In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
df= pd.read_csv(r'C:\Users\siyad\AppData\Local\Temp\Temp1_Dataset-20200813T141334Z-001.zip\Dataset\general_data.csv')
```

In [4]:
```
df.columns
```

Out[4]:
```
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 [5]:
```
from sklearn import preprocessing
```

In [6]:
```
le=preprocessing.LabelEncoder()
```

In [7]:
```
df['Attrition']=le.fit_transform(df['Attrition'])
```

In [8]:
```
df['BusinessTravel']=le.fit_transform(df['BusinessTravel'])
```

In [9]:
```
df['Department']=le.fit_transform(df['Department'])
```

In [10]:
```
df['EducationField']=le.fit_transform(df['EducationField'])
```

In [11]:
```
df['Gender']=le.fit_transform(df['Gender'])
```

In [12]:
```
df['MaritalStatus']=le.fit_transform(df['MaritalStatus'])
```

In [13]:
```
df['JobRole']=le.fit_transform(df['JobRole'])
```

In [14]:
```
df2=df.drop(['EmployeeCount', 'EmployeeID','Over18','StandardHours'],axis=1)
```

In [15]:
```
df2.columns
```

Out[15]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',
       'Education', 'EducationField', 'Gender', 'JobLevel', 'JobRole',
       'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked',
       'PercentSalaryHike', 'StockOptionLevel', 'TotalWorkingYears',
       'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

In [16]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [17]:
```
df3=df2.dropna()
```

In [18]:
```
df4=df3.drop_duplicates()
```

In [19]:
```
rf_model=RandomForestClassifier(n_estimators=1000,max_features=2,oob_score=True)
```

In [73]:
```
features=['Age', 'BusinessTravel', 'Department', 'DistanceFromHome',
       'Education', 'EducationField', 'Gender', 'JobLevel', 'JobRole',
       'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked',
       'PercentSalaryHike', 'StockOptionLevel', 'TotalWorkingYears',
       'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',
       'YearsWithCurrManager']
```

In [74]:
```
rf_model.fit(X=df4[features],y=df4['Attrition'])
```

Out[74]:
```
RandomForestClassifier(max_features=2, n_estimators=1000, oob_score=True)
```

In [75]:
```
print('OOB Accuracy: ')
print(rf_model.oob_score_)
```

```
OOB Accuracy:
0.8442176870748299
```

In [76]:
```
for feature,imp in zip(features,rf_model.feature_importances_):
    print(feature,imp)
```

```
Age 0.09830358093771875
BusinessTravel 0.027295272128891022
Department 0.0267790086457988
DistanceFromHome 0.07005069258499955
Education 0.040254050302009334
EducationField 0.04236509891074774
Gender 0.018092243627388708
JobLevel 0.03825055704360027
JobRole 0.055051555228577485
MaritalStatus 0.039747552994345944
MonthlyIncome 0.0924281792899583
NumCompaniesWorked 0.05572178237000869
PercentSalaryHike 0.06492096415661418
StockOptionLevel 0.03480373451378508
TotalWorkingYears 0.08582439794516951
TrainingTimesLastYear 0.04398216484632851
YearsAtCompany 0.06957023688738793
YearsSinceLastPromotion 0.04359490835894176
YearsWithCurrManager 0.05296401922772851
```

In [44]:
```
from sklearn import tree
```

In [45]:
```
tree_model=tree.DecisionTreeClassifier(max_depth=6,max_leaf_nodes=10)
```

In [56]:
```
pred=pd.DataFrame([df4['Age'],df4['MonthlyIncome'],df4['TotalWorkingYears']]).T
```

In [57]:
```
df4['TotalWorkingYears']=np.round(df['TotalWorkingYears'])
```

```
c:\python36\lib\site-packages\ipykernel_launcher.py:1: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  """Entry point for launching an IPython kernel.
```

In [58]:
```
tree_model.fit(X=pred,y=df4['Attrition'])
```

Out[58]:
```
DecisionTreeClassifier(max_depth=6, max_leaf_nodes=10)
```

In [59]:
```
with open('Dtree2.dot','w') as f:
    f=tree.export_graphviz(tree_model,feature_names=['MonthlyIncome','Age','TotalWorkingYears'],out_file=f)
```

In [60]:
```
import statsmodels.api as sm
```

In [61]:
```
Y=df4.Attrition
```

In [63]:
```
X=df4[['Age', 'BusinessTravel', 'Department', 'DistanceFromHome',
       'Education', 'EducationField', 'Gender', 'JobLevel', 'JobRole',
       'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked',
       'PercentSalaryHike', 'StockOptionLevel', 'TotalWorkingYears',
       'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',
       'YearsWithCurrManager']]
```

In [64]:
```
X1=sm.add_constant(X)
```

In [65]:
```
Logistic_Att=sm.Logit(Y,X1)
```

In [66]:
```
Result=Logistic_Att.fit()
```

```
Optimization terminated successfully.
         Current function value: 0.392756
         Iterations 7
```

In [68]:
```
Result.summary()
```

Out[68]:

Logit Regression Results

Dep. Variable:	Attrition	**No. Observations:**	1470
Model:	Logit	**Df Residuals:**	1450
Method:	MLE	**Df Model:**	19
Date:	Sat, 15 Aug 2020	**Pseudo R-squ.:**	0.1108
Time:	01:00:23	**Log-Likelihood:**	-577.35
converged:	True	**LL-Null:**	-649.29
Covariance Type:	nonrobust	**LLR p-value:**	3.295e-21

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0650	0.717	0.091	0.928	-1.340	1.470
Age	-0.0306	0.012	-2.583	0.010	-0.054	-0.007
BusinessTravel	-0.0166	0.113	-0.146	0.884	-0.239	0.206
Department	-0.2421	0.141	-1.720	0.085	-0.518	0.034
DistanceFromHome	-0.0014	0.009	-0.145	0.884	-0.020	0.017
Education	-0.0625	0.074	-0.847	0.397	-0.207	0.082
EducationField	-0.0965	0.058	-1.669	0.095	-0.210	0.017
Gender	0.0869	0.155	0.560	0.576	-0.217	0.391
JobLevel	-0.0249	0.069	-0.363	0.717	-0.159	0.110
JobRole	0.0378	0.031	1.219	0.223	-0.023	0.099
MaritalStatus	0.5885	0.109	5.379	0.000	0.374	0.803
MonthlyIncome	-1.868e-06	1.66e-06	-1.128	0.259	-5.11e-06	1.38e-06
NumCompaniesWorked	0.1184	0.032	3.729	0.000	0.056	0.181
PercentSalaryHike	0.0117	0.020	0.576	0.565	-0.028	0.052
StockOptionLevel	-0.0645	0.089	-0.721	0.471	-0.240	0.111
TotalWorkingYears	-0.0593	0.021	-2.856	0.004	-0.100	-0.019
TrainingTimesLastYear	-0.1465	0.061	-2.406	0.016	-0.266	-0.027
YearsAtCompany	0.0136	0.032	0.428	0.669	-0.049	0.076
YearsSinceLastPromotion	0.1323	0.035	3.732	0.000	0.063	0.202
YearsWithCurrManager	-0.1396	0.038	-3.642	0.000	-0.215	-0.064

In [77]:
```
from scipy.stats import pearsonr
stats,p=pearsonr(df.Attrition,df.MonthlyIncome)
print(stats,p)
```

```
-0.031176281698115076 0.03842748490600132
```

In [ ]: