IBM_Employee_Attrition_Prediction

September 19, 2022

[1]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
import matplotlib.pyplot as plt
%matplotlib inline
from patsy import dmatrices
import sklearn
import seaborn as sns
```

[2]:
```
dataframe=pd.read_csv("Attrition Data.csv")
```

[3]:
```
dataframe.head()
```

[3]:
```
   Age Attrition     BusinessTravel  DailyRate              Department  \
0   41       Yes      Travel_Rarely       1102                   Sales
1   49        No  Travel_Frequently        279  Research & Development
2   37       Yes      Travel_Rarely       1373  Research & Development
3   33        No  Travel_Frequently       1392  Research & Development
4   27        No      Travel_Rarely        591  Research & Development

   DistanceFromHome  Education EducationField  EmployeeCount  EmployeeNumber  \
0                 1          2  Life Sciences              1               1
1                 8          1  Life Sciences              1               2
2                 2          2          Other              1               4
3                 3          4  Life Sciences              1               5
4                 2          1        Medical              1               7

   …  RelationshipSatisfaction StandardHours  StockOptionLevel  \
0  …                         1            80                 0
1  …                         4            80                 1
2  …                         2            80                 0
3  …                         3            80                 0
4  …                         4            80                 1

   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
0                  8                      0               1               6
1                 10                      3               3              10
```

```
2                  7                      3               3               0
3                  8                      3               3               8
4                  6                      3               3               2

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]
```

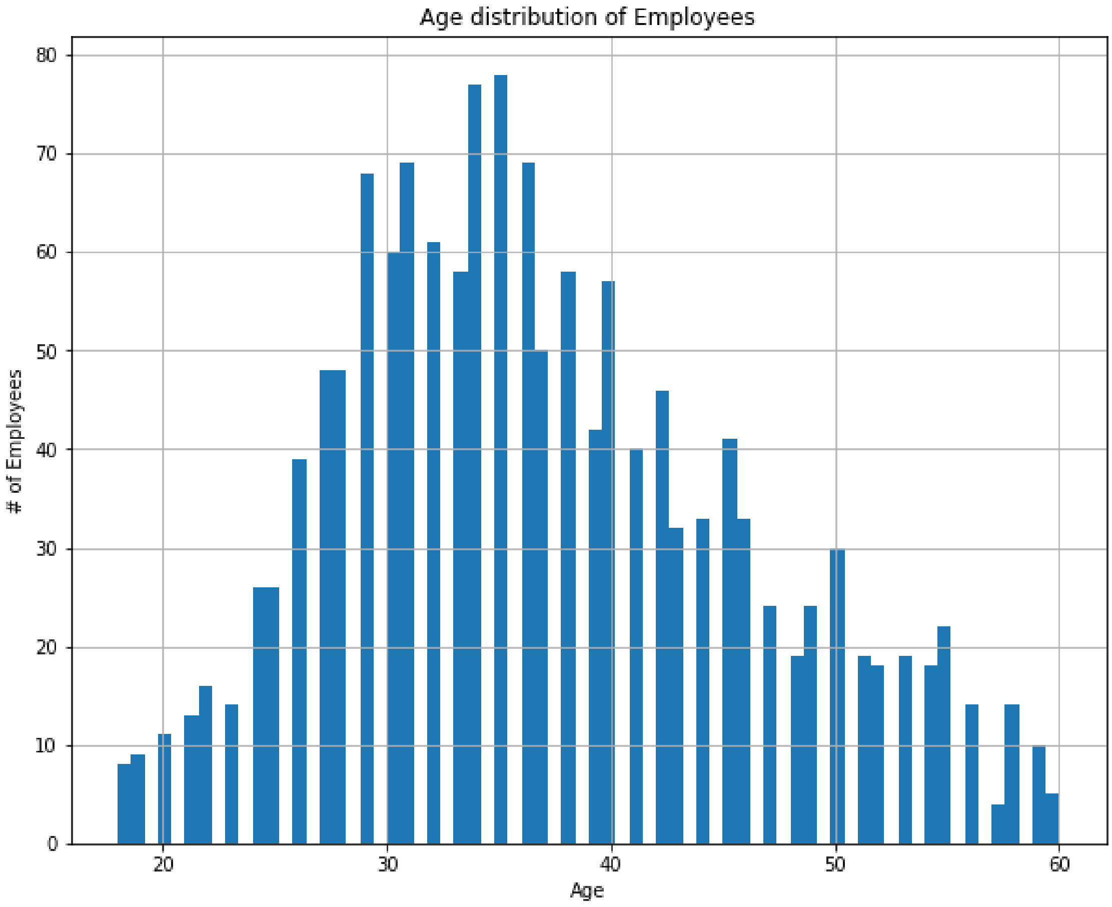
[4]:

```python
names = dataframe.columns.values
print(names)
```

```
['Age' 'Attrition' 'BusinessTravel' 'DailyRate' 'Department'
 'DistanceFromHome' 'Education' 'EducationField' 'EmployeeCount'
 'EmployeeNumber' 'EnvironmentSatisfaction' 'Gender' 'HourlyRate'
 'JobInvolvement' 'JobLevel' 'JobRole' 'JobSatisfaction' 'MaritalStatus'
 'MonthlyIncome' 'MonthlyRate' 'NumCompaniesWorked' 'Over18' 'OverTime'
 'PercentSalaryHike' 'PerformanceRating' 'RelationshipSatisfaction'
 'StandardHours' 'StockOptionLevel' 'TotalWorkingYears'
 'TrainingTimesLastYear' 'WorkLifeBalance' 'YearsAtCompany'
 'YearsInCurrentRole' 'YearsSinceLastPromotion' 'YearsWithCurrManager']
```

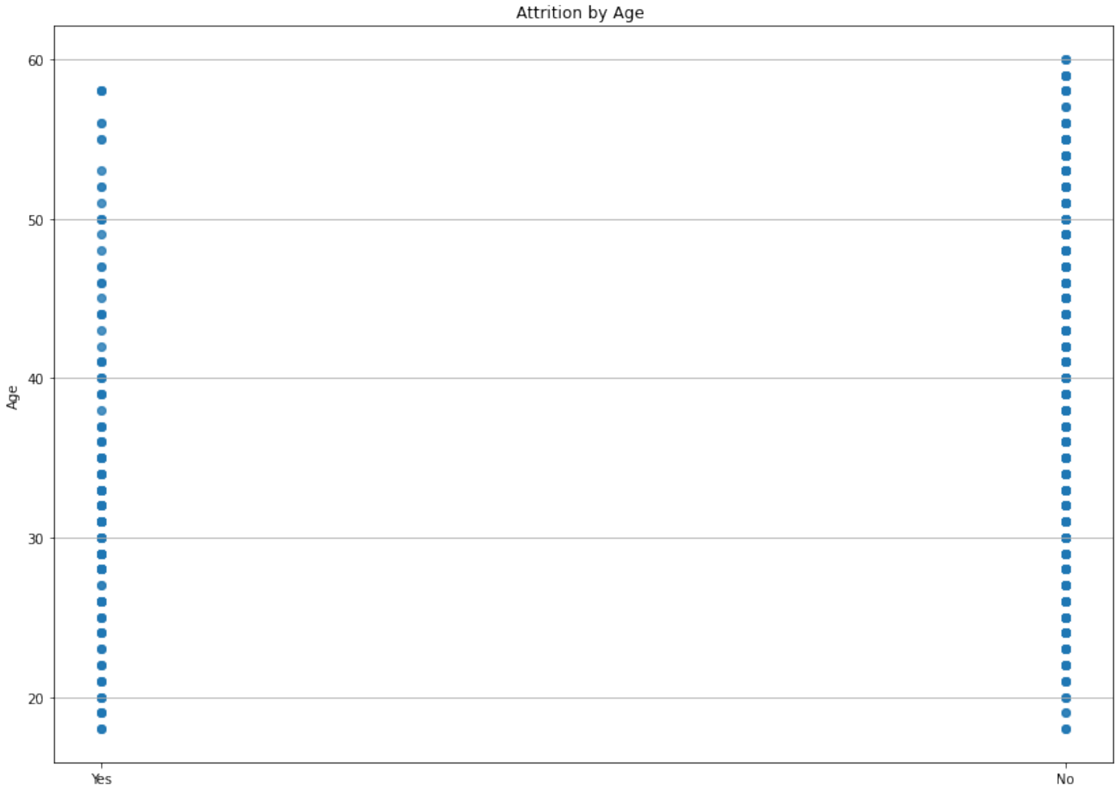
[5]:

```python
# histogram for age
plt.figure(figsize=(10,8))
dataframe['Age'].hist(bins=70)
plt.title("Age distribution of Employees")
plt.xlabel("Age")
plt.ylabel("# of Employees")
plt.show()
```

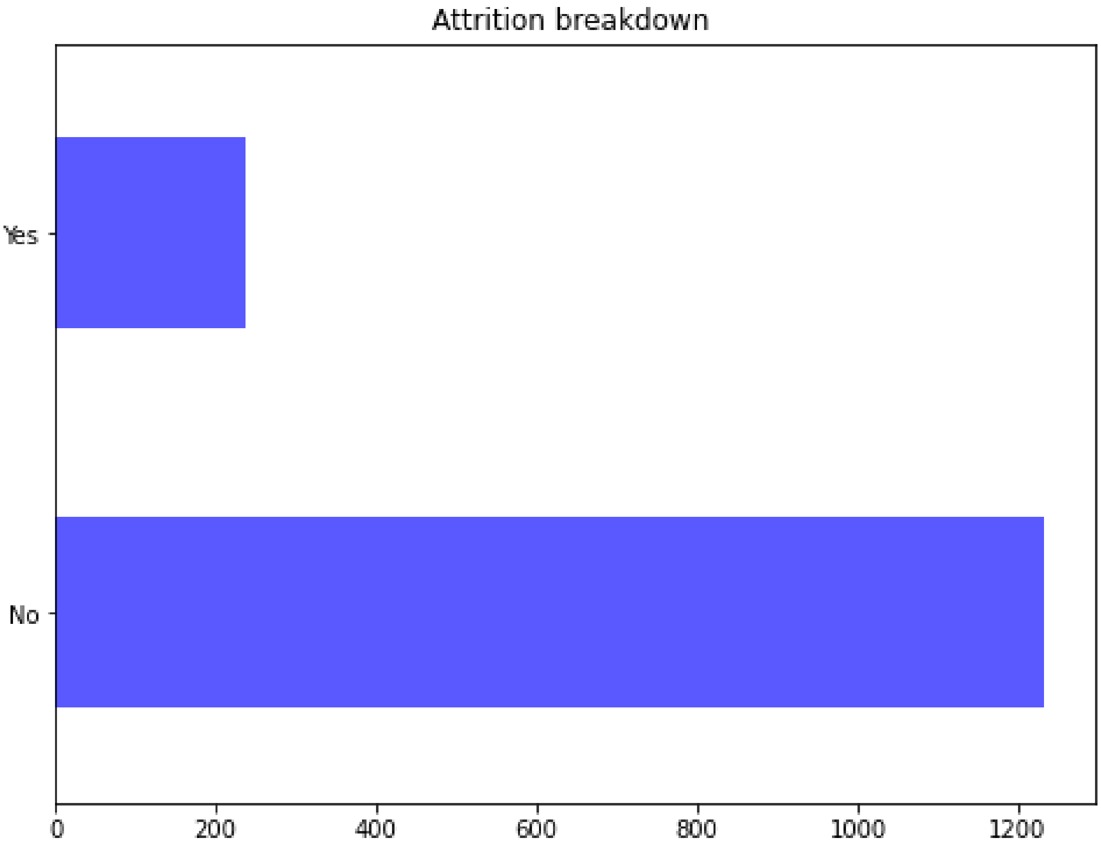


```
[6]: # explore data for Attrition by Age
plt.figure(figsize=(14,10))
plt.scatter(dataframe.Attrition,dataframe.Age, alpha=.55)
plt.title("Attrition by Age ")
plt.ylabel("Age")
plt.grid(b=True, which='major',axis='y')
plt.show()
```

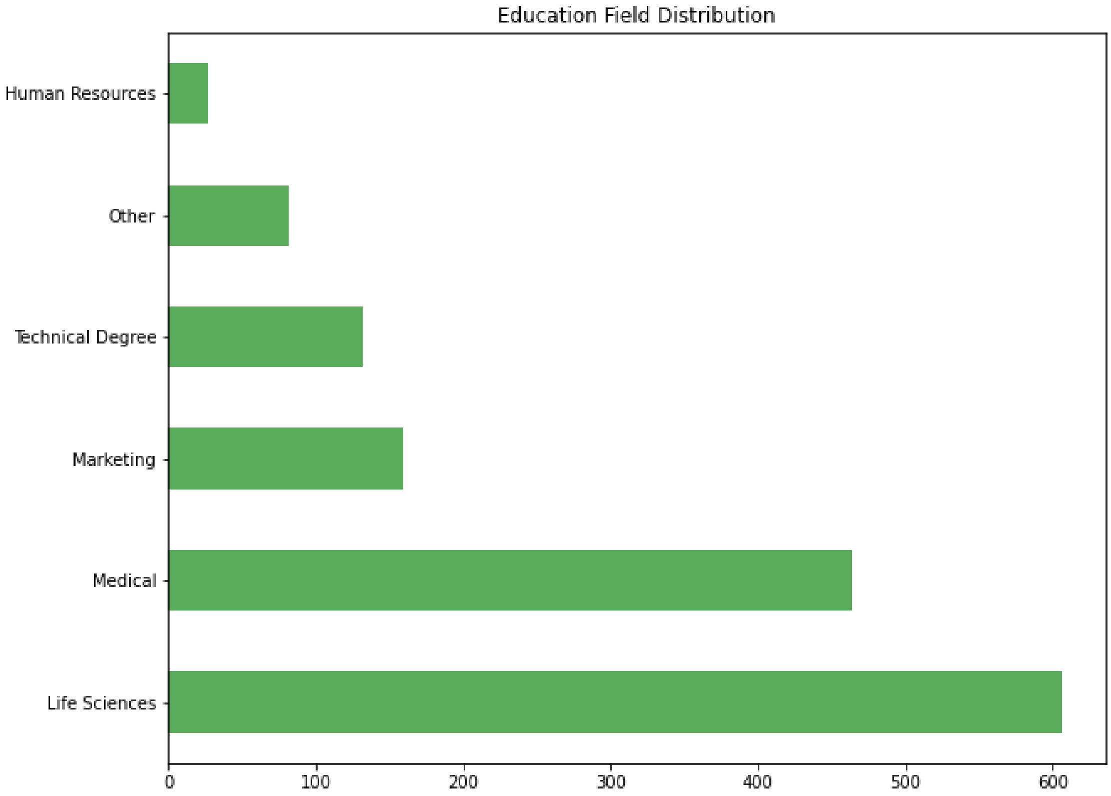
```
/usr/local/lib/python3.7/site-packages/ipykernel_launcher.py:6:
MatplotlibDeprecationWarning: The 'b' parameter of grid() has been renamed
'visible' since Matplotlib 3.5; support for the old name will be dropped two
minor releases later.
```



```
[7]: # explore data for Left employees breakdown
     plt.figure(figsize=(8,6))
     dataframe.Attrition.value_counts().plot(kind='barh',color='blue',alpha=.65)
     plt.title("Attrition breakdown ")
     plt.show()
```

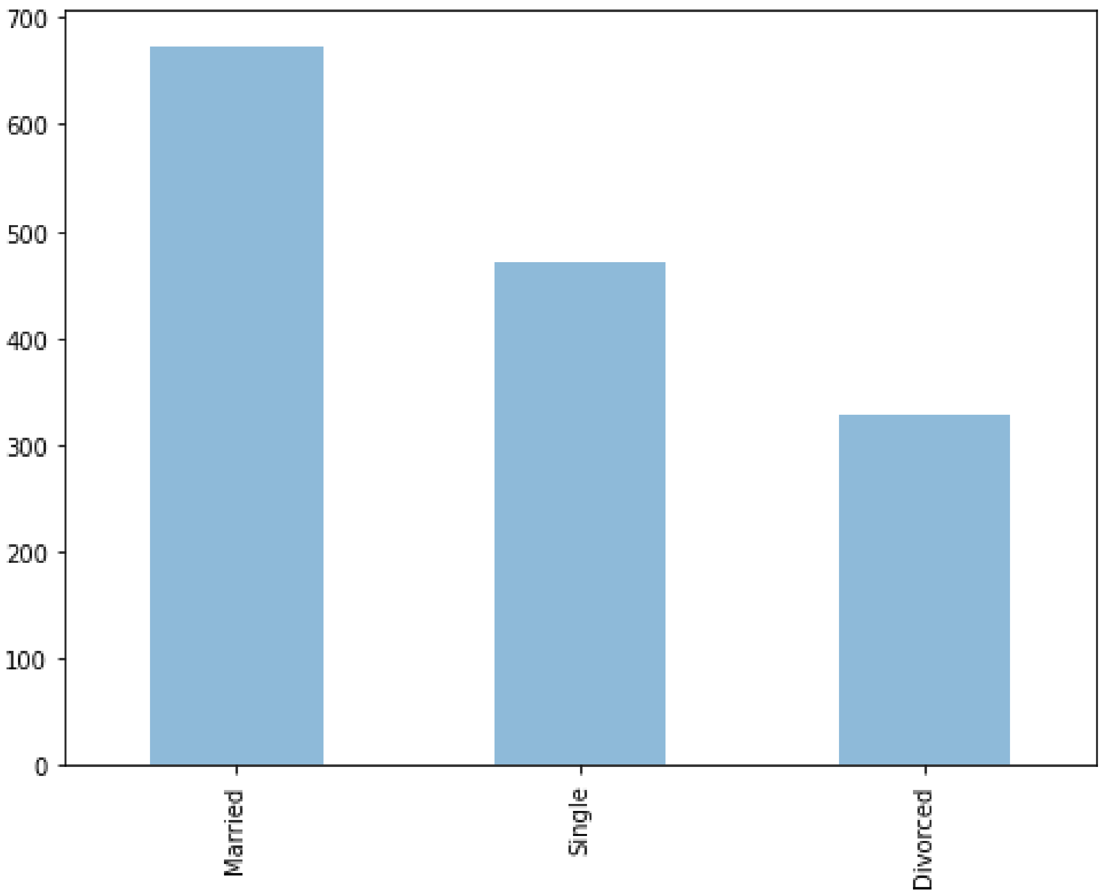


[8]:
```
# explore data for Education Field distribution
plt.figure(figsize=(10,8))
dataframe.EducationField.value_counts().plot(kind='barh',color='g',alpha=.65)
plt.title("Education Field Distribution")
plt.show()
```



[9]:

```
# explore data for Marital Status
plt.figure(figsize=(8,6))
dataframe.MaritalStatus.value_counts().plot(kind='bar',alpha=.5)
plt.show()
```



[10]:

```
dataframe.describe()
```

[10]:

```
                Age    DailyRate  DistanceFromHome    Education  EmployeeCount  \
count   1470.000000  1470.000000       1470.000000  1470.000000         1470.0
mean      36.923810   802.485714          9.192517     2.912925            1.0
std        9.135373   403.509100          8.106864     1.024165            0.0
min       18.000000   102.000000          1.000000     1.000000            1.0
25%       30.000000   465.000000          2.000000     2.000000            1.0
50%       36.000000   802.000000          7.000000     3.000000            1.0
75%       43.000000  1157.000000         14.000000     4.000000            1.0
max       60.000000  1499.000000         29.000000     5.000000            1.0

       EmployeeNumber  EnvironmentSatisfaction   HourlyRate  JobInvolvement  \
count     1470.000000              1470.000000  1470.000000     1470.000000
mean      1024.865306                 2.721769    65.891156        2.729932
std        602.024335                 1.093082    20.329428        0.711561
min          1.000000                 1.000000    30.000000        1.000000
25%        491.250000                 2.000000    48.000000        2.000000
50%       1020.500000                 3.000000    66.000000        3.000000
```

```
75%    1555.750000                  4.000000   83.750000        3.000000
max    2068.000000                  4.000000  100.000000        4.000000

          JobLevel  …  RelationshipSatisfaction  StandardHours  \
count  1470.000000  …               1470.000000         1470.0
mean      2.063946  …                  2.712245           80.0
std       1.106940  …                  1.081209            0.0
min       1.000000  …                  1.000000           80.0
25%       1.000000  …                  2.000000           80.0
50%       2.000000  …                  3.000000           80.0
75%       3.000000  …                  4.000000           80.0
max       5.000000  …                  4.000000           80.0

       StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
count       1470.000000        1470.000000            1470.000000
mean           0.793878          11.279592               2.799320
std            0.852077           7.780782               1.289271
min            0.000000           0.000000               0.000000
25%            0.000000           6.000000               2.000000
50%            1.000000          10.000000               3.000000
75%            1.000000          15.000000               3.000000
max            3.000000          40.000000               6.000000

       WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
count      1470.000000     1470.000000         1470.000000
mean          2.761224        7.008163            4.229252
std           0.706476        6.126525            3.623137
min           1.000000        0.000000            0.000000
25%           2.000000        3.000000            2.000000
50%           3.000000        5.000000            3.000000
75%           3.000000        9.000000            7.000000
max           4.000000       40.000000           18.000000

       YearsSinceLastPromotion  YearsWithCurrManager
count              1470.000000           1470.000000
mean                  2.187755              4.123129
std                   3.222430              3.568136
min                   0.000000              0.000000
25%                   0.000000              2.000000
50%                   1.000000              3.000000
75%                   3.000000              7.000000
max                  15.000000             17.000000

[8 rows x 26 columns]
```

[11]:

```
dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Age                       1470 non-null   int64 
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64 
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64 
 6   Education                 1470 non-null   int64 
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64 
 9   EmployeeNumber            1470 non-null   int64 
 10  EnvironmentSatisfaction   1470 non-null   int64 
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64 
 13  JobInvolvement            1470 non-null   int64 
 14  JobLevel                  1470 non-null   int64 
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64 
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64 
 19  MonthlyRate               1470 non-null   int64 
 20  NumCompaniesWorked        1470 non-null   int64 
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64 
 24  PerformanceRating         1470 non-null   int64 
 25  RelationshipSatisfaction  1470 non-null   int64 
 26  StandardHours             1470 non-null   int64 
 27  StockOptionLevel          1470 non-null   int64 
 28  TotalWorkingYears         1470 non-null   int64 
 29  TrainingTimesLastYear     1470 non-null   int64 
 30  WorkLifeBalance           1470 non-null   int64 
 31  YearsAtCompany            1470 non-null   int64 
 32  YearsInCurrentRole        1470 non-null   int64 
 33  YearsSinceLastPromotion   1470 non-null   int64 
 34  YearsWithCurrManager      1470 non-null   int64 
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

[12]:
```
dataframe.columns
```

[12]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

```
[13]: dataframe.std()
```

```
[13]: Age                            9.135373
      DailyRate                    403.509100
      DistanceFromHome               8.106864
      Education                      1.024165
      EmployeeCount                  0.000000
      EmployeeNumber               602.024335
      EnvironmentSatisfaction        1.093082
      HourlyRate                    20.329428
      JobInvolvement                 0.711561
      JobLevel                       1.106940
      JobSatisfaction                1.102846
      MonthlyIncome               4707.956783
      MonthlyRate                 7117.786044
      NumCompaniesWorked             2.498009
      PercentSalaryHike              3.659938
      PerformanceRating              0.360824
      RelationshipSatisfaction       1.081209
      StandardHours                  0.000000
      StockOptionLevel               0.852077
      TotalWorkingYears              7.780782
      TrainingTimesLastYear          1.289271
      WorkLifeBalance                0.706476
      YearsAtCompany                 6.126525
      YearsInCurrentRole             3.623137
      YearsSinceLastPromotion        3.222430
      YearsWithCurrManager           3.568136
      dtype: float64
```

```
[14]: dataframe['Attrition'].value_counts()
```

```
[14]: No     1233
      Yes     237
      Name: Attrition, dtype: int64
```

```
[15]: dataframe['Attrition'].dtypes
```

```
[15]: dtype('O')
```

```
[16]: dataframe['Attrition'].replace('Yes',1, inplace=True)
      dataframe['Attrition'].replace('No',0, inplace=True)
```

```
[17]: dataframe.head(10)
```

```
[17]:    Age  Attrition     BusinessTravel  DailyRate              Department  \
      0   41          1      Travel_Rarely       1102                   Sales
      1   49          0  Travel_Frequently        279  Research & Development
      2   37          1      Travel_Rarely       1373  Research & Development
      3   33          0  Travel_Frequently       1392  Research & Development
      4   27          0      Travel_Rarely        591  Research & Development
      5   32          0  Travel_Frequently       1005  Research & Development
      6   59          0      Travel_Rarely       1324  Research & Development
      7   30          0      Travel_Rarely       1358  Research & Development
      8   38          0  Travel_Frequently        216  Research & Development
      9   36          0      Travel_Rarely       1299  Research & Development

         DistanceFromHome  Education EducationField  EmployeeCount  EmployeeNumber  \
      0                 1          2  Life Sciences              1               1
      1                 8          1  Life Sciences              1               2
      2                 2          2          Other              1               4
      3                 3          4  Life Sciences              1               5
      4                 2          1        Medical              1               7
      5                 2          2  Life Sciences              1               8
      6                 3          3        Medical              1              10
      7                24          1  Life Sciences              1              11
      8                23          3  Life Sciences              1              12
      9                27          3        Medical              1              13

         …  RelationshipSatisfaction StandardHours  StockOptionLevel  \
      0  …                         1            80                 0
      1  …                         4            80                 1
      2  …                         2            80                 0
      3  …                         3            80                 0
      4  …                         4            80                 1
      5  …                         3            80                 0
      6  …                         1            80                 3
      7  …                         2            80                 1
      8  …                         2            80                 0
      9  …                         2            80                 2

         TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
      0                  8                      0               1               6
      1                 10                      3               3              10
      2                  7                      3               3               0
```

```
3                   8                      3               3               8
4                   6                      3               3               2
5                   8                      2               2               7
6                  12                      3               2               1
7                   1                      2               3               1
8                  10                      2               3               9
9                  17                      3               2               7

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                   4                        0                     5
1                   7                        1                     7
2                   0                        0                     0
3                   7                        3                     0
4                   2                        2                     2
5                   7                        3                     6
6                   0                        0                     0
7                   0                        0                     0
8                   7                        1                     8
9                   7                        7                     7

[10 rows x 35 columns]
```

[18]:

```python
# building up a logistic regression model
X = dataframe.drop(['Attrition'],axis=1)
X.head()
Y = dataframe['Attrition']
Y.head()
```

[18]:

```
0    1
1    0
2    1
3    0
4    0
Name: Attrition, dtype: int64
```

[19]:

```python
dataframe['EducationField'].replace('Life Sciences',1, inplace=True)
dataframe['EducationField'].replace('Medical',2, inplace=True)
dataframe['EducationField'].replace('Marketing', 3, inplace=True)
dataframe['EducationField'].replace('Other',4, inplace=True)
dataframe['EducationField'].replace('Technical Degree',5, inplace=True)
dataframe['EducationField'].replace('Human Resources', 6, inplace=True)
```

[20]:

```python
dataframe['EducationField'].value_counts()
```

[20]:

```
1    606
2    464
3    159
```

```
5    132
4     82
6     27
Name: EducationField, dtype: int64
```

[21]:
```
dataframe['Department'].value_counts()
```

[21]:
```
Research & Development    961
Sales                     446
Human Resources            63
Name: Department, dtype: int64
```

[22]:
```
dataframe['Department'].replace('Research & Development',1, inplace=True)
dataframe['Department'].replace('Sales',2, inplace=True)
dataframe['Department'].replace('Human Resources', 3, inplace=True)
```

[23]:
```
dataframe['Department'].value_counts()
```

[23]:
```
1    961
2    446
3     63
Name: Department, dtype: int64
```

[24]:
```
dataframe['MaritalStatus'].value_counts()
```

[24]:
```
Married     673
Single      470
Divorced    327
Name: MaritalStatus, dtype: int64
```

[25]:
```
dataframe['MaritalStatus'].replace('Married',1, inplace=True)
dataframe['MaritalStatus'].replace('Single',2, inplace=True)
dataframe['MaritalStatus'].replace('Divorced',3, inplace=True)
```

[26]:
```
dataframe['MaritalStatus'].value_counts()
```

[26]:
```
1    673
2    470
3    327
Name: MaritalStatus, dtype: int64
```

[27]:
```
x=dataframe.select_dtypes(include=['int64'])
x.dtypes
```

[27]:
```
Age                            int64
Attrition                      int64
DailyRate                      int64
```

```
Department                  int64
DistanceFromHome            int64
Education                   int64
EducationField              int64
EmployeeCount               int64
EmployeeNumber              int64
EnvironmentSatisfaction     int64
HourlyRate                  int64
JobInvolvement              int64
JobLevel                    int64
JobSatisfaction             int64
MaritalStatus               int64
MonthlyIncome               int64
MonthlyRate                 int64
NumCompaniesWorked          int64
PercentSalaryHike           int64
PerformanceRating           int64
RelationshipSatisfaction    int64
StandardHours               int64
StockOptionLevel            int64
TotalWorkingYears           int64
TrainingTimesLastYear       int64
WorkLifeBalance             int64
YearsAtCompany              int64
YearsInCurrentRole          int64
YearsSinceLastPromotion     int64
YearsWithCurrManager        int64
dtype: object
```

[28]:

```
x.columns
```

[28]:

```
Index(['Age', 'Attrition', 'DailyRate', 'Department', 'DistanceFromHome',
       'Education', 'EducationField', 'EmployeeCount', 'EmployeeNumber',
       'EnvironmentSatisfaction', 'HourlyRate', 'JobInvolvement', 'JobLevel',
       'JobSatisfaction', 'MaritalStatus', 'MonthlyIncome', 'MonthlyRate',
       'NumCompaniesWorked', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

[29]:

```
y=dataframe['Attrition']
```

[30]:

```
y.head()
```

[30]:
```
0    1
1    0
2    1
3    0
4    0
Name: Attrition, dtype: int64
```

[31]:
```
y, x = dmatrices('Attrition ~ Age + Department + \
                  DistanceFromHome + Education + EducationField +␣
 ↪YearsAtCompany',
                  dataframe, return_type="dataframe")
print (x.columns)
```

```
Index(['Intercept', 'Age', 'Department', 'DistanceFromHome', 'Education',
       'EducationField', 'YearsAtCompany'],
      dtype='object')
```

[32]:
```
y = np.ravel(y)
```

[33]:
```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model = model.fit(x, y)

# check the accuracy on the training set
model.score(x, y)
```

[33]:
```
0.8408163265306122
```

[34]:
```
y.mean()
```

[34]:
```
0.16122448979591836
```

[35]:
```
X_train,X_test,y_train,y_test=sklearn.model_selection.train_test_split(x,y,␣
 ↪test_size=0.3, random_state=0)
model2=LogisticRegression()
model2.fit(X_train, y_train)
```

[35]:
```
LogisticRegression()
```

[36]:
```
predicted= model2.predict(X_test)
print (predicted)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

[37]:

```
probs = model2.predict_proba(X_test)
print (probs)
```

```
[[0.86179625 0.13820375]
 [0.80754593 0.19245407]
 [0.74123939 0.25876061]
 [0.83441335 0.16558665]
 [0.73499938 0.26500062]
 [0.79097744 0.20902256]
 [0.85615198 0.14384802]
 [0.85699671 0.14300329]
 [0.96699056 0.03300944]
 [0.93685207 0.06314793]
 [0.95099274 0.04900726]
 [0.83101547 0.16898453]
 [0.86296555 0.13703445]
 [0.86581193 0.13418807]
 [0.88750601 0.11249399]
 [0.88892617 0.11107383]
 [0.88569724 0.11430276]
 [0.78516585 0.21483415]
 [0.7979449  0.2020551 ]
 [0.88511301 0.11488699]
 [0.70651596 0.29348404]
 [0.94676691 0.05323309]
 [0.86736255 0.13263745]
 [0.84276454 0.15723546]
 [0.60336851 0.39663149]
 [0.811292   0.188708  ]
 [0.91813729 0.08186271]
 [0.93285521 0.06714479]
 [0.68230761 0.31769239]
 [0.87027136 0.12972864]
```

```
[0.87266384 0.12733616]
[0.7696874  0.2303126 ]
[0.86435769 0.13564231]
[0.95758879 0.04241121]
[0.84461485 0.15538515]
[0.8671935  0.1328065 ]
[0.90465979 0.09534021]
[0.68936427 0.31063573]
[0.90703616 0.09296384]
[0.80663477 0.19336523]
[0.91515724 0.08484276]
[0.82351275 0.17648725]
[0.93711514 0.06288486]
[0.93411324 0.06588676]
[0.89447653 0.10552347]
[0.85317747 0.14682253]
[0.78922388 0.21077612]
[0.84879887 0.15120113]
[0.66402457 0.33597543]
[0.76252297 0.23747703]
[0.92851109 0.07148891]
[0.78953697 0.21046303]
[0.86166595 0.13833405]
[0.85837887 0.14162113]
[0.87217673 0.12782327]
[0.78950898 0.21049102]
[0.87690792 0.12309208]
[0.84165447 0.15834553]
[0.72847153 0.27152847]
[0.83181407 0.16818593]
[0.90095035 0.09904965]
[0.71077332 0.28922668]
[0.92823021 0.07176979]
[0.84375681 0.15624319]
[0.79544108 0.20455892]
[0.86826158 0.13173842]
[0.91679451 0.08320549]
[0.84763056 0.15236944]
[0.89253707 0.10746293]
[0.62872122 0.37127878]
[0.9387539  0.0612461 ]
[0.72620335 0.27379665]
[0.85652974 0.14347026]
[0.84226022 0.15773978]
[0.77436391 0.22563609]
[0.71899557 0.28100443]
[0.93587386 0.06412614]
[0.95710069 0.04289931]
```

```
[0.79185841 0.20814159]
[0.89370438 0.10629562]
[0.91382038 0.08617962]
[0.7935459  0.2064541 ]
[0.77934023 0.22065977]
[0.79638992 0.20361008]
[0.83800496 0.16199504]
[0.71395671 0.28604329]
[0.97772715 0.02227285]
[0.94645972 0.05354028]
[0.88617624 0.11382376]
[0.79620164 0.20379836]
[0.61863836 0.38136164]
[0.81866469 0.18133531]
[0.74504132 0.25495868]
[0.86779495 0.13220505]
[0.87071139 0.12928861]
[0.81717471 0.18282529]
[0.71840764 0.28159236]
[0.59825899 0.40174101]
[0.83951549 0.16048451]
[0.88351325 0.11648675]
[0.74352581 0.25647419]
[0.76631615 0.23368385]
[0.98033036 0.01966964]
[0.91857466 0.08142534]
[0.77432841 0.22567159]
[0.92514814 0.07485186]
[0.88123383 0.11876617]
[0.74587179 0.25412821]
[0.90478361 0.09521639]
[0.78685526 0.21314474]
[0.8114777  0.1885223 ]
[0.93472171 0.06527829]
[0.93836501 0.06163499]
[0.79411745 0.20588255]
[0.81372902 0.18627098]
[0.91610918 0.08389082]
[0.90428343 0.09571657]
[0.84669423 0.15330577]
[0.95384551 0.04615449]
[0.91283691 0.08716309]
[0.85919602 0.14080398]
[0.85902499 0.14097501]
[0.87519518 0.12480482]
[0.7611467  0.2388533 ]
[0.92217685 0.07782315]
[0.96859409 0.03140591]
```

```
[0.94398218 0.05601782]
[0.81780292 0.18219708]
[0.88058703 0.11941297]
[0.77894283 0.22105717]
[0.97124464 0.02875536]
[0.88807661 0.11192339]
[0.78715261 0.21284739]
[0.82001482 0.17998518]
[0.94934542 0.05065458]
[0.95888932 0.04111068]
[0.73559224 0.26440776]
[0.93416995 0.06583005]
[0.73750637 0.26249363]
[0.8213675  0.1786325 ]
[0.821712   0.178288  ]
[0.89896703 0.10103297]
[0.78745762 0.21254238]
[0.89825348 0.10174652]
[0.91433817 0.08566183]
[0.92724744 0.07275256]
[0.96594965 0.03405035]
[0.94417364 0.05582636]
[0.93073081 0.06926919]
[0.66320586 0.33679414]
[0.84168651 0.15831349]
[0.82636817 0.17363183]
[0.80616639 0.19383361]
[0.96157632 0.03842368]
[0.93515159 0.06484841]
[0.94778623 0.05221377]
[0.97337825 0.02662175]
[0.79297057 0.20702943]
[0.87770194 0.12229806]
[0.86103662 0.13896338]
[0.95185478 0.04814522]
[0.93131494 0.06868506]
[0.75685359 0.24314641]
[0.74997802 0.25002198]
[0.95527865 0.04472135]
[0.8695072  0.1304928 ]
[0.81376769 0.18623231]
[0.76990617 0.23009383]
[0.80077911 0.19922089]
[0.9280112  0.0719888 ]
[0.90949245 0.09050755]
[0.94557625 0.05442375]
[0.93330465 0.06669535]
[0.69145909 0.30854091]
```

```
[0.93056881 0.06943119]
[0.74525426 0.25474574]
[0.78594343 0.21405657]
[0.933086   0.066914  ]
[0.8087829  0.1912171 ]
[0.85096457 0.14903543]
[0.66956281 0.33043719]
[0.90338379 0.09661621]
[0.91158931 0.08841069]
[0.87300498 0.12699502]
[0.92939054 0.07060946]
[0.66661726 0.33338274]
[0.89098513 0.10901487]
[0.86216707 0.13783293]
[0.78838554 0.21161446]
[0.53099617 0.46900383]
[0.73344328 0.26655672]
[0.71053983 0.28946017]
[0.85530492 0.14469508]
[0.86960596 0.13039404]
[0.75470006 0.24529994]
[0.89823512 0.10176488]
[0.79247841 0.20752159]
[0.90643139 0.09356861]
[0.77650778 0.22349222]
[0.88387788 0.11612212]
[0.8540431  0.1459569 ]
[0.81899362 0.18100638]
[0.74448519 0.25551481]
[0.86259434 0.13740566]
[0.77822625 0.22177375]
[0.76928325 0.23071675]
[0.79679228 0.20320772]
[0.92096591 0.07903409]
[0.74573771 0.25426229]
[0.87499226 0.12500774]
[0.85503288 0.14496712]
[0.77292156 0.22707844]
[0.87227273 0.12772727]
[0.67395357 0.32604643]
[0.93621663 0.06378337]
[0.82479478 0.17520522]
[0.95146046 0.04853954]
[0.83462451 0.16537549]
[0.81106478 0.18893522]
[0.80828258 0.19171742]
[0.87664152 0.12335848]
[0.66786996 0.33213004]
```

```
[0.59616853 0.40383147]
[0.98979364 0.01020636]
[0.706344   0.293656  ]
[0.91633089 0.08366911]
[0.92133474 0.07866526]
[0.71383909 0.28616091]
[0.62453949 0.37546051]
[0.76440578 0.23559422]
[0.95401652 0.04598348]
[0.88186723 0.11813277]
[0.86019899 0.13980101]
[0.92117379 0.07882621]
[0.88019167 0.11980833]
[0.8045143  0.1954857 ]
[0.80585183 0.19414817]
[0.91345491 0.08654509]
[0.72101811 0.27898189]
[0.94508002 0.05491998]
[0.90906906 0.09093094]
[0.73381751 0.26618249]
[0.98109184 0.01890816]
[0.85490449 0.14509551]
[0.89817069 0.10182931]
[0.82436993 0.17563007]
[0.83387097 0.16612903]
[0.88136645 0.11863355]
[0.88031307 0.11968693]
[0.87560622 0.12439378]
[0.816172   0.183828  ]
[0.88028846 0.11971154]
[0.61530839 0.38469161]
[0.88826156 0.11173844]
[0.89692812 0.10307188]
[0.85328926 0.14671074]
[0.98297041 0.01702959]
[0.77341241 0.22658759]
[0.62184884 0.37815116]
[0.82739135 0.17260865]
[0.84083339 0.15916661]
[0.84808607 0.15191393]
[0.84937028 0.15062972]
[0.75540991 0.24459009]
[0.86188858 0.13811142]
[0.90684688 0.09315312]
[0.84663273 0.15336727]
[0.81290905 0.18709095]
[0.7405358  0.2594642 ]
[0.87035276 0.12964724]
```

```
[0.83938093 0.16061907]
[0.86267949 0.13732051]
[0.66362901 0.33637099]
[0.90671715 0.09328285]
[0.87056296 0.12943704]
[0.9258079  0.0741921 ]
[0.84386854 0.15613146]
[0.89878141 0.10121859]
[0.91239494 0.08760506]
[0.79867634 0.20132366]
[0.64052599 0.35947401]
[0.84683943 0.15316057]
[0.75503197 0.24496803]
[0.85264544 0.14735456]
[0.99251432 0.00748568]
[0.86138215 0.13861785]
[0.8814014  0.1185986 ]
[0.82757086 0.17242914]
[0.93027074 0.06972926]
[0.87307443 0.12692557]
[0.88802396 0.11197604]
[0.8375332  0.1624668 ]
[0.86481031 0.13518969]
[0.86515549 0.13484451]
[0.8975479  0.1024521 ]
[0.78409081 0.21590919]
[0.79165113 0.20834887]
[0.88388123 0.11611877]
[0.65418957 0.34581043]
[0.94059378 0.05940622]
[0.89913435 0.10086565]
[0.72525616 0.27474384]
[0.69085518 0.30914482]
[0.8759163  0.1240837 ]
[0.86562246 0.13437754]
[0.97508876 0.02491124]
[0.86265127 0.13734873]
[0.54036075 0.45963925]
[0.91302274 0.08697726]
[0.74609588 0.25390412]
[0.86758147 0.13241853]
[0.88852756 0.11147244]
[0.87911718 0.12088282]
[0.85664964 0.14335036]
[0.77281961 0.22718039]
[0.80713981 0.19286019]
[0.85285506 0.14714494]
[0.77595544 0.22404456]
```

```
[0.70409841 0.29590159]
[0.88971656 0.11028344]
[0.48963093 0.51036907]
[0.92370649 0.07629351]
[0.75572313 0.24427687]
[0.67788863 0.32211137]
[0.91301004 0.08698996]
[0.94029213 0.05970787]
[0.88009787 0.11990213]
[0.88591167 0.11408833]
[0.95656826 0.04343174]
[0.89909423 0.10090577]
[0.94784647 0.05215353]
[0.83255775 0.16744225]
[0.87822139 0.12177861]
[0.819458   0.180542  ]
[0.81668786 0.18331214]
[0.95036169 0.04963831]
[0.86940138 0.13059862]
[0.90541598 0.09458402]
[0.83661188 0.16338812]
[0.8458365  0.1541635 ]
[0.79307978 0.20692022]
[0.81808124 0.18191876]
[0.81598411 0.18401589]
[0.83677557 0.16322443]
[0.91443033 0.08556967]
[0.91533087 0.08466913]
[0.68503447 0.31496553]
[0.99073758 0.00926242]
[0.76795009 0.23204991]
[0.79871927 0.20128073]
[0.7317683  0.2682317 ]
[0.67465194 0.32534806]
[0.7971246  0.2028754 ]
[0.84935619 0.15064381]
[0.86467084 0.13532916]
[0.85844151 0.14155849]
[0.84533263 0.15466737]
[0.83038801 0.16961199]
[0.92192752 0.07807248]
[0.83189721 0.16810279]
[0.97707551 0.02292449]
[0.90418232 0.09581768]
[0.92691201 0.07308799]
[0.84797357 0.15202643]
[0.76712573 0.23287427]
[0.94895156 0.05104844]
```

```
[0.94782087 0.05217913]
[0.75181508 0.24818492]
[0.8788047  0.1211953 ]
[0.80770784 0.19229216]
[0.93864301 0.06135699]
[0.86008492 0.13991508]
[0.76056702 0.23943298]
[0.90892473 0.09107527]
[0.75402141 0.24597859]
[0.94271842 0.05728158]
[0.9182732  0.0817268 ]
[0.90545128 0.09454872]
[0.76879459 0.23120541]
[0.92358137 0.07641863]
[0.8070588  0.1929412 ]
[0.90079528 0.09920472]
[0.87933329 0.12066671]
[0.80585694 0.19414306]
[0.83150059 0.16849941]
[0.53816776 0.46183224]
[0.95031824 0.04968176]
[0.73291906 0.26708094]
[0.89182872 0.10817128]
[0.80079856 0.19920144]
[0.87739804 0.12260196]
[0.9680523  0.0319477 ]
[0.81741597 0.18258403]
[0.8615009  0.1384991 ]
[0.59407558 0.40592442]
[0.82625732 0.17374268]
[0.92534976 0.07465024]
[0.81692204 0.18307796]
[0.92586783 0.07413217]
[0.890946   0.109054  ]
[0.70071759 0.29928241]
[0.82181894 0.17818106]
[0.9658994  0.0341006 ]
[0.86996989 0.13003011]
[0.89918549 0.10081451]
[0.88983804 0.11016196]
[0.81432462 0.18567538]
[0.85830037 0.14169963]
[0.83878654 0.16121346]
[0.84056579 0.15943421]
[0.8266117  0.1733883 ]
[0.94075167 0.05924833]
[0.83183533 0.16816467]
[0.77561355 0.22438645]
```

```
 [0.69399543 0.30600457]
 [0.85962129 0.14037871]
 [0.82513942 0.17486058]
 [0.84107715 0.15892285]
 [0.87191015 0.12808985]
 [0.89447653 0.10552347]
 [0.82945349 0.17054651]
 [0.72856797 0.27143203]
 [0.94692821 0.05307179]
 [0.96074299 0.03925701]
 [0.90500199 0.09499801]
 [0.88599989 0.11400011]
 [0.84865642 0.15134358]
 [0.79086975 0.20913025]
 [0.6730297  0.3269703 ]
 [0.93390208 0.06609792]
 [0.65644884 0.34355116]
 [0.7438285  0.2561715 ]
 [0.94248413 0.05751587]
 [0.78365674 0.21634326]
 [0.90655666 0.09344334]
 [0.81578727 0.18421273]
 [0.89149736 0.10850264]
 [0.85791873 0.14208127]
 [0.67453857 0.32546143]
 [0.93130395 0.06869605]
 [0.89999984 0.10000016]]
```

[38]:

```python
from sklearn import metrics

print (metrics.accuracy_score(y_test, predicted))
print (metrics.roc_auc_score(y_test, probs[:, 1]))
```

```
0.8435374149659864
0.6502502887947632
```

[39]:

```python
print (metrics.confusion_matrix(y_test, predicted))
print (metrics.classification_report(y_test, predicted))
```

```
[[371   0]
 [ 69   1]]
              precision    recall  f1-score   support

         0.0       0.84      1.00      0.91       371
         1.0       1.00      0.01      0.03        70

    accuracy                           0.84       441
   macro avg       0.92      0.51      0.47       441
```

```
weighted avg       0.87      0.84      0.77       441
```

[40]:
```
print (X_train)
```

```
      Intercept   Age  Department  DistanceFromHome  Education  \
338         1.0  30.0         2.0               5.0        3.0
363         1.0  33.0         2.0               5.0        3.0
759         1.0  45.0         3.0              24.0        4.0
793         1.0  28.0         1.0              15.0        2.0
581         1.0  30.0         1.0               1.0        3.0
…         …   …         …               …        …
763         1.0  34.0         2.0              10.0        4.0
835         1.0  35.0         3.0               8.0        4.0
1216        1.0  43.0         2.0               2.0        3.0
559         1.0  38.0         1.0               2.0        5.0
684         1.0  40.0         2.0              10.0        4.0

      EducationField  YearsAtCompany
338              3.0            10.0
363              3.0             1.0
759              2.0             6.0
793              1.0             4.0
581              1.0             2.0
…              …             …
763              1.0             1.0
835              5.0             5.0
1216             2.0            10.0
559              2.0             1.0
684              3.0             1.0

[1029 rows x 7 columns]
```

[41]:
```
#add random values to KK according to the parameters mentioned above to check␣
 ↪the proabily of attrition of the employee
kk=[[1.0, 23.0, 1.0, 500.0, 3.0, 24.0, 1.0]]
print(model.predict_proba(kk))
```

```
[[6.25572040e-07 9.99999374e-01]]
```

```
/usr/local/lib/python3.7/site-packages/sklearn/base.py:451: UserWarning: X does
not have valid feature names, but LogisticRegression was fitted with feature
names
  "X does not have valid feature names, but"
```

[ ]:
```

```

[ ]:
```

```

[ ]:

[ ]:

[ ]: