**Attrition Decision Tree**

**Importing the packages**

import pandas as pd

import numpy as np

from sklearn import tree

from sklearn import preprocessing

### Loading the dataset

data = pd.read\_csv('general\_data.csv')

data.head()

Out[6]:

Age Attrition ... YearsSinceLastPromotion YearsWithCurrManager

0 51 No ... 0 0

1 31 Yes ... 1 4

2 32 No ... 0 3

3 38 No ... 7 5

4 32 No ... 0 4

[5 rows x 24 columns]

### Checking for null values

data.isna().sum()

Out[7]:

Age 0

Attrition 0

BusinessTravel 0

Department 0

DistanceFromHome 0

Education 0

EducationField 0

EmployeeCount 0

EmployeeID 0

Gender 0

JobLevel 0

JobRole 0

MaritalStatus 0

MonthlyIncome 0

NumCompaniesWorked 19

Over18 0

PercentSalaryHike 0

StandardHours 0

StockOptionLevel 0

TotalWorkingYears 9

TrainingTimesLastYear 0

YearsAtCompany 0

YearsSinceLastPromotion 0

YearsWithCurrManager 0

dtype: int64

### Filling the null values

data['NumCompaniesWorked'] = data['NumCompaniesWorked'].fillna(method='pad')

data['TotalWorkingYears'] = data['TotalWorkingYears'].fillna(method='pad')

data.isna().sum()

Out[10]:

Age 0

Attrition 0

BusinessTravel 0

Department 0

DistanceFromHome 0

Education 0

EducationField 0

EmployeeCount 0

EmployeeID 0

Gender 0

JobLevel 0

JobRole 0

MaritalStatus 0

MonthlyIncome 0

NumCompaniesWorked 0

Over18 0

PercentSalaryHike 0

StandardHours 0

StockOptionLevel 0

TotalWorkingYears 0

TrainingTimesLastYear 0

YearsAtCompany 0

YearsSinceLastPromotion 0

YearsWithCurrManager 0

dtype: int64

### Converting string to int

l\_enc = preprocessing.LabelEncoder()

data['Attrition'] = l\_enc.fit\_transform(data['Attrition'])

data['BusinessTravel'] = l\_enc.fit\_transform(data['BusinessTravel'])

data['Department'] = l\_enc.fit\_transform(data['Department'])

data['EducationField'] = l\_enc.fit\_transform(data['EducationField'])

data['Gender'] = l\_enc.fit\_transform(data['Gender'])

data['JobRole'] = l\_enc.fit\_transform(data['JobRole'])

data['Over18'] = l\_enc.fit\_transform(data['Over18'])

data['MaritalStatus'] = l\_enc.fit\_transform(data['MaritalStatus'])

### Random Forest

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

features = ['Age', 'BusinessTravel', 'Department', 'DistanceFromHome','Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender','JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome','NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours','StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear','YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']

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

rf\_model.oob\_score\_

Out[14]: 0.9997732426303855

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

print(features,imp)

Age 0.0929216339468433

BusinessTravel 0.027253137140807494

Department 0.02528805908056082

DistanceFromHome 0.06718574567253537

Education 0.03975555929182885

EducationField 0.039892531256717434

EmployeeCount 0.0

EmployeeID 0.037362672442453414

Gender 0.016715138079544854

JobLevel 0.03602588353094581

JobRole 0.05365748889999081

MaritalStatus 0.03870528802112051

MonthlyIncome 0.08852071836159893

NumCompaniesWorked 0.053904934682478395

Over18 0.0

PercentSalaryHike 0.06224449684965307

StandardHours 0.0

StockOptionLevel 0.03180753248658706

TotalWorkingYears 0.08238375935324435

TrainingTimesLastYear 0.04397007834037007

YearsAtCompany 0.0677789585644639

YearsSinceLastPromotion 0.04177923862903743

YearsWithCurrManager 0.052847145369218106

#### Inference: This shows that the important features are Age, DistanceFromHome, MonthlyIncome, PercentSalaryHike, TotalWorkingYears,YearsAtCompany

### Decision Tree

tree\_model = tree.DecisionTreeClassifier(max\_depth=3)

predictors = pd.DataFrame([data['Age'],data['DistanceFromHome'],data['MonthlyIncome'],data['PercentSalaryHike'],data['TotalWorkingYears'],data['YearsAtCompany']]).T

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

Out[16]:

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

max\_depth=12, max\_features=None, max\_leaf\_nodes=None,

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',

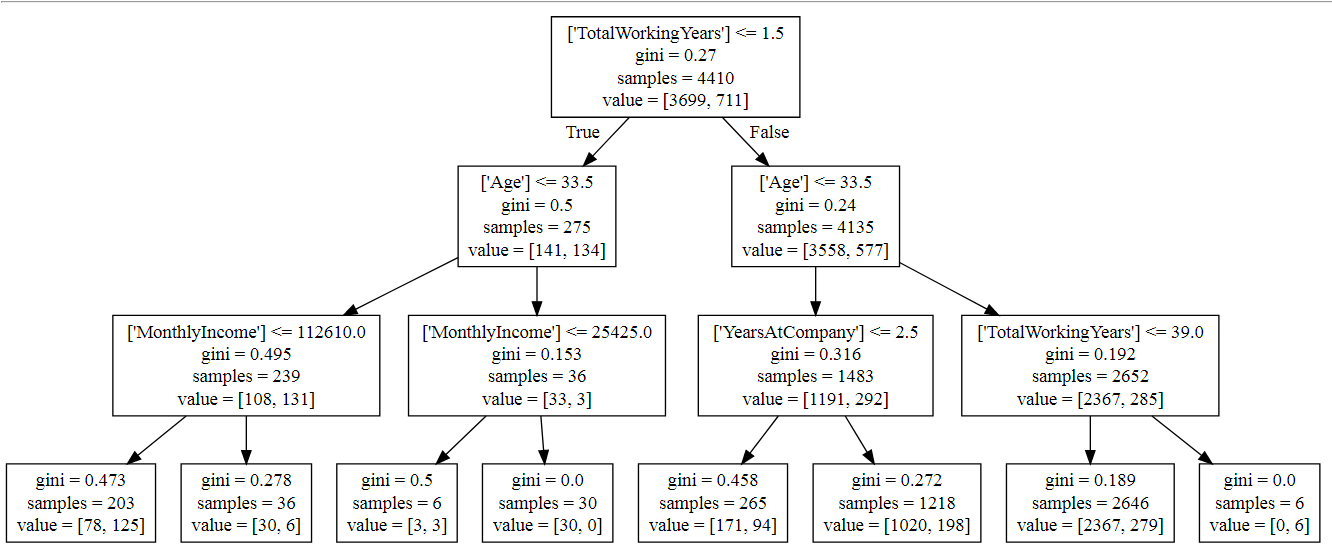
random\_state=None, splitter='best')

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

Out[18]: 0.8507936507936508

with open('Dtree5.dot','w') as f:

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



**Rules corresponding to given tree:**

1. **From total 4410 employee 3699 employee doesn’t left the company and 711 had left the company.**
2. **If age is <=33.5 and Monthly Income<=112610 then from 239 employee 108 employee doesn’t left the company,131 employee left the company.**
3. **Monthly Income<=25425,from total 36 employee 33 doesn’t left the company, 3 employee left the company.**
4. **If age is > 33.5 and Monthly Income > 25425, no employee left the company.**
5. **If age is <=33.5 and years at company <=2.5 then from 1483 employee 1192 employee doesn’t left the company and 292 employee left the company.**
6. **If Total Working years <=39.0 from 2646 employee, 2367 has not left the company and 279 employee left the company.**
7. **If Total Working years >39.0 then 6 employee left the company.**