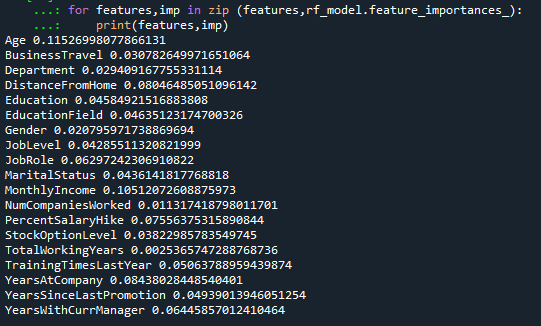
print("OOB\_Score:",rf\_model.oob\_score\_)



**#####To Find the Important variable**

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

print(features,imp)



**##### important feature are “Age”, ”MonthlyIncome”, ”YearAtCompany”, ”DistanceFromHome”, ”PercentSalaryHike”. Those are all independent variable. Dependent variable is “Attrition”.**

**##### Bulid Decision Tree Models**

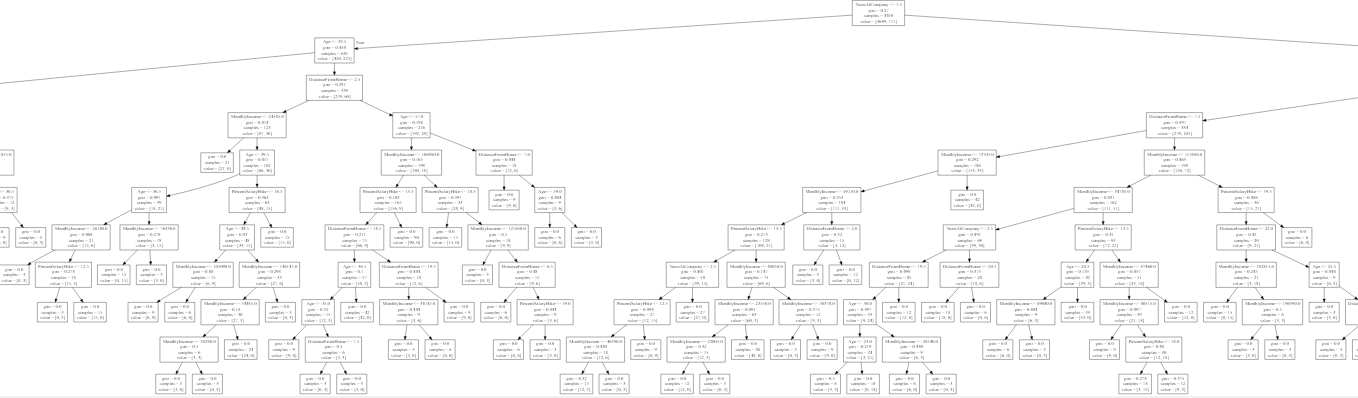
tree\_model = tree.DecisionTreeClassifier(max\_depth=10)

predictors = pd.DataFrame([df["Age"],df["MonthlyIncome"],df["YearsAtCompany"],df["DistanceFromHome"],df["PercentSalaryHike"]]).T

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

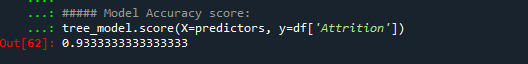
with open ("Dattrition1.dot",'w') as f:

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

****

**##### Model Accuracy score:**

tree\_model.score(X=predictors, y=df['Attrition'])

****

**##### Bulid Decision Tree Models for Indepent variable = Age,MontlyIncome**

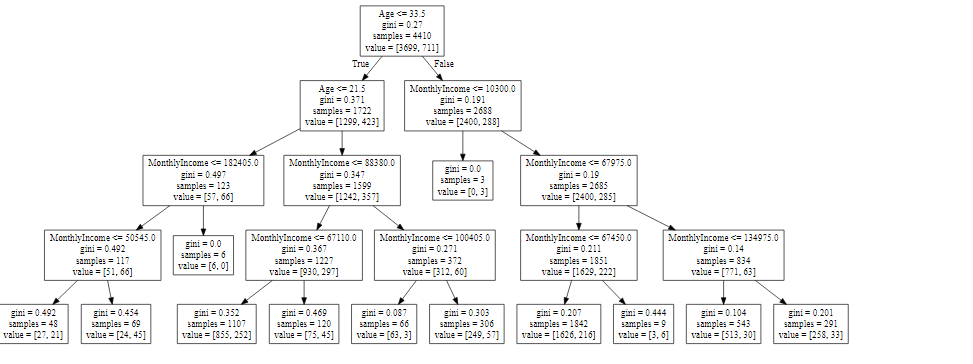
tree\_model = tree.DecisionTreeClassifier(max\_depth=4)

predictors = pd.DataFrame([df["Age"],df["MonthlyIncome"]]).T

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

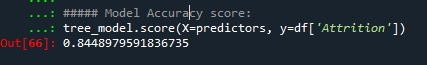
with open ("Dattrition2.dot",'w') as f:

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

****

**##### Model Accuracy score:**

tree\_model.score(X=predictors, y=df['Attrition'])

****