# untitled7

September 28, 2023

[1]:
```
'''1.Download the Employee Attrition Dataset
https://www.kaggle.com/datasets/patelprashant/employee-attrition
2.Perfrom Data Preprocessing
3.Model Building using Logistic Regression and Decision Tree and Random Forest
4.Calculate Performance metrics'''
```

[1]:
```
'1.Download the Employee Attrition
Dataset\nhttps://www.kaggle.com/datasets/patelprashant/employee-
attrition\n2.Perfrom Data Preprocessing\n3.Model Building using Logistic
Regression and Decision Tree and Random Forest\n4.Calculate Performance metrics'
```

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[3]:
```
df=pd.read_csv('Employee-Attrition.csv')
df
```

[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
…    …         …                  …        …                       …
1465   36        No  Travel_Frequently        884  Research & Development
1466   39        No      Travel_Rarely        613  Research & Development
1467   27        No      Travel_Rarely        155  Research & Development
1468   49        No  Travel_Frequently       1023                   Sales
1469   34        No      Travel_Rarely        628  Research & Development

      DistanceFromHome  Education EducationField  EmployeeCount  \
0                    1          2  Life Sciences              1
1                    8          1  Life Sciences              1
2                    2          2          Other              1
3                    3          4  Life Sciences              1
```

```
4                   2          1        Medical              1
…                 …        …           …            …
1465               23          2        Medical              1
1466                6          1        Medical              1
1467                4          3  Life Sciences              1
1468                2          3        Medical              1
1469                8          3        Medical              1

      EmployeeNumber  …  RelationshipSatisfaction StandardHours  \
0                  1  …                         1            80
1                  2  …                         4            80
2                  4  …                         2            80
3                  5  …                         3            80
4                  7  …                         4            80
…                …  …                       …           …
1465            2061  …                         3            80
1466            2062  …                         1            80
1467            2064  …                         2            80
1468            2065  …                         4            80
1469            2068  …                         1            80

      StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
0                    0                  8                      0
1                    1                 10                      3
2                    0                  7                      3
3                    0                  8                      3
4                    1                  6                      3
…                 …               …                    …
1465                 1                 17                      3
1466                 1                  9                      5
1467                 1                  6                      0
1468                 0                 17                      3
1469                 0                  6                      3

     WorkLifeBalance  YearsAtCompany YearsInCurrentRole  \
0                  1               6                  4
1                  3              10                  7
2                  3               0                  0
3                  3               8                  7
4                  3               2                  2
…                …             …                 …
1465               3               5                  2
1466               3               7                  7
1467               3               6                  2
1468               2               9                  6
1469               4               4                  3
```

```
      YearsSinceLastPromotion  YearsWithCurrManager
0                           0                     5
1                           1                     7
2                           0                     0
3                           3                     0
4                           2                     2
…                         …                     …
1465                        0                     3
1466                        1                     7
1467                        0                     3
1468                        0                     8
1469                        1                     2

[1470 rows x 35 columns]
```

[4]:

```
df.isnull().sum()
```

[4]:

```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
```

```
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

[5]:
```python
df.head()
```

[5]:
```
   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]
```

[6]:
```python
df.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
```

[7]:
```
df.hist(figsize=(15,15))
```

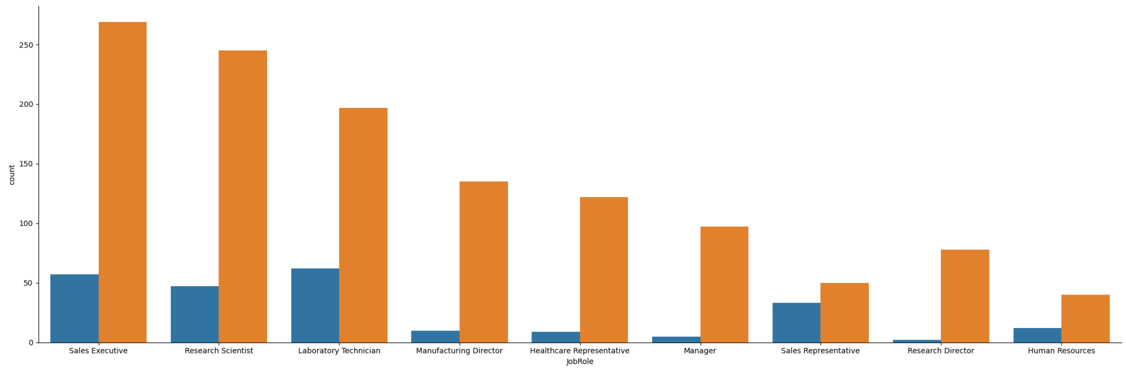
[7]:
```
array([[<Axes: title={'center': 'Age'}>,
        <Axes: title={'center': 'DailyRate'}>,
```

```
 <Axes: title={'center': 'DistanceFromHome'}>,
 <Axes: title={'center': 'Education'}>,
 <Axes: title={'center': 'EmployeeCount'}>],
[<Axes: title={'center': 'EmployeeNumber'}>,
 <Axes: title={'center': 'EnvironmentSatisfaction'}>,
 <Axes: title={'center': 'HourlyRate'}>,
 <Axes: title={'center': 'JobInvolvement'}>,
 <Axes: title={'center': 'JobLevel'}>],
[<Axes: title={'center': 'JobSatisfaction'}>,
 <Axes: title={'center': 'MonthlyIncome'}>,
 <Axes: title={'center': 'MonthlyRate'}>,
 <Axes: title={'center': 'NumCompaniesWorked'}>,
 <Axes: title={'center': 'PercentSalaryHike'}>],
[<Axes: title={'center': 'PerformanceRating'}>,
 <Axes: title={'center': 'RelationshipSatisfaction'}>,
 <Axes: title={'center': 'StandardHours'}>,
 <Axes: title={'center': 'StockOptionLevel'}>,
 <Axes: title={'center': 'TotalWorkingYears'}>],
[<Axes: title={'center': 'TrainingTimesLastYear'}>,
 <Axes: title={'center': 'WorkLifeBalance'}>,
 <Axes: title={'center': 'YearsAtCompany'}>,
 <Axes: title={'center': 'YearsInCurrentRole'}>,
 <Axes: title={'center': 'YearsSinceLastPromotion'}>],
[<Axes: title={'center': 'YearsWithCurrManager'}>, <Axes: >,
 <Axes: >, <Axes: >, <Axes: >]], dtype=object)
```


[8]:
```
#inference: Monthly Income, Total Working hours, Years at company, Distance␣
↪from home are all righ skewed.
#inference: Employee count and Standard hours are redundant so they can be␣
↪removed.
```

[9]:
```
sns.
↪catplot(x='JobRole',hue='Attrition',data=df,kind='count',height=7,aspect=3,legend=False)
```

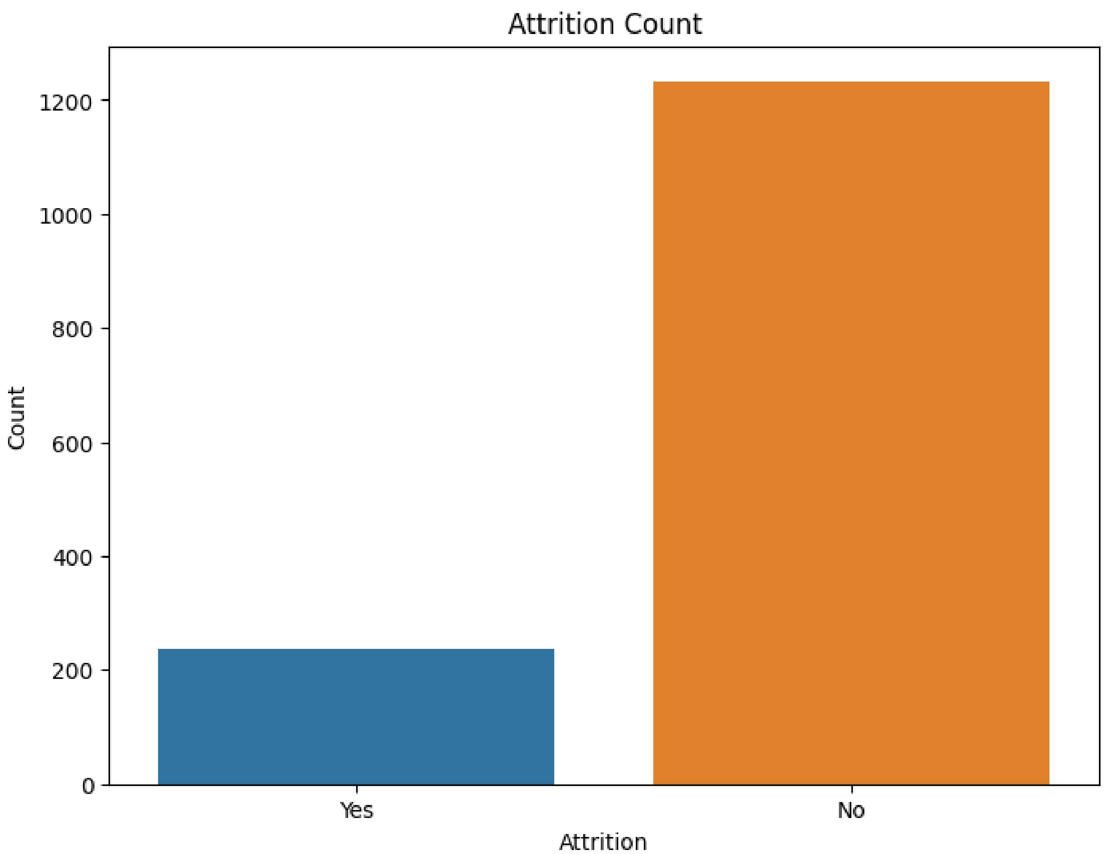
[9]: <seaborn.axisgrid.FacetGrid at 0x7ddce526b280>




[10]:
```
corr=df.corr()
```

```
<ipython-input-10-0014364bc22a>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  corr=df.corr()
```

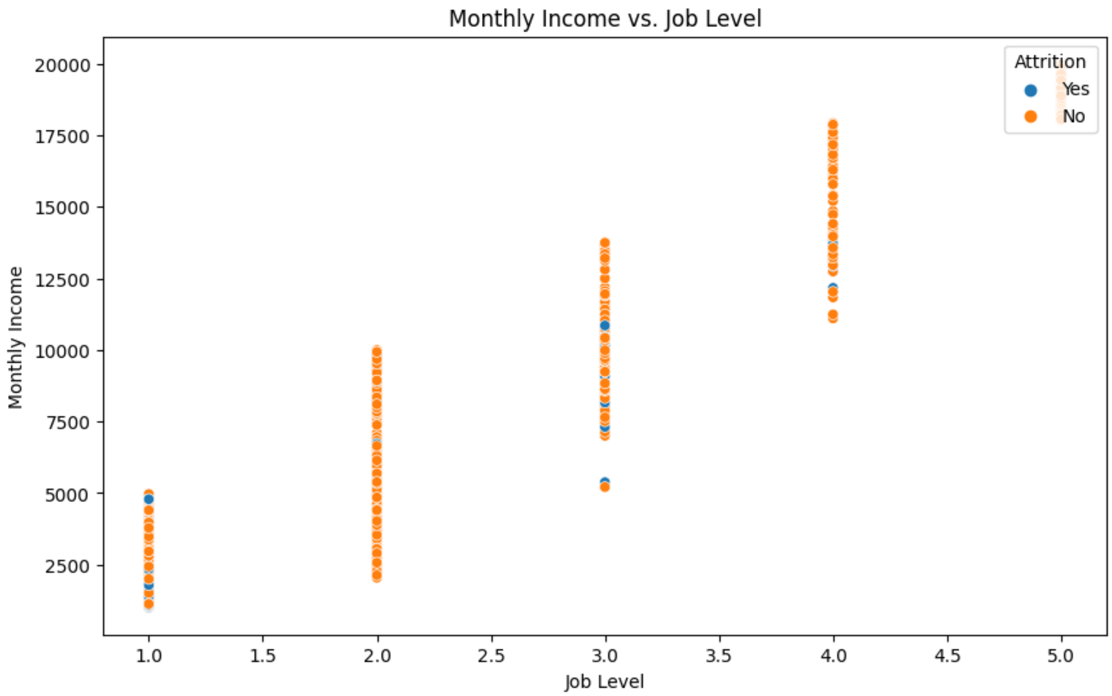
[11]:
```
plt.figure(figsize=(8, 6))
sns.countplot(x='Attrition', data=df)
plt.title('Attrition Count')
plt.xlabel('Attrition')
plt.ylabel('Count')
plt.show()
```




[12]:
```
#inference: The plot suggests that the attrition is relatively less common in␣
↪dataset
```

[13]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='JobLevel', y='MonthlyIncome', hue='Attrition', data=df)
plt.title('Monthly Income vs. Job Level')
plt.xlabel('Job Level')
plt.ylabel('Monthly Income')
plt.legend(title='Attrition', loc='upper right')
```

[13]: <matplotlib.legend.Legend at 0x7ddca1c91c90>

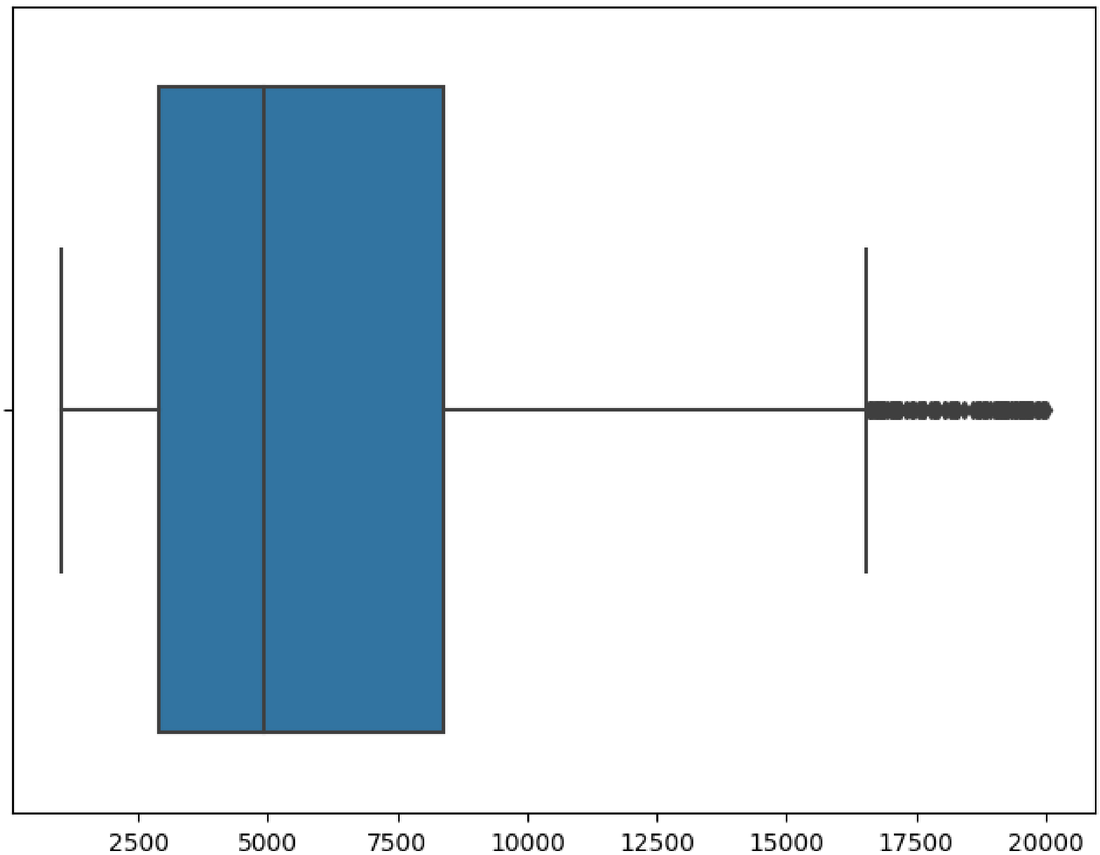



[14]:
```
#inference: As the job level increases the monthly income also increases
```

[15]:
```
sns.heatmap(corr,vmax=0.5,linewidth=0.2)
```

[15]: <Axes: >


[16]:
```
plt.figure(figsize=(8, 6))
sns.boxplot(x=df['MonthlyIncome'])
plt.xlabel('')
plt.show()
```




[17]:
```
Q1 = df['MonthlyIncome'].quantile(0.25)
Q3 = df['MonthlyIncome'].quantile(0.75)
IQR = Q3 - Q1
outliers = (df['MonthlyIncome'] < Q1 - 1.5 * IQR) | (df['MonthlyIncome'] > Q3 +
 ↪1.5 * IQR)
outlier_rows = df[outliers]
print(outlier_rows)
```

```
      Age Attrition     BusinessTravel  DailyRate              Department  \
25     53        No      Travel_Rarely       1282  Research & Development
29     46        No      Travel_Rarely        705                   Sales
45     41       Yes      Travel_Rarely       1360  Research & Development
62     50        No      Travel_Rarely        989  Research & Development
105    59        No         Non-Travel       1420         Human Resources
…     …         …                …        …                     …
1374   58        No      Travel_Rarely        605                   Sales
1377   49        No  Travel_Frequently       1064  Research & Development
1401   55        No      Travel_Rarely        189         Human Resources
1437   39        No         Non-Travel        105  Research & Development
1443   42        No      Travel_Rarely        300  Research & Development
```

```
      DistanceFromHome  Education    EducationField  EmployeeCount  \
25                   5          3             Other              1
29                   2          4         Marketing              1
45                  12          3  Technical Degree              1
62                   7          2           Medical              1
105                  2          4   Human Resources              1
…                  …        …               …             …
1374                21          3     Life Sciences              1
1377                 2          1     Life Sciences              1
1401                26          4   Human Resources              1
1437                 9          3     Life Sciences              1
1443                 2          3     Life Sciences              1

      EmployeeNumber  …  RelationshipSatisfaction StandardHours  \
25                32  …                         4            80
29                38  …                         4            80
45                58  …                         4            80
62                80  …                         4            80
105              140  …                         4            80
…                …  …                       …           …
1374            1938  …                         3            80
1377            1941  …                         4            80
1401            1973  …                         1            80
1437            2022  …                         3            80
1443            2031  …                         1            80

      StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
25                   1                 26                      3
29                   0                 22                      2
45                   0                 23                      0
62                   1                 29                      2
105                  1                 30                      3
…                  …                …                    …
1374                 1                 29                      2
1377                 0                 28                      3
1401                 1                 35                      0
1437                 0                 21                      3
1443                 0                 24                      2

     WorkLifeBalance  YearsAtCompany YearsInCurrentRole  \
25                 2              14                 13
29                 2               2                  2
45                 3              22                 15
62                 2              27                  3
105                3               3                  2
…                …             …                …
1374               2               1                  0
```

```
1377                3             5                   4
1401                3            10                   9
1437                2             6                   0
1443                2            22                   6

      YearsSinceLastPromotion  YearsWithCurrManager
25                          4                     8
29                          2                     1
45                         15                     8
62                         13                     8
105                         2                     2
…                         …                     …
1374                        0                     0
1377                        4                     3
1401                        1                     4
1437                        1                     3
1443                        4                    14

[114 rows x 35 columns]
```

[18]:
```
df= pd.get_dummies(df, columns=['BusinessTravel', 'Department',␣
↪'EducationField', 'Gender', 'JobRole', 'MaritalStatus', 'Over18',␣
↪'OverTime'])
```

[19]:
```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
df['Attrition']=lb.fit_transform(df['Attrition'])
```

[20]:
```
x=df.drop('Attrition',axis=1)
y=df['Attrition']
```

[21]:
```
x.head()
```

[21]:
```
   Age  DailyRate  DistanceFromHome  Education  EmployeeCount  EmployeeNumber  \
0   41       1102                 1          2              1               1
1   49        279                 8          1              1               2
2   37       1373                 2          2              1               4
3   33       1392                 3          4              1               5
4   27        591                 2          1              1               7

   EnvironmentSatisfaction  HourlyRate  JobInvolvement  JobLevel  …  \
0                        2          94               3         2  …
1                        3          61               2         2  …
2                        4          92               2         1  …
3                        4          56               3         1  …
4                        1          40               3         1  …
```

```
   JobRole_Research Director  JobRole_Research Scientist  \
0                          0                           0
1                          0                           1
2                          0                           0
3                          0                           1
4                          0                           0

   JobRole_Sales Executive  JobRole_Sales Representative  \
0                        1                             0
1                        0                             0
2                        0                             0
3                        0                             0
4                        0                             0

   MaritalStatus_Divorced  MaritalStatus_Married  MaritalStatus_Single  \
0                       0                      0                     1
1                       0                      1                     0
2                       0                      0                     1
3                       0                      1                     0
4                       0                      1                     0

   Over18_Y  OverTime_No  OverTime_Yes
0         1            0             1
1         1            1             0
2         1            0             1
3         1            0             1
4         1            1             0

[5 rows x 55 columns]
```

[22]:
```python
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

[23]:
```python
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

[23]:
```
((1176, 55), (294, 55), (1176,), (294,))
```

[24]:
```python
from sklearn.preprocessing import MinMaxScaler
sc=MinMaxScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
x_train,x_test
```

[24]:
```
(array([[0.95238095, 0.35913978, 0.71428571, …, 0.        , 0.        ,
         1.        ],
        [0.64285714, 0.60645161, 0.96428571, …, 0.        , 1.        ,
         0.        ],
```

```
        [0.52380952, 0.14050179, 0.89285714, …, 0.        , 1.        ,
         0.        ],
        …,
        [0.5952381 , 0.77060932, 0.03571429, …, 0.        , 0.        ,
         1.        ],
        [0.47619048, 0.11756272, 0.03571429, …, 0.        , 0.        ,
         1.        ],
        [0.52380952, 0.39713262, 0.32142857, …, 0.        , 1.        ,
         0.        ]]),
 array([[0.42857143, 0.38064516, 0.32142857, …, 0.        , 1.        ,
         0.        ],
        [0.35714286, 0.33763441, 0.85714286, …, 0.        , 1.        ,
         0.        ],
        [0.4047619 , 0.4       , 0.60714286, …, 0.        , 0.        ,
         1.        ],
        …,
        [0.30952381, 0.10394265, 0.17857143, …, 0.        , 0.        ,
         1.        ],
        [0.47619048, 0.82939068, 0.03571429, …, 0.        , 1.        ,
         0.        ],
        [0.52380952, 0.18996416, 0.25      , …, 0.        , 0.        ,
         1.        ]]))
```

[45]:
```python
from sklearn.linear_model import LogisticRegression
x_train_logistic=x_train
y_train_logistic=y_train
x_test_logistic=x_test
y_test_logistic=y_test
model=LogisticRegression()
model.fit(x_train_logistic,y_train_logistic)
```

[45]:
```
LogisticRegression()
```

[46]:
```python
pred_logistic=model.predict(x_test_logistic)
pred_logistic
```

[46]:
```
array([0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 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, 1, 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, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 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, 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, 1, 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, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0])
```

[47]:
```
y_test_logistic
```

[47]:
```
442     0
1091    0
981     1
785     0
1332    1
       ..
1439    0
481     0
124     1
198     0
1229    0
Name: Attrition, Length: 294, dtype: int64
```

[48]:
```
from sklearn.metrics import␣
 ↪accuracy_score,confusion_matrix,classification_report,roc_auc_score,roc_curve,precision_sco
accuracy_logistic = accuracy_score(y_test_logistic, pred_logistic)
print("Accuracy:", accuracy_logistic)
precision_logistic = precision_score(y_test_logistic, pred_logistic)
print("Precision:", precision_logistic)
recall_logistic = recall_score(y_test_logistic, pred_logistic)
print("Recall:", recall_logistic)
f1_logistic = f1_score(y_test_logistic, pred_logistic)
print("F1 Score:", f1_logistic)
roc_auc_logistic = roc_auc_score(y_test_logistic, pred_logistic)
print("ROC AUC Score:", roc_auc_logistic)
```

```
Accuracy: 0.8877551020408163
Precision: 0.8076923076923077
Recall: 0.42857142857142855
F1 Score: 0.5599999999999999
ROC AUC Score: 0.7040816326530612
```

[49]:
```
print(classification_report(y_test_logistic,pred_logistic))
```

```
              precision    recall  f1-score   support

           0       0.90      0.98      0.94       245
           1       0.81      0.43      0.56        49

    accuracy                           0.89       294
   macro avg       0.85      0.70      0.75       294
```

```
weighted avg       0.88      0.89      0.87       294
```

[50]:
```
from sklearn.tree import DecisionTreeClassifier
x_train_dtc=x_train
y_train_dtc=y_train
x_test_dtc=x_test
y_test_dtc=y_test
dtc=DecisionTreeClassifier()
dtc.fit(x_train_dtc,y_train_dtc)
```

[50]:
```
DecisionTreeClassifier()
```

[52]:
```
pred_dtc=dtc.predict(x_test_dtc)
pred_dtc
```

[52]:
```
array([0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 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, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 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, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 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, 1, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 0, 0, 0, 0, 0])
```

[53]:
```
y_test_dtc
```

[53]:
```
442     0
1091    0
981     1
785     0
1332    1
       ..
1439    0
481     0
124     1
198     0
1229    0
Name: Attrition, Length: 294, dtype: int64
```

[54]:
```
accuracy_dtc = accuracy_score(y_test_dtc, pred_dtc)
print("Accuracy:", accuracy_dtc)
precision_dtc = precision_score(y_test_dtc, pred_dtc)
print("Precision:", precision_dtc)
recall_dtc = recall_score(y_test_dtc, pred_dtc)
print("Recall:", recall_dtc)
f1_dtc = f1_score(y_test_dtc, pred_dtc)
print("F1 Score:", f1_dtc)
roc_auc_dtc = roc_auc_score(y_test_dtc, pred_dtc)
print("ROC AUC Score:", roc_auc_dtc)
```

```
Accuracy: 0.7755102040816326
Precision: 0.3333333333333333
Recall: 0.3469387755102041
F1 Score: 0.33999999999999997
ROC AUC Score: 0.6040816326530611
```

[55]:
```
print(classification_report(y_test_dtc,pred_dtc))
```

```
              precision    recall  f1-score   support

           0       0.87      0.86      0.86       245
           1       0.33      0.35      0.34        49

    accuracy                           0.78       294
   macro avg       0.60      0.60      0.60       294
weighted avg       0.78      0.78      0.78       294
```

[35]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
```

[36]:
```
param_grid = {
    'solver':['newton-cg','liblinear'],
    'penalty':['l2'],
    'C':np.logspace(-4.5,4.5,50),
    'class_weight':['balanced'],
    'tol':[0.0001,0.001,0.01,0.1],
    'fit_intercept':[True,False],
    'intercept_scaling':[1,2,3]
}
```

[56]:
```
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=5,␣
 ↪scoring='roc_auc', n_jobs=-1)
grid_search.fit(x_train_logistic, y_train_logistic)
```

[56]:
```
GridSearchCV(cv=5, estimator=LogisticRegression(), n_jobs=-1,
             param_grid={'C': array([3.16227766e-05, 4.82695744e-05,
7.36795456e-05, 1.12465782e-04,
       1.71669791e-04, 2.62039853e-04, 3.99982340e-04, 6.10540230e-04,
       9.31939576e-04, 1.42252931e-03, 2.17137430e-03, 3.31442475e-03,
       5.05919749e-03, 7.72244995e-03, 1.17876863e-02, 1.79929362e-02,
       2.74647411e-02, 4.19226744e-0…
       1.29492584e+02, 1.97659807e+02, 3.01711481e+02, 4.60537826e+02,
       7.02973212e+02, 1.07303094e+03, 1.63789371e+03, 2.50011038e+03,
       3.81621341e+03, 5.82513671e+03, 8.89159334e+03, 1.35722878e+04,
       2.07169840e+04, 3.16227766e+04]),
                         'class_weight': ['balanced'],
                         'fit_intercept': [True, False],
                         'intercept_scaling': [1, 2, 3], 'penalty': ['l2'],
                         'solver': ['newton-cg', 'liblinear'],
                         'tol': [0.0001, 0.001, 0.01, 0.1]},
             scoring='roc_auc')
```

[57]:
```python
grid_search.fit(x_train_logistic,y_train_logistic)
```

[57]:
```
GridSearchCV(cv=5, estimator=LogisticRegression(), n_jobs=-1,
             param_grid={'C': array([3.16227766e-05, 4.82695744e-05,
7.36795456e-05, 1.12465782e-04,
       1.71669791e-04, 2.62039853e-04, 3.99982340e-04, 6.10540230e-04,
       9.31939576e-04, 1.42252931e-03, 2.17137430e-03, 3.31442475e-03,
       5.05919749e-03, 7.72244995e-03, 1.17876863e-02, 1.79929362e-02,
       2.74647411e-02, 4.19226744e-0…
       1.29492584e+02, 1.97659807e+02, 3.01711481e+02, 4.60537826e+02,
       7.02973212e+02, 1.07303094e+03, 1.63789371e+03, 2.50011038e+03,
       3.81621341e+03, 5.82513671e+03, 8.89159334e+03, 1.35722878e+04,
       2.07169840e+04, 3.16227766e+04]),
                         'class_weight': ['balanced'],
                         'fit_intercept': [True, False],
                         'intercept_scaling': [1, 2, 3], 'penalty': ['l2'],
                         'solver': ['newton-cg', 'liblinear'],
                         'tol': [0.0001, 0.001, 0.01, 0.1]},
             scoring='roc_auc')
```

[58]:
```python
best_params = grid_search.best_params_
best_model = grid_search.best_estimator_
```

[59]:
```python
print(classification_report(y_test_logistic,pred_logistic))
```

```
              precision    recall  f1-score   support

           0       0.90      0.98      0.94       245
           1       0.81      0.43      0.56        49
```

```
    accuracy                           0.89       294
   macro avg       0.85      0.70      0.75       294
weighted avg       0.88      0.89      0.87       294
```

[61]:
```
accuracy_logistic = accuracy_score(y_test_logistic, pred_logistic)
print("Accuracy:", accuracy_logistic)
precision_logistic = precision_score(y_test_logistic, pred_logistic)
print("Precision:", precision_logistic)
recall_logistic = recall_score(y_test_logistic, pred_logistic)
print("Recall:", recall_logistic)
f1_logistic = f1_score(y_test_logistic, pred_logistic)
print("F1 Score:", f1_logistic)
roc_auc_logistic = roc_auc_score(y_test_logistic, pred_logistic)
print("ROC AUC Score:", roc_auc_logistic)
```

```
Accuracy: 0.8877551020408163
Precision: 0.8076923076923077
Recall: 0.42857142857142855
F1 Score: 0.5599999999999999
ROC AUC Score: 0.7040816326530612
```

[42]:
```
#inference: There is a class imbalance-Attrition=1
```