**Random Forest on Employee Attrition**

**Code**

# -\*- coding: utf-8 -\*-

"""

Spyder Editor

This is a temporary script file.

"""

#DECISION TREE APPROACH

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cross\_validation import train\_test\_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import classification\_report,confusion\_matrix

from sklearn.metrics import accuracy\_score

#Reading the data from the excel

df = pd.read\_csv('D:\\Desktop\\Attrition python\_new.csv')

#Assigning headers to the dataframe

df.columns = ['Age','DistanceFromHome','Education']

df.head()

#Checking the total data in the dataframe

[df.info](http://df.info/)()

#plot to see the relationship between the variables

sns.pairplot(df,hue='Attrition1')

#Dividing the data into train

X = df.drop('Attrition1', axis=1)

print(X)

#Test

y = df['Attrition1']

print(y)

X\_train, X\_test, target\_train, target\_test = train\_test\_split(X, y, test\_size =  0.3)

dtree = DecisionTreeClassifier()

#Fit the model to the training data

dtree.fit(X\_train, target\_train)

#Prediction

pred = dtree.predict(X\_test)

#Print confusion matrix

print(confusion\_matrix(target\_test,pred))

print('\n')

print(classification\_report(target\_test,pred))

#Checking Accuracy

Accuracy = accuracy\_score(target\_test, pred, normalize = True)

print ("Train Accuracy :: ", accuracy\_score(target\_train, dtree.predict(X\_train)))

print ("Test Accuracy  :: ", accuracy\_score(target\_test, pred))

#RANDOM FOREST CLASSIFIER APPROACH

from sklearn.ensemble import RandomForestClassifier

#Random Forest Classifier model

rfc = RandomForestClassifier(n\_estimators = 200)

rfc.fit(X\_train, target\_train)

rfc\_pred = rfc.predict(X\_test)

#Print confusion matrix

print(confusion\_matrix(target\_test,rfc\_pred))

print('\n')

print(classification\_report(target\_test,rfc\_pred))

#Checking Accuracy

Accuracy = accuracy\_score(target\_test, pred, normalize = True)

print ("Train Accuracy :: ", accuracy\_score(target\_train, rfc.predict(X\_train)))

print ("Test Accuracy  :: ", accuracy\_score(target\_test, rfc\_pred))

**Output**

[[312  58]

 [ 43  28]]

             precision    recall  f1-score   support

          0       0.88      0.84      0.86       370

          1       0.33      0.39      0.36        71

avg / total       0.79      0.77      0.78       441

Train Accuracy ::  1.0

Test Accuracy  ::  0.770975056689

[[366   4]

 [ 59  12]]

             precision    recall  f1-score   support

          0       0.86      0.99      0.92       370

          1       0.75      0.17      0.28        71

avg / total       0.84      0.86      0.82       441

Train Accuracy ::  0.972736124635

Test Accuracy  ::  0.857142857143