Shaik Sharuk 21BCE9523 Assignment-4

1.Importing Libraries ¶

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

2.Import Dataset

In [2]:
```
df = pd.read_csv("WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

In [3]:
```
df.head()
```

Out[3]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educatio
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sc
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sc
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sc
4	27	No	Travel_Rarely	591	Research & Development	2	1	M

5 rows × 35 columns

In [4]:
```
df.tail()
```

Out[4]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educ
1465	36	No	Travel_Frequently	884	Research & Development	23	2	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Lif
1468	49	No	Travel_Frequently	1023	Sales	2	3	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	

5 rows × 35 columns

In [5]:
```
df.shape
```

Out[5]: (1470, 35)

5 rows × 35 columns

In [5]:
```
df.shape
```

Out[5]:
```
(1470, 35)
```

In [6]:
```
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
```

In [7]:
```
df.describe()
```

Out[7]:

In [7]:
```
df.describe()
```

Out[7]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumbe
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.00000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.86530
std	9.135373	403.509100	8.106864	1.024165	0.0	602.02433
min	18.000000	102.000000	1.000000	1.000000	1.0	1.00000
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.25000
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.50000
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.75000
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.00000

8 rows × 26 columns

In [8]:
```
df.columns
```

Out[8]:
```
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')
```

3.Handling Null Values

3.Handling Null Values

In [9]:
```
df.isnull().any()
```

Out[9]:
```
Age                         False
Attrition                   False
BusinessTravel              False
DailyRate                   False
Department                  False
DistanceFromHome            False
Education                   False
EducationField              False
EmployeeCount               False
EmployeeNumber              False
EnvironmentSatisfaction     False
Gender                      False
HourlyRate                  False
JobInvolvement              False
JobLevel                    False
JobRole                     False
JobSatisfaction             False
MaritalStatus               False
MonthlyIncome               False
MonthlyRate                 False
NumCompaniesWorked          False
Over18                      False
OverTime                    False
PercentSalaryHike           False
PerformanceRating           False
RelationshipSatisfaction    False
StandardHours               False
StockOptionLevel            False
TotalWorkingYears           False
TrainingTimesLastYear       False
WorkLifeBalance             False
YearsAtCompany              False
YearsInCurrentRole          False
YearsSinceLastPromotion     False
YearsWithCurrManager        False
dtype: bool
```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
Age                         0
```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
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
```

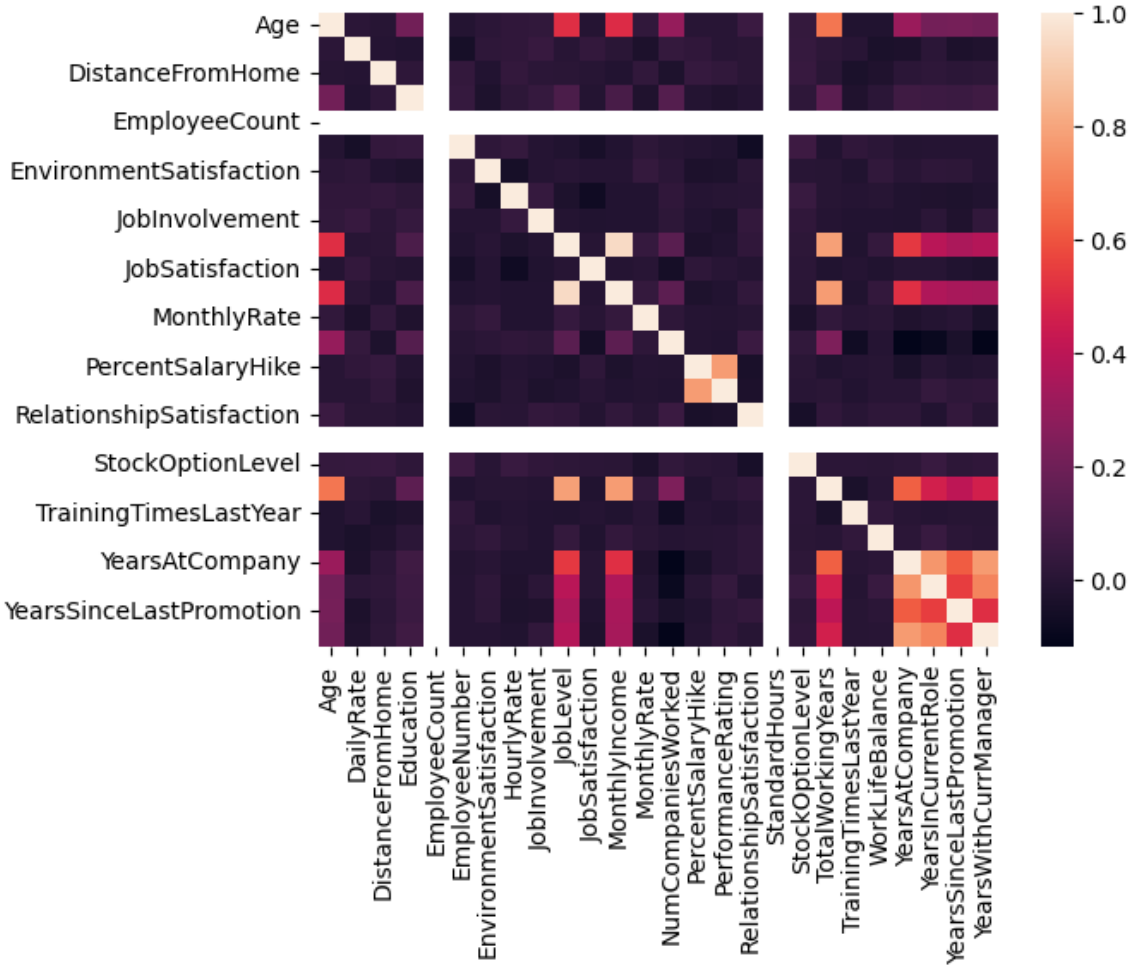
4.Data Visualization

4.Data Visualization

In [11]:
```
sns.heatmap(df.corr())
```

```
C:\Users\shaik\AppData\Local\Temp\ipykernel_16268\58359773.py: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 val
ue of numeric_only to silence this warning.
  sns.heatmap(df.corr())
```

Out[11]: <Axes: >



In [12]:
```
df.head()
```

Out[12]:

In [12]:
```
df.head()
```

Out[12]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educatio
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sc
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sc
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sc
4	27	No	Travel_Rarely	591	Research & Development	2	1	M

5 rows × 35 columns

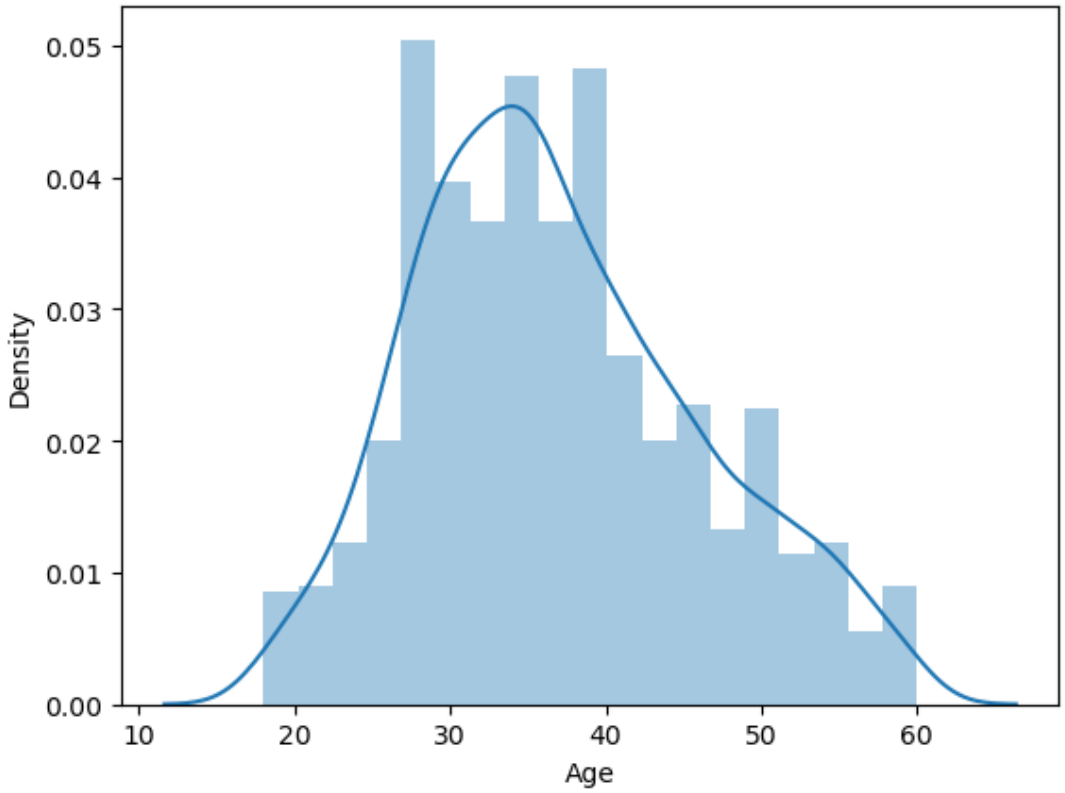
In [13]:
```
sns.distplot(df["Age"])
```

```
C:\Users\shaik\AppData\Local\Temp\ipykernel_16268\2732350774.py:1: UserWarning:
```

In [13]:
```
sns.distplot(df["Age"])
```

```
C:\Users\shaik\AppData\Local\Temp\ipykernel_16268\2732350774.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.
github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df["Age"])
```

Out[13]: <Axes: xlabel='Age', ylabel='Density'>

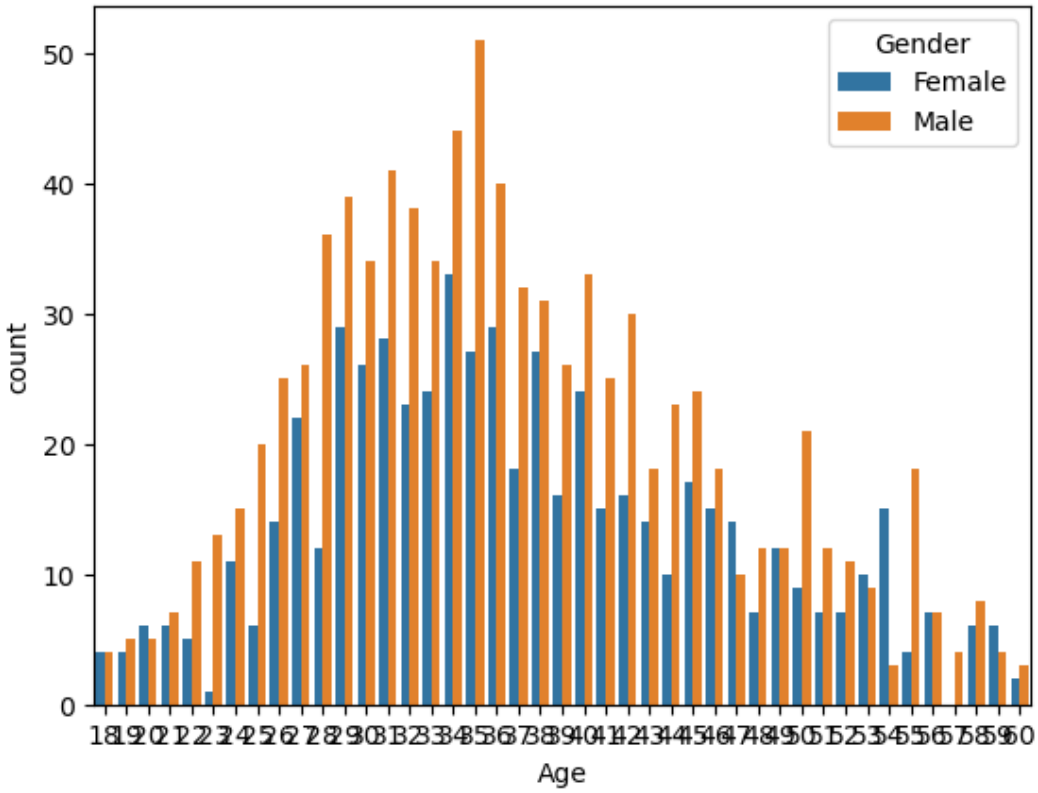


In [14]:
```
sns.countplot(x="Age",data=df,hue="Gender")
```

Out[14]: <Axes: xlabel='Age', ylabel='count'>

In [14]:
```
sns.countplot(x="Age",data=df,hue="Gender")
```

Out[14]:
```
<Axes: xlabel='Age', ylabel='count'>
```

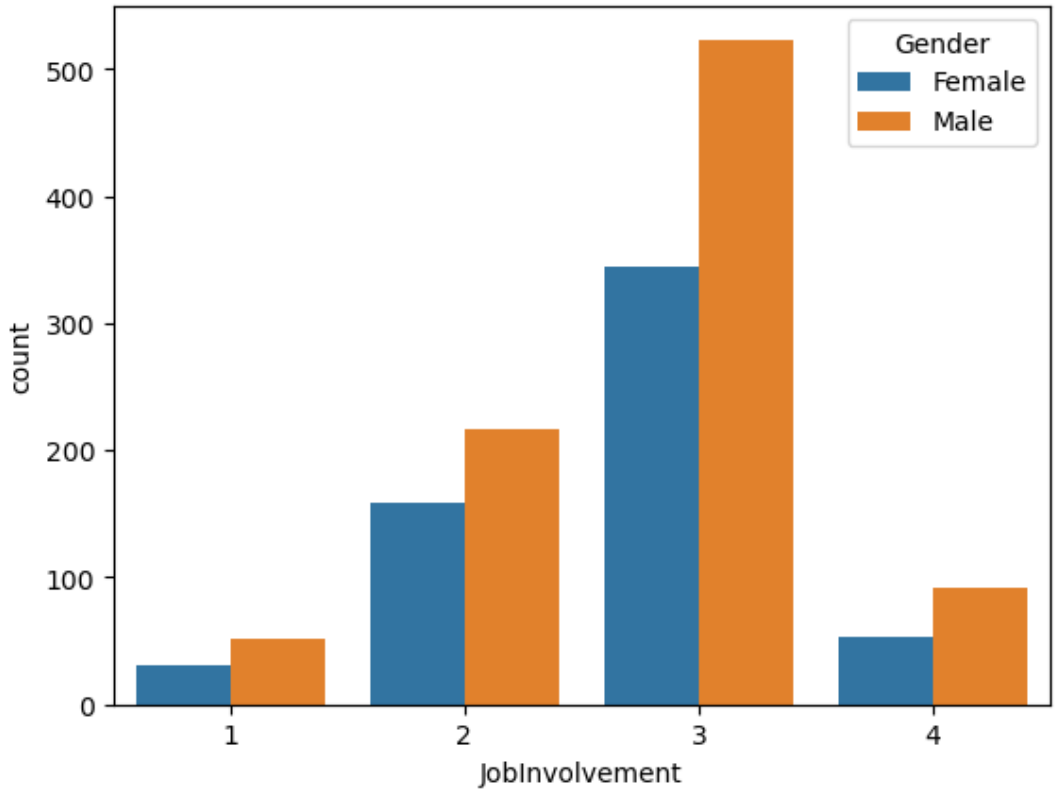


In [15]:
```
sns.countplot(x="JobInvolvement",data=df,hue="Gender")
```

Out[15]:
```
<Axes: xlabel='JobInvolvement', ylabel='count'>
```

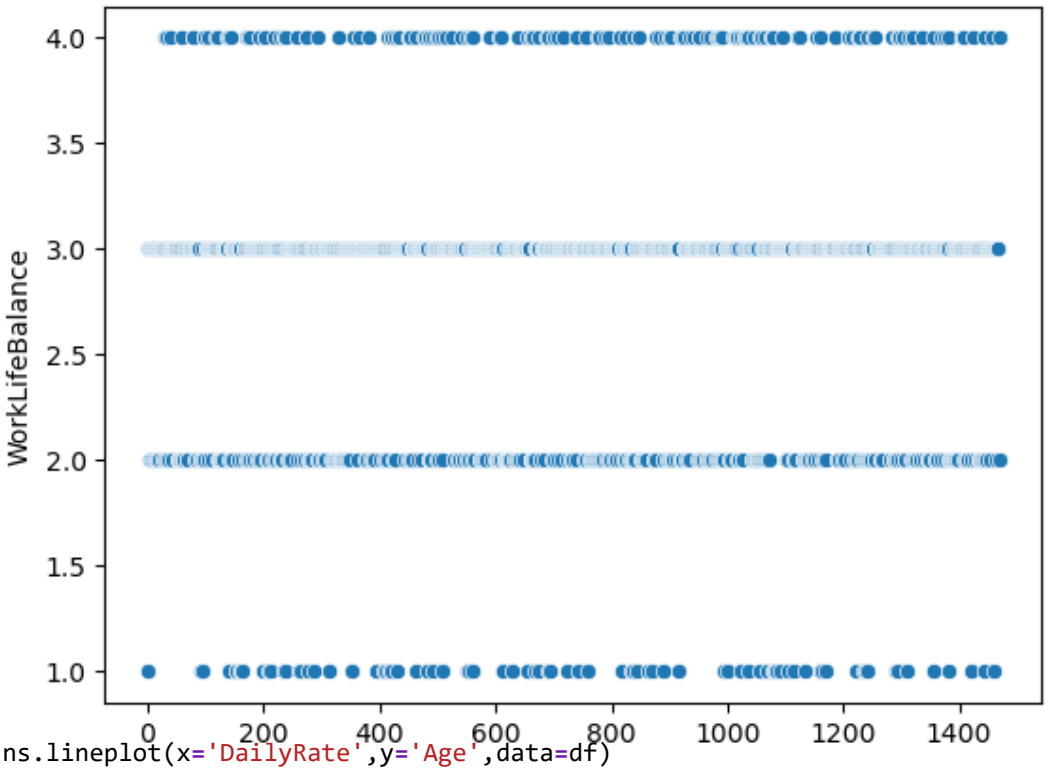
In [15]:
```
sns.countplot(x="JobInvolvement",data=df,hue="Gender")
```

Out[15]: <Axes: xlabel='JobInvolvement', ylabel='count'>



In [16]:
```
sns.scatterplot(df['WorkLifeBalance'])
```

Out[16]: <Axes: ylabel='WorkLifeBalance'>

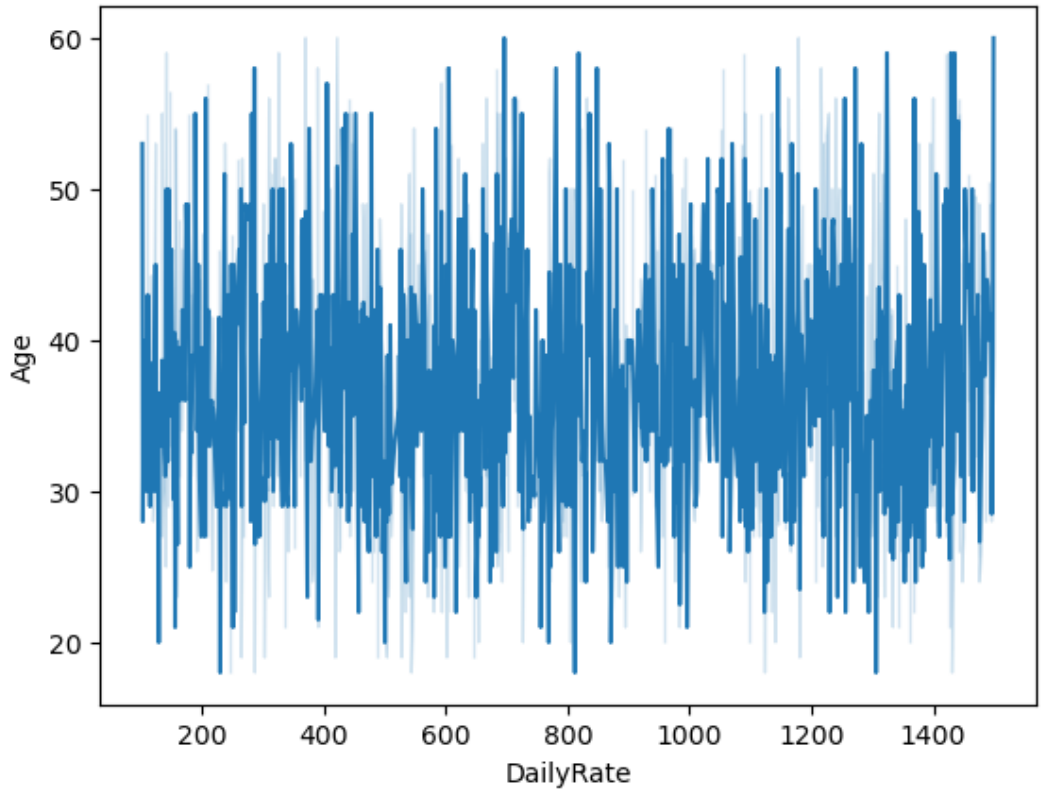


In [17]:
```
sns.lineplot(x='DailyRate',y='Age',data=df)
```

Out[17]: <Axes: xlabel='DailyRate', ylabel='Age'>

In [17]:
```
sns.lineplot(x='DailyRate',y='Age',data=df)
```

Out[17]: <Axes: xlabel='DailyRate', ylabel='Age'>



In [18]:
```
df.corr()
```

```
C:\Users\shaik\AppData\Local\Temp\ipykernel_16268\1134722465.py:1: FutureWarnin
```

In [18]:
```
df.corr()
```

```
C:\Users\shaik\AppData\Local\Temp\ipykernel_16268\1134722465.py:1: FutureWarnin
g: The default value of numeric_only in DataFrame.corr is deprecated. In a futu
re version, it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  df.corr()
```

Out[18]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	Emp
Age	1.000000	0.010661	-0.001686	0.208034	NaN	
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN	
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	NaN	
Education	0.208034	-0.016806	0.021042	1.000000	NaN	
EmployeeCount	NaN	NaN	NaN	NaN	NaN	
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070	NaN	
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	NaN	
HourlyRate	0.024287	0.023381	0.031131	0.016775	NaN	
JobInvolvement	0.029820	0.046135	0.008783	0.042438	NaN	
JobLevel	0.509604	0.002966	0.005303	0.101589	NaN	
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	NaN	
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	NaN	
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	NaN	
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	NaN	
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	NaN	
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	NaN	
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	NaN	
StandardHours	NaN	NaN	NaN	NaN	NaN	
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	NaN	
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280	NaN	
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100	NaN	
WorkLifeBalance	-0.021490	-0.037848	-0.026556	0.009819	NaN	
YearsAtCompany	0.311309	-0.034055	0.009508	0.069114	NaN	
YearsInCurrentRole	0.212901	0.009932	0.018845	0.060236	NaN	
YearsSinceLastPromotion	0.216513	-0.033229	0.010029	0.054254	NaN	
YearsWithCurrManager	0.202089	-0.026363	0.014406	0.069065	NaN	

26 rows × 26 columns

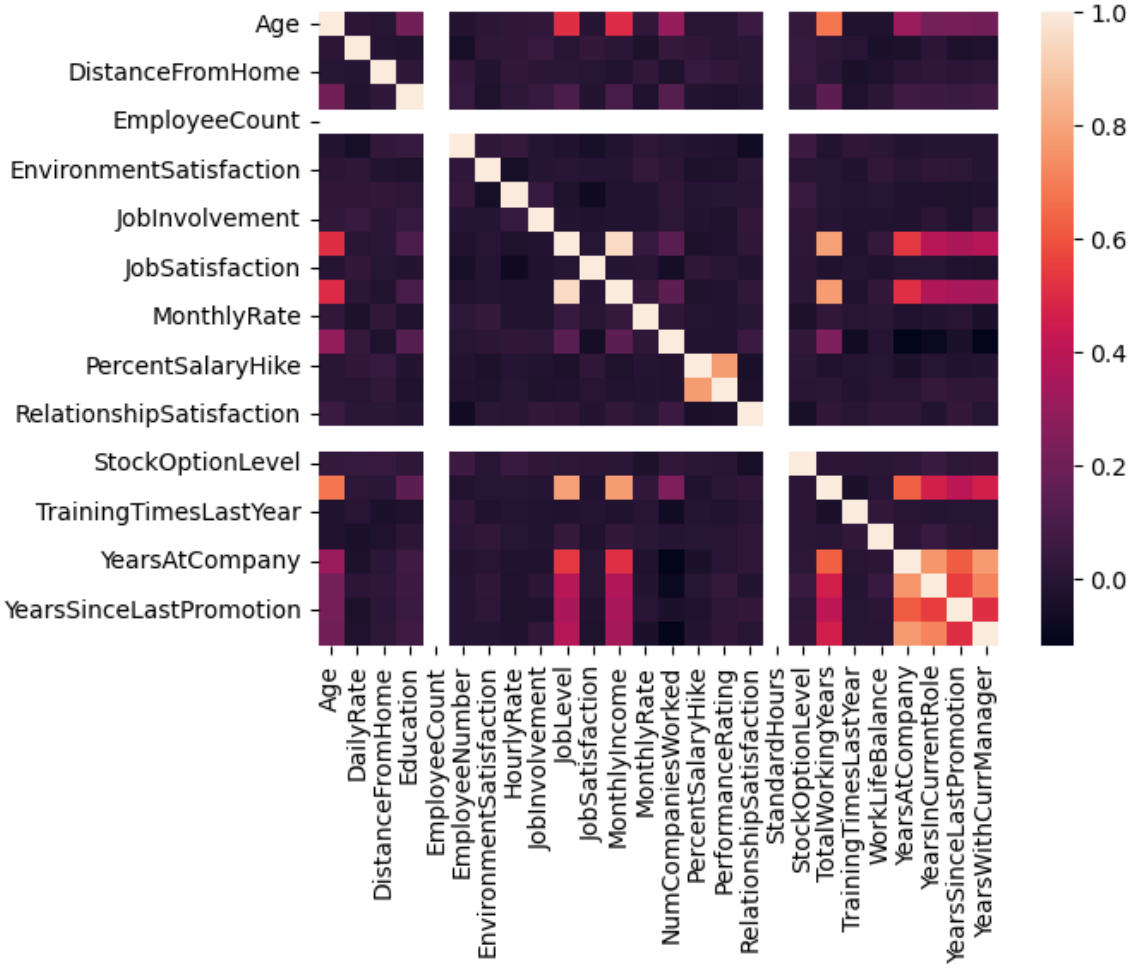
In [19]:
```
sns.heatmap(df.corr())
```

```
C:\Users\shaik\AppData\Local\Temp\ipykernel_16268\58359773.py:1: FutureWarning:
```

In [19]:
```
sns.heatmap(df.corr())
```

```
C:\Users\shaik\AppData\Local\Temp\ipykernel_16268\58359773.py: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 val
ue of numeric_only to silence this warning.
  sns.heatmap(df.corr())
```

Out[19]: <Axes: >

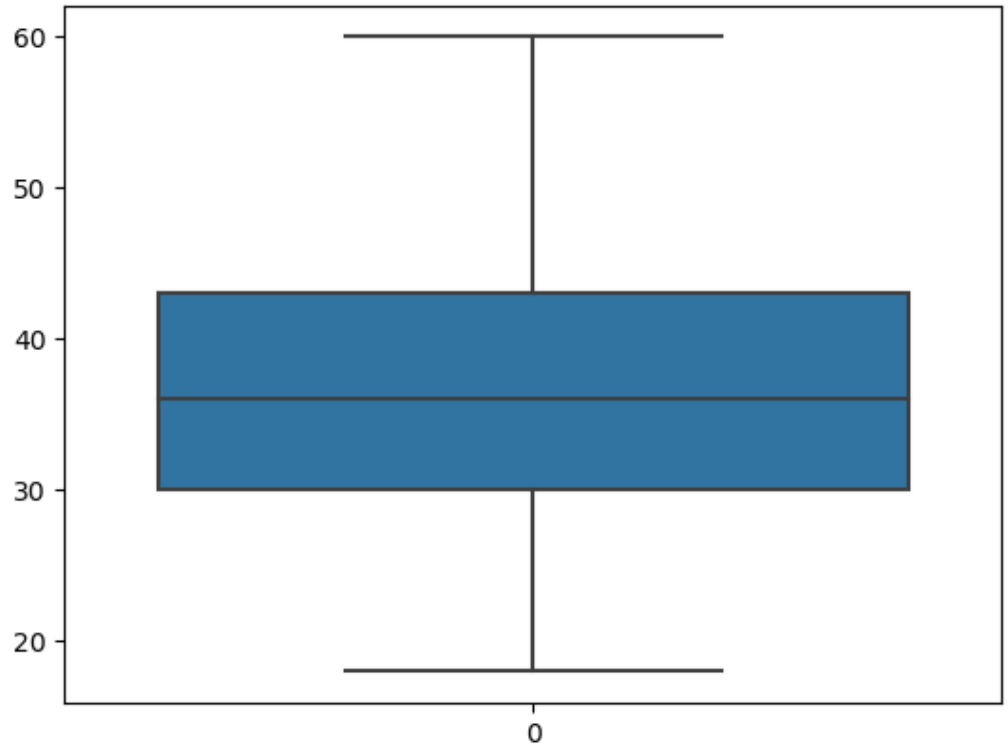


In [20]:
```
sns.boxplot(df['Age'])
```

Out[20]: <Axes: >

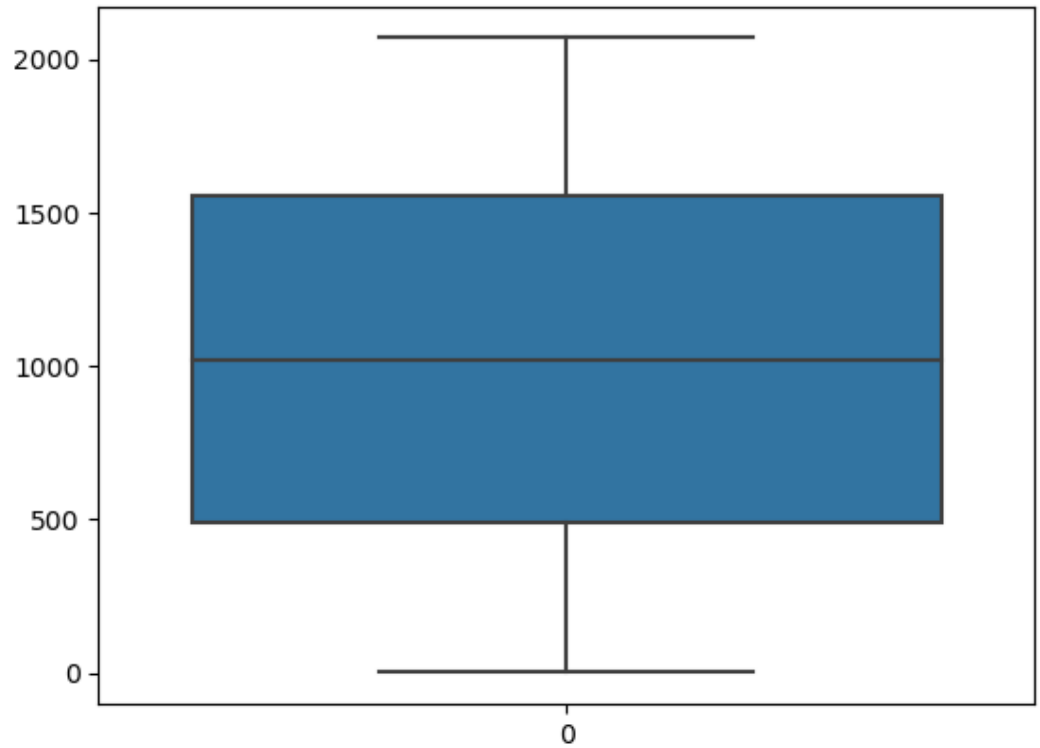
In [20]:
```
sns.boxplot(df['Age'])
```

Out[20]: <Axes: >



In [21]:
```
sns.boxplot(df['EmployeeNumber'])
```

Out[21]: <Axes: >

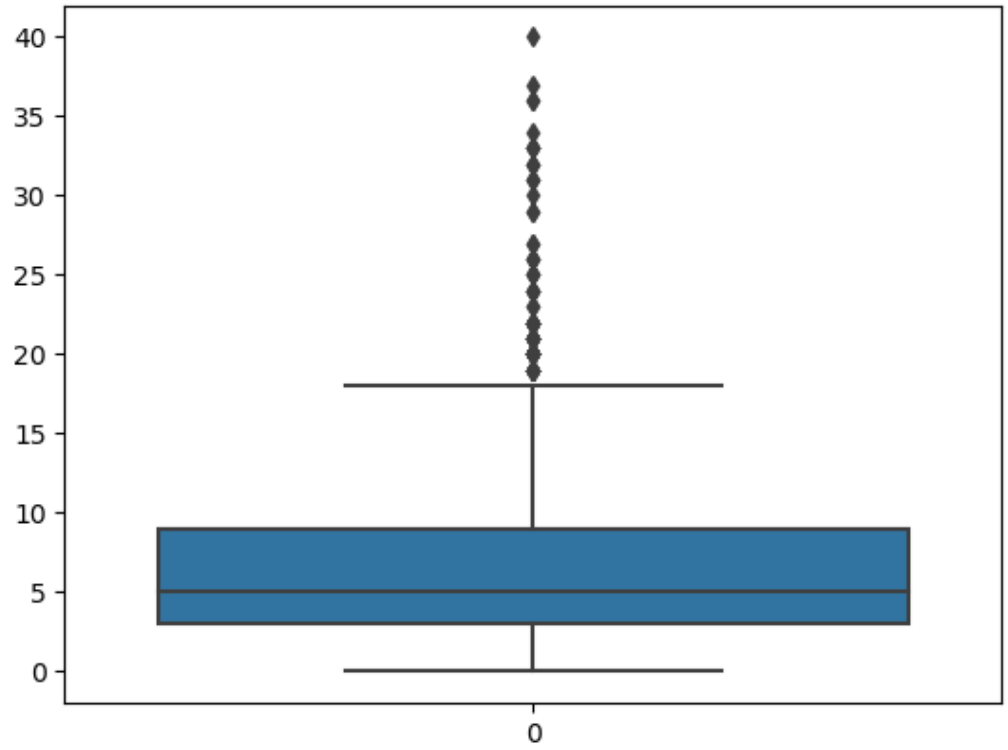


In [22]:
```
sns.boxplot(df['YearsAtCompany'])
```

Out[22]: <Axes: >

In [22]:
```
sns.boxplot(df['YearsAtCompany'])
```

Out[22]: <Axes: >



In [23]:
```
q1=df.YearsAtCompany.quantile(0.25)
q3=df.YearsAtCompany.quantile(0.75)
```

In [24]:
```
q1,q3
```

Out[24]: (3.0, 9.0)

In [25]:
```
IQR=q3-q1
```

In [26]:
```
upper_limit = q3+1.5*IQR
upper_limit
```

Out[26]: 18.0

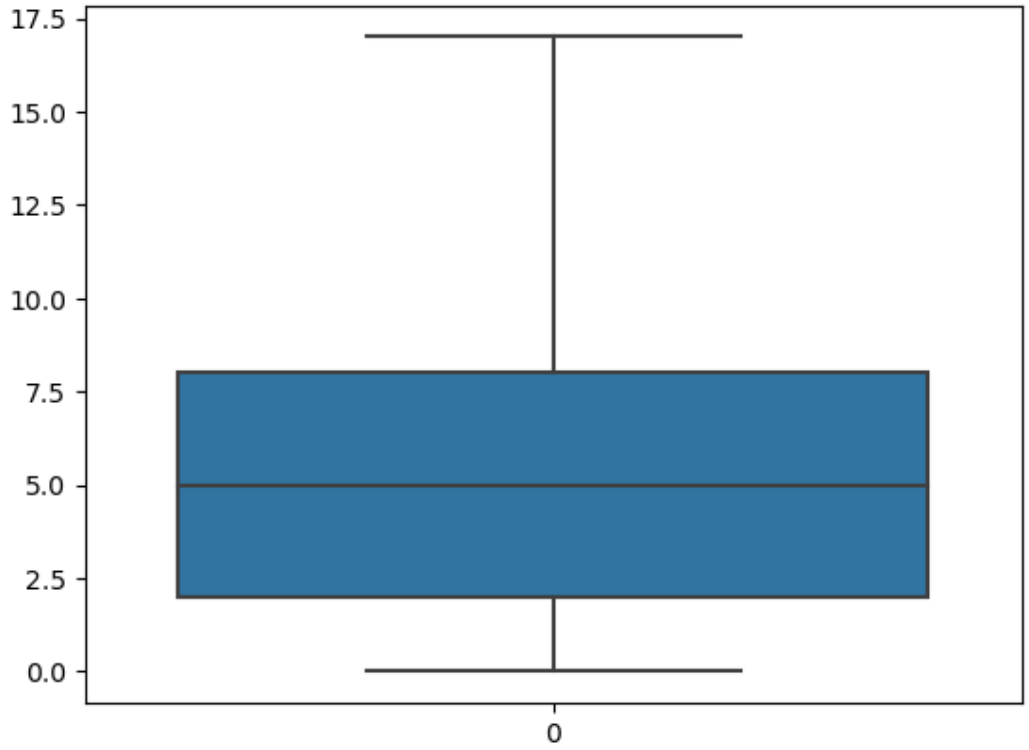
In [27]:
```
df=df[df['YearsAtCompany']<upper_limit]
```

In [28]:
```
sns.boxplot(df['YearsAtCompany'])
```

Out[28]: <Axes: >

In [28]:
```
sns.boxplot(df['YearsAtCompany'])
```

Out[28]: <Axes: >



In [29]:
```
#Dropping the unwanted columns
df.drop(['JobSatisfaction','JobInvolvement','Over18','RelationshipSatisfaction',
```

In [30]:
```
df.head()
```

Out[30]:

	Age	Attrition	BusinessTravel	DailyRate	Department	Education	EducationField	EmployeeCou
0	41	Yes	Travel_Rarely	1102	Sales	2	Life Sciences	
1	49	No	Travel_Frequently	279	Research & Development	1	Life Sciences	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	Other	
3	33	No	Travel_Frequently	1392	Research & Development	4	Life Sciences	
4	27	No	Travel_Rarely	591	Research & Development	1	Medical	

5 rows × 28 columns

In [31]:
```
x=df.drop(['Attrition'],axis=1)
```

In [32]:
```
y=df.Attrition
```

In [32]:
```
y=df.Attrition
```

In [33]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

In [34]:
```
columns=['BusinessTravel','Department','EducationField','Gender','JobRole','Marit
x[columns]=x[columns].apply(le.fit_transform)
```

In [35]:
```
x.head()
```

Out[35]:

	Age	BusinessTravel	DailyRate	Department	Education	EducationField	EmployeeCount	Employ
0	41	2	1102	2	2	1	1	
1	49	1	279	1	1	1	1	
2	37	2	1373	1	2	4	1	
3	33	1	1392	1	4	1	1	
4	27	2	591	1	1	3	1	

5 rows × 27 columns

In [36]:
```
from sklearn.preprocessing import MinMaxScaler
ms=MinMaxScaler()
```

In [37]:
```
x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
```

In [38]:
```
x_scaled.head()
```

Out[38]:

	Age	BusinessTravel	DailyRate	Department	Education	EducationField	EmployeeCount	Em
0	0.547619	1.0	0.716332	1.0	0.25	0.2	0.0	
1	0.738095	0.5	0.126791	0.5	0.00	0.2	0.0	
2	0.452381	1.0	0.910458	0.5	0.25	0.8	0.0	
3	0.357143	0.5	0.924069	0.5	0.75	0.2	0.0	
4	0.214286	1.0	0.350287	0.5	0.00	0.6	0.0	

5 rows × 27 columns

In [39]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x_scaled,y,test_size=0.2,random_
```

In [40]:
```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
```

Out[40]:
```
((1082, 27), (271, 27), (1082,), (271,))
```

Model Building

Model Building

Logistic Regression

In [41]:
```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
```

In [42]:
```
model.fit(x_train,y_train)
```

Out[42]:
```
LogisticRegression
LogisticRegression()
```

In [43]:
```
pred=model.predict(x_test)
pred
```

Out[43]:
```
array(['No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No'],
      dtype=object)
```

In [44]:
```
y_test
```

Out[44]:
```
459      No
```

In [44]:
```
y_test
```

Out[44]:
```
459      No
1076     No
1377     No
91       No
1103     No
        ...
525     Yes
1159     No
1179     No
1435     No
531      No
Name: Attrition, Length: 271, dtype: object
```

In [45]:
```
model.predict(ms.transform([[37,2,1373,1,2,4,1,4,1,92,2,4,3,1,3,1,1,15,3,80,7,3,3
```

```
C:\Users\shaik\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X
does not have valid feature names, but MinMaxScaler was fitted with feature nam
es
  warnings.warn(
C:\Users\shaik\anaconda3\lib\site-packages\sklearn\base.py:420: UserWarning: X
does not have valid feature names, but LogisticRegression was fitted with featu
re names
  warnings.warn(
```

Out[45]:
```
array(['No'], dtype=object)
```

In [46]:
```
from sklearn.metrics import accuracy_score,classification_report,roc_auc_score,rc
```

In [47]:
```
print(accuracy_score(y_test,pred))
```

```
0.8560885608856088
```

In [48]:
```
print(classification_report(y_test,pred))
```

```
              precision    recall  f1-score   support

          No       0.87      0.97      0.92       228
         Yes       0.62      0.23      0.34        43

    accuracy                           0.86       271
   macro avg       0.75      0.60      0.63       271
weighted avg       0.83      0.86      0.83       271
```

Decision Tree Classifier

In [49]:
```
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier()
```

In [50]:
```
dtc.fit(x_train,y_train)
```

Out[50]:
```
DecisionTreeClassifier
```

In [50]:
```
dtc.fit(x_train,y_train)
```

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

In [51]:
```
y_pred=dtc.predict(x_test)
y_pred
```

Out[51]:
```
array(['No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'Yes',
       'Yes', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No',
       'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No',
       'Yes', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'No', 'Yes', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes',
       'Yes', 'Yes', 'Yes', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'Yes',
       'Yes', 'Yes', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'Yes', 'Yes', 'No', 'No', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'No', 'Yes',
       'No', 'No', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes',
       'No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'Yes',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'Yes',
       'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No'], dtype=object)
```

In [52]:
```
y_test
```

Out[52]:
```
459      No
1076     No
1377     No
91       No
1103     No
        ...
525     Yes
1159     No
1179     No
1435     No
531      No
Name: Attrition, Length: 271, dtype: object
```

In [53]:
```
from sklearn.metrics import accuracy_score,classification_report
```

In [54]:
```
print(accuracy_score(y_test,y_pred))
```

In [55]:
```
print(classification_report(y_test,y_pred))
```

```
0.7380073800738007
              precision    recall  f1-score   support
```

In [55]:
```
0.7380073800738007
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

          No       0.87      0.81      0.84       228
         Yes       0.27      0.37      0.31        43

    accuracy                           0.74       271
   macro avg       0.57      0.59      0.57       271
weighted avg       0.78      0.74      0.75       271
```

Huper parameter Tuning

In [56]:
```
from sklearn.model_selection import GridSearchCV
parameter={
 'criterion':['gini','entropy'],
  'splitter':['best','random'],
  'max_depth':[1,2,3,4,5],
  'max_features':['auto', 'sqrt', 'log2']

}
```

In [57]:
```
grid_search=GridSearchCV(estimator=dtc,param_grid=parameter,cv=5,scoring="accurac
```

In [58]:
```
grid_search.fit(x_train,y_train)
```

```
C:\Users\shaik\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: Futu
reWarning: `max_features='auto'` has been deprecated in 1.1 and will be remov
ed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\shaik\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: Futu
reWarning: `max_features='auto'` has been deprecated in 1.1 and will be remov
ed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\shaik\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: Futu
reWarning: `max_features='auto'` has been deprecated in 1.1 and will be remov
ed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\shaik\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: Futu
reWarning: `max_features='auto'` has been deprecated in 1.1 and will be remov
ed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
C:\Users\shaik\anaconda3\lib\site-packages\sklearn\tree\_classes.py:269: Futu
reWarning: `max_features='auto'` has been deprecated in 1.1 and will be remov
ed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'`.
  warnings.warn(
```

In [59]:
```
grid_search.best_params_
```

Out[59]:
```
{'criterion': 'entropy',
 'max_depth': 3,
 'max_features': 'log2',
 'splitter': 'best'}
```

In [60]:
```
dtc_cv=DecisionTreeClassifier(criterion= 'gini',
 max_depth=4,
 max_features='sqrt',
```

In [60]:
```
dtc_cv=DecisionTreeClassifier(criterion= 'gini',
 max_depth=4,
 max_features='sqrt',
 splitter='random')
dtc_cv.fit(x_train,y_train)
```

Out[60]:
```
DecisionTreeClassifier
DecisionTreeClassifier(max_depth=4, max_features='sqrt', splitter='random')
```

In [61]:
```
y1_pred=dtc_cv.predict(x_test)
```

In [62]:
```
print(accuracy_score(y_test,y1_pred))
```

```
0.8523985239852399
```

In [63]:
```
print(classification_report(y_test,y1_pred))
```

```
              precision    recall  f1-score   support

          No       0.88      0.96      0.92       228
         Yes       0.57      0.28      0.37        43

    accuracy                           0.85       271
   macro avg       0.72      0.62      0.65       271
weighted avg       0.83      0.85      0.83       271
```

Random Forest Classifier

In [64]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
```

In [65]:
```
rfc.fit(x_train,y_train)
```

Out[65]:
```
RandomForestClassifier
RandomForestClassifier()
```

In [66]:
```
y2_pred=rfc.predict(x_test)
```

In [67]:
```
y2_pred
```

Out[67]:
```
array(['No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
```

In [67]:
```
y2_pred
```

Out[67]:
```
array(['No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'Yes',
       'No', 'No', 'No', 'Yes', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No',
       'Yes', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'Yes', 'No', 'Yes', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No', 'No',
       'No', 'No', 'No', 'No', 'No', 'No', 'No'], dtype=object)
```

In [68]:
```
y_test
```

Out[68]:
```
459      No
1076     No
1377     No
91       No
1103     No
        ...
525     Yes
1159     No
1179     No
1435     No
531      No
Name: Attrition, Length: 271, dtype: object
```

In [69]:
```
print(accuracy_score(y_test,y2_pred))
```

```
0.8671586715867159
```

In [70]:
```
print(classification_report(y_test,y2_pred))
```

```
              precision    recall  f1-score   support

          No       0.87      0.99      0.93       228
         Yes       0.77      0.23      0.36        43

    accuracy                           0.87       271
   macro avg       0.82      0.61      0.64       271
weighted avg       0.86      0.87      0.84       271
```

In [71]:
```
forest_params = [{'max_depth': list(range(10, 15)), 'max_features': list(range(0,
```

```
macro avg       0.82      0.61      0.64       271
weighted avg    0.86      0.87      0.84       271
```

In [71]:
```
forest_params = [{'max_depth': list(range(10, 15)), 'max_features': list(range(0,
```

In [72]:
```
rfc_cv= GridSearchCV(rfc,param_grid=forest_params,cv=10,scoring="accuracy")
```

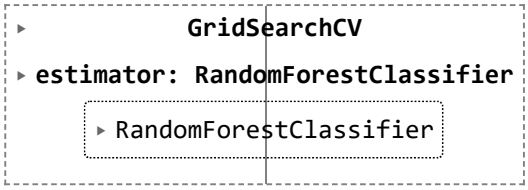
In [73]:
```
rfc_cv.fit(x_train,y_train)
```

```
C:\Users\shaik\anaconda3\lib\site-packages\sklearn\model_selection\_validation.
```

In [73]:
```
rfc_cv.fit(x_train,y_train)
```

```
C:\Users\shaik\anaconda3\lib\site-packages\sklearn\model_selection\_validation.
py:378: FitFailedWarning:
50 fits failed out of a total of 700.
The score on these train-test partitions for these parameters will be set to na
n.
If these failures are not expected, you can try to debug them by setting error_
score='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
-
50 fits failed with the following error:
Traceback (most recent call last):
  File "C:\Users\shaik\anaconda3\lib\site-packages\sklearn\model_selection\_val
idation.py", line 686, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "C:\Users\shaik\anaconda3\lib\site-packages\sklearn\ensemble\_forest.p
y", line 340, in fit
    self._validate_params()
  File "C:\Users\shaik\anaconda3\lib\site-packages\sklearn\base.py", line 581,
in _validate_params
    validate_parameter_constraints(
  File "C:\Users\shaik\anaconda3\lib\site-packages\sklearn\utils\_param_validat
ion.py", line 97, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'max_features' param
eter of RandomForestClassifier must be an int in the range [1, inf), a float in
the range (0.0, 1.0], a str among {'log2', 'sqrt', 'auto' (deprecated)} or Non
e. Got 0 instead.

  warnings.warn(some_fits_failed_message, FitFailedWarning)
C:\Users\shaik\anaconda3\lib\site-packages\sklearn\model_selection\_search.py:9
52: UserWarning: One or more of the test scores are non-finite: [       nan 0.8
354825  0.83823479 0.84379035 0.8456422  0.84287292
 0.84655963 0.84749405 0.84655963 0.84101257 0.8520982  0.84749405
 0.84749405 0.84379035        nan 0.835491   0.8419385  0.84195549
 0.83824329 0.84287292 0.84471628 0.84751104 0.84563371 0.84934591
 0.84471628 0.85024635 0.8456422  0.84288991        nan 0.83641692
 0.83362215 0.8456507  0.84288991 0.85120625 0.85118077 0.84842847
 0.85211519 0.84379884 0.84748556 0.85025484 0.84195549 0.84933741
        nan 0.84103806 0.83917771 0.84103806 0.84563371 0.84472477
 0.84563371 0.84194699 0.85116378 0.84656813 0.84379035 0.85118926
 0.84932892 0.84841148        nan 0.83825178 0.84010364 0.84197248
 0.83917771 0.84285593 0.84840299 0.84932892 0.85303262 0.8483945
 0.84932892 0.84840299 0.84656813 0.84659361]
  warnings.warn(
```

Out[73]:



In [74]:
```
y3_pred=rfc_cv.predict(x_test)
```

In [74]:
```
y3_pred=rfc_cv.predict(x_test)
```

In [75]:
```
print(accuracy_score(y_test,y3_pred))
```

```
0.8376383763837638
```

In [76]:
```
print(classification_report(y_test,y3_pred))
```

```
              precision    recall  f1-score   support

          No       0.87      0.96      0.91       228
         Yes       0.47      0.21      0.29        43

    accuracy                           0.84       271
   macro avg       0.67      0.58      0.60       271
weighted avg       0.80      0.84      0.81       271
```

In [77]:
```
rfc_cv.best_params_
```

Out[77]:
```
{'max_depth': 14, 'max_features': 8}
```

In [ ]:
```

```