Day 21 - Logistic Regression - Attrition

# We are going to predict the probabilty of attrition according to the given dataset using the logistic regression method.[¶](#Xabde165d73ac1506cbb695b024bda80c5e0285b)

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'])  
  
df.columns

Out[2]:

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 [44]:

df1 = df.drop(['EmployeeCount','EmployeeID','Over18','StandardHours'], axis=1)  
  
df1 = df1.dropna()  
  
df1['TotalWorkingYears'] = np.round(df['TotalWorkingYears'])  
df1['MonthlyIncome'] = np.round(df['MonthlyIncome'])  
df1['Age'] = np.round(df['Age'])  
  
df1.head()

Out[44]:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Age | Attrition | BusinessTravel | Department | DistanceFromHome | Education | EducationField | Gender | JobLevel | JobRole | MaritalStatus | MonthlyIncome | NumCompaniesWorked | PercentSalaryHike | StockOptionLevel | TotalWorkingYears | TrainingTimesLastYear | YearsAtCompany | YearsSinceLastPromotion | YearsWithCurrManager |
| 0 | 51 | 0 | 2 | 2 | 6 | 2 | 1 | 0 | 1 | 0 | 1 | 131160 | 1.0 | 11 | 0 | 1.0 | 6 | 1 | 0 | 0 |
| 1 | 31 | 1 | 1 | 1 | 10 | 1 | 1 | 0 | 1 | 6 | 2 | 41890 | 0.0 | 23 | 1 | 6.0 | 3 | 5 | 1 | 4 |
| 2 | 32 | 0 | 1 | 1 | 17 | 4 | 4 | 1 | 4 | 7 | 1 | 193280 | 1.0 | 15 | 3 | 5.0 | 2 | 5 | 0 | 3 |
| 3 | 38 | 0 | 0 | 1 | 2 | 5 | 1 | 1 | 3 | 1 | 1 | 83210 | 3.0 | 11 | 3 | 13.0 | 5 | 8 | 7 | 5 |
| 4 | 32 | 0 | 2 | 1 | 10 | 1 | 3 | 1 | 1 | 7 | 2 | 23420 | 4.0 | 12 | 2 | 9.0 | 2 | 6 | 0 | 4 |

## Performing Logistic Regression training[¶](#Performing-Logistic-Regression-training)

In [9]:

Y = df1['Attrition']  
X = df1[['Age', 'BusinessTravel', 'Department', 'DistanceFromHome','Education', 'EducationField','Gender','JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome','NumCompaniesWorked','PercentSalaryHike','StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear','YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']]  
  
import statsmodels.api as sm  
  
X1 = sm.add\_constant(X)  
  
logist = sm.Logit(Y,X1)  
result = logist.fit()  
  
print(result.summary())

Optimization terminated successfully.  
 Current function value: 0.392916  
 Iterations 7  
 Logit Regression Results   
==============================================================================  
Dep. Variable: Attrition No. Observations: 4382  
Model: Logit Df Residuals: 4362  
Method: MLE Df Model: 19  
Date: Wed, 12 Aug 2020 Pseudo R-squ.: 0.1093  
Time: 16:04:13 Log-Likelihood: -1721.8  
converged: True LL-Null: -1933.1  
Covariance Type: nonrobust LLR p-value: 8.681e-78  
===========================================================================================  
 coef std err z P>|z| [0.025 0.975]  
-------------------------------------------------------------------------------------------  
const 0.0270 0.414 0.065 0.948 -0.785 0.839  
Age -0.0307 0.007 -4.478 0.000 -0.044 -0.017  
BusinessTravel -0.0137 0.066 -0.209 0.834 -0.143 0.115  
Department -0.2229 0.082 -2.735 0.006 -0.383 -0.063  
DistanceFromHome -0.0012 0.005 -0.231 0.818 -0.012 0.009  
Education -0.0664 0.043 -1.555 0.120 -0.150 0.017  
EducationField -0.0954 0.034 -2.849 0.004 -0.161 -0.030  
Gender 0.0855 0.090 0.952 0.341 -0.091 0.262  
JobLevel -0.0285 0.040 -0.716 0.474 -0.107 0.050  
JobRole 0.0400 0.018 2.226 0.026 0.005 0.075  
MaritalStatus 0.5835 0.063 9.212 0.000 0.459 0.708  
MonthlyIncome -1.815e-06 9.57e-07 -1.897 0.058 -3.69e-06 6.01e-08  
NumCompaniesWorked 0.1174 0.018 6.390 0.000 0.081 0.153  
PercentSalaryHike 0.0126 0.012 1.067 0.286 -0.011 0.036  
StockOptionLevel -0.0675 0.052 -1.302 0.193 -0.169 0.034  
TotalWorkingYears -0.0584 0.012 -4.873 0.000 -0.082 -0.035  
TrainingTimesLastYear -0.1443 0.035 -4.097 0.000 -0.213 -0.075  
YearsAtCompany 0.0132 0.018 0.718 0.473 -0.023 0.049  
YearsSinceLastPromotion 0.1328 0.020 6.479 0.000 0.093 0.173  
YearsWithCurrManager -0.1394 0.022 -6.288 0.000 -0.183 -0.096  
===========================================================================================

### Here according to the p-value except for "BusinessTravel", "DistanceFromHome", "Education", "Gender", "JobLevel", "PercentSalaryHike", "StockOptionLevel", "YearsAtCompany", the rest of the variables are significant in finding the attrition status.[¶](#X65bee2d4ce2197192dc724b34908b20b07ac776)

### Now Creating the model with significant variables[¶](#X03cbb95cf9fc2aaa7568595aad5a7b8cc16086a)

In [43]:

# Calculated Coefficient  
  
B0 = 0.0270  
AgeX = -0.0307  
DepartmentX = -0.2229  
EducationFieldX = -0.0954  
JobRoleX = 0.0400  
MaritalStatusX = 0.5835  
MonthlyIncomeX = -1.815e-06  
NumCompaniesWorkedX = 0.1174  
TotalWorkingYearsX = -0.0584   
TrainingTimesLastYearX = -0.1443  
YearsSinceLastPromotionX = 0.1328  
YearsWithCurrManagerX = -0.1394  
  
  
#input values for probability prediction  
  
Age = 27  
Department = 1  
EducationField = 1  
JobRole = 6  
MaritalStatus = 2  
MonthlyIncome = 41600  
NumCompaniesWorked = 3  
TotalWorkingYears = 3   
TrainingTimesLastYear = 2  
YearsSinceLastPromotion = 0  
YearsWithCurrManager = 0  
  
  
# Probability model equation  
import math  
  
p = 1/(1+math.exp(-(B0+(Age\*AgeX)+(Department\*DepartmentX)+(EducationField\*EducationFieldX)+(JobRole\*JobRoleX)+(MaritalStatus\*MaritalStatusX)+(MonthlyIncome\*MonthlyIncomeX)+(NumCompaniesWorked\*NumCompaniesWorkedX)+(TotalWorkingYears\*TotalWorkingYearsX)+(TrainingTimesLastYear\*TrainingTimesLastYearX)+(YearsSinceLastPromotion\*YearsSinceLastPromotionX)+(YearsWithCurrManager\*YearsWithCurrManagerX))))   
print("Probability of attrition is ", p)

Probability of attrition is 0.5249033765876221

### Since the Probability of attrition is little more than 0.5 the person with the value entered above is having a slight chance of attrition in this case.[¶](#X0aa03fd733e3d04414ba9ee31682c071c643969)

In [ ]: