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

In [2]:
```
df = pd.read_csv("C:/Users/Sriharsha/Downloads/Data.csv")
```

In [3]:
```
df
```

Out[3]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	Relationsh
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	
...	...	...	...	...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	...	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	...	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	...	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	...	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	...	

1470 rows × 35 columns

In [4]:
```
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 [5]:
```
df.Attrition.unique()
```

Out[5]:
```
array(['Yes', 'No'], dtype=object)
```

In [6]:
```
df.Attrition.value_counts()
```

Out[6]:
```
No     1233
Yes     237
Name: Attrition, dtype: int64
```

Outof 1470 employees there are 237 people who dropped/changed from the company

In [7]:
```
df.duplicated().sum()
#No Duplicates in the data set
```

Out[7]:
```
0
```

In [8]:
```
df.Attrition.replace(['Yes', 'No'],[1,0],inplace = True)
#Yes in Attrition replaced with 1
#No in Attrition replaced with 0.
```

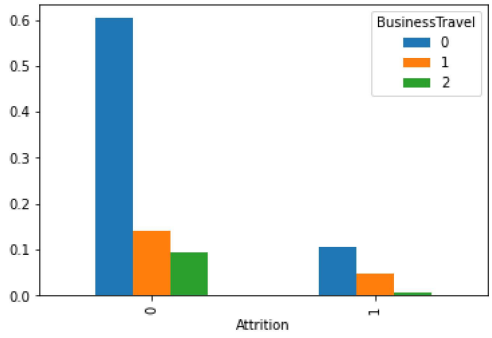
In [9]:
```
df.BusinessTravel.unique()
```

Out[9]:
```
array(['Travel_Rarely', 'Travel_Frequently', 'Non-Travel'], dtype=object)
```

In [10]:
```
df.BusinessTravel.replace(['Travel_Rarely', 'Travel_Frequently', 'Non-Travel'],[0,1,2],inplace = True)
```

In [11]:
```
pd.crosstab(index = df.Attrition,
            columns=df.BusinessTravel,
            normalize = True).plot(kind = 'bar')
```

Out[11]:
```
<AxesSubplot:xlabel='Attrition'>
```



People Who travel Rarely tends to attriate more from The Bar plot

In [12]:
```
df.columns
```

Out[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')
```

In [13]:
```
df.Department.value_counts()
```

Out[13]:
```
Research & Development    961
Sales                     446
Human Resources            63
Name: Department, dtype: int64
```

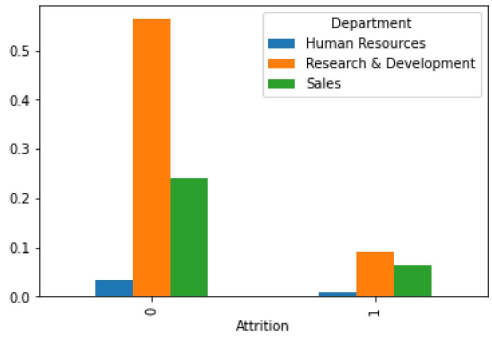
In [14]:
```
pd.crosstab(index = df.Attrition,
            columns= df.Department,
             margins = True,
            normalize = True)
```

Out[14]:

Department	Human Resources	Research & Development	Sales	All
Attrition				
0	0.034694	0.563265	0.240816	0.838776
1	0.008163	0.090476	0.062585	0.161224
All	0.042857	0.653741	0.303401	1.000000

In [15]:
```
pd.crosstab(index = df.Attrition,
            columns= df.Department,
            normalize = True).plot(kind = 'bar')
```

Out[15]: <AxesSubplot:xlabel='Attrition'>



Out of All the 3 departments 'Sales', 'Research & Development', 'Human Resources' --> R&D has More attrition Rate

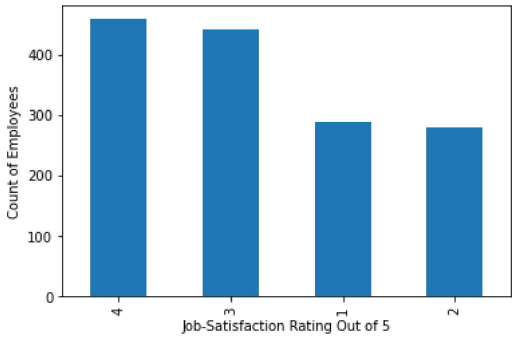
In [16]:
```
df.Department.unique()
```

Out[16]: array(['Sales', 'Research & Development', 'Human Resources'], dtype=object)

In [17]:
```
df.Department.replace(['Sales', 'Research & Development', 'Human Resources'],[1,2,3],inplace = True)
```

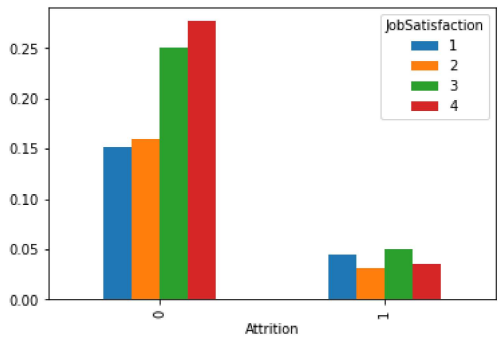
In [18]:
```
df.JobSatisfaction.value_counts().plot(kind = 'bar')
plt.xlabel('Job-Satisfaction Rating Out of 5')
plt.ylabel('Count of Employees')
```

Out[18]: Text(0, 0.5, 'Count of Employees')



In [19]:
```
pd.crosstab(index = df.Attrition,
            columns= df.JobSatisfaction,
            normalize = True).plot(kind = 'bar')#No use
```

Out[19]: <AxesSubplot:xlabel='Attrition'>



People With high Job Satisfaction are interested in staying Back in the Company

People with Average and low job satisfaction are Attriating from the company

In [20]:
```
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   int64
 2   BusinessTravel            1470 non-null   int64
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   int64
 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(29), object(6)
memory usage: 402.1+ KB
```

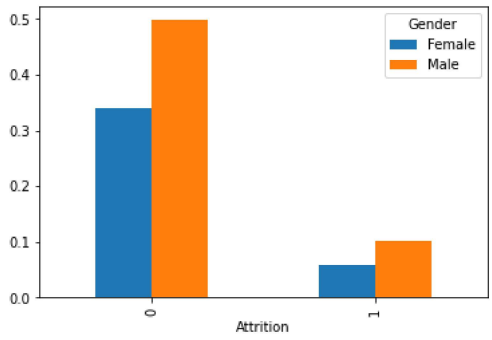
In [21]:
```
pd.crosstab(index = df.Attrition,
            columns= df.Gender,
             margins = True,
            normalize = True)

#Attrition rate in males is 10% and where as in females it is only 5%
```

Out[21]:

Gender	Female	Male	All
Attrition			
0	0.340816	0.497959	0.838776
1	0.059184	0.102041	0.161224
All	0.400000	0.600000	1.000000

In [22]:
```
pd.crosstab(index = df.Attrition,
            columns= df.Gender,
            normalize = True).plot(kind = 'bar')
```

Out[22]: <AxesSubplot:xlabel='Attrition'>



In [23]:
```
df.Gender.replace(['Male','Female'],[1,0],inplace = True)
```

In [24]:
```
df.Over18.unique()
```

Out[24]:
```
array(['Y'], dtype=object)
```

In [25]:
```
df.drop(columns='Over18',inplace = True)
```

In [26]:
```
df.OverTime.value_counts()
```

Out[26]:
```
No     1054
Yes     416
Name: OverTime, dtype: int64
```

In [27]:
```
pd.crosstab(index = df.Attrition,
            columns= df.OverTime,
             margins = True,
            normalize = True)
```

Out[27]:

OverTime	No	Yes	All
Attrition			
0	0.642177	0.196599	0.838776
1	0.074830	0.086395	0.161224
All	0.717007	0.282993	1.000000

People Who are doing Over time are willing to stay back in the company--> Their Stay back rate is around 83.8%

In [28]:
```
df.OverTime.replace(['Yes','No'],[1,0],inplace = True)
```

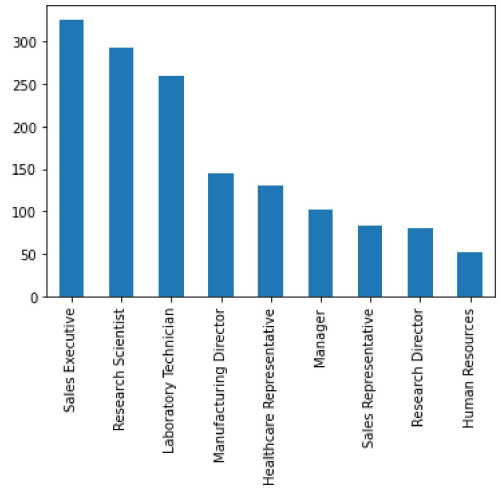
In [ ]:

In [29]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 34 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   int64
 2   BusinessTravel            1470 non-null   int64
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   int64
 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   int64
 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  OverTime                  1470 non-null   int64
 22  PercentSalaryHike         1470 non-null   int64
 23  PerformanceRating         1470 non-null   int64
 24  RelationshipSatisfaction  1470 non-null   int64
 25  StandardHours             1470 non-null   int64
 26  StockOptionLevel          1470 non-null   int64
 27  TotalWorkingYears         1470 non-null   int64
 28  TrainingTimesLastYear     1470 non-null   int64
 29  WorkLifeBalance           1470 non-null   int64
 30  YearsAtCompany            1470 non-null   int64
 31  YearsInCurrentRole        1470 non-null   int64
 32  YearsSinceLastPromotion   1470 non-null   int64
 33  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(31), object(3)
memory usage: 390.6+ KB
```

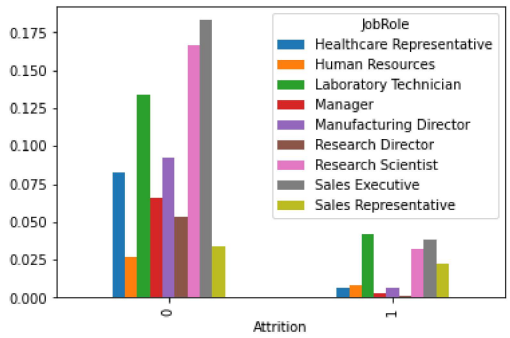
In [30]:
```
df.JobRole.value_counts().plot(kind = 'bar')
```

Out[30]: <AxesSubplot:>



In [31]:
```
pd.crosstab(index = df.Attrition,
            columns= df.JobRole,
            normalize = True).plot(kind = 'bar')
```

Out[31]: <AxesSubplot:xlabel='Attrition'>



Compared to all Job roles Attrition rate is high in Laboratory Technicians and Low in Sales Executive Roles

In [ ]:
```
```

In [32]:
```
df.MaritalStatus.value_counts()
```

Out[32]:
```
Married     673
Single      470
Divorced    327
Name: MaritalStatus, dtype: int64
```

In [33]:
```
pd.crosstab(index = df.Attrition,
            columns= df.MaritalStatus,
             margins = True,
            normalize = True)
```

Out[33]:

MaritalStatus	Divorced	Married	Single	All
Attrition				
0	0.200000	0.400680	0.238095	0.838776
1	0.022449	0.057143	0.081633	0.161224
All	0.222449	0.457823	0.319728	1.000000

In [34]:
```
df.MaritalStatus.unique()
```

Out[34]:
```
array(['Single', 'Married', 'Divorced'], dtype=object)
```

In [35]:
```
df.MaritalStatus.replace(['Single', 'Married', 'Divorced'],[1,2,3],inplace = True)
```

In [36]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 34 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   int64
 2   BusinessTravel            1470 non-null   int64
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   int64
 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   int64
 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   int64
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  OverTime                  1470 non-null   int64
 22  PercentSalaryHike         1470 non-null   int64
 23  PerformanceRating         1470 non-null   int64
 24  RelationshipSatisfaction  1470 non-null   int64
 25  StandardHours             1470 non-null   int64
 26  StockOptionLevel          1470 non-null   int64
 27  TotalWorkingYears         1470 non-null   int64
 28  TrainingTimesLastYear     1470 non-null   int64
 29  WorkLifeBalance           1470 non-null   int64
 30  YearsAtCompany            1470 non-null   int64
 31  YearsInCurrentRole        1470 non-null   int64
 32  YearsSinceLastPromotion   1470 non-null   int64
 33  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(32), object(2)
memory usage: 390.6+ KB
```

In [37]:
```
df.drop(columns='EmployeeCount',inplace = True)
```

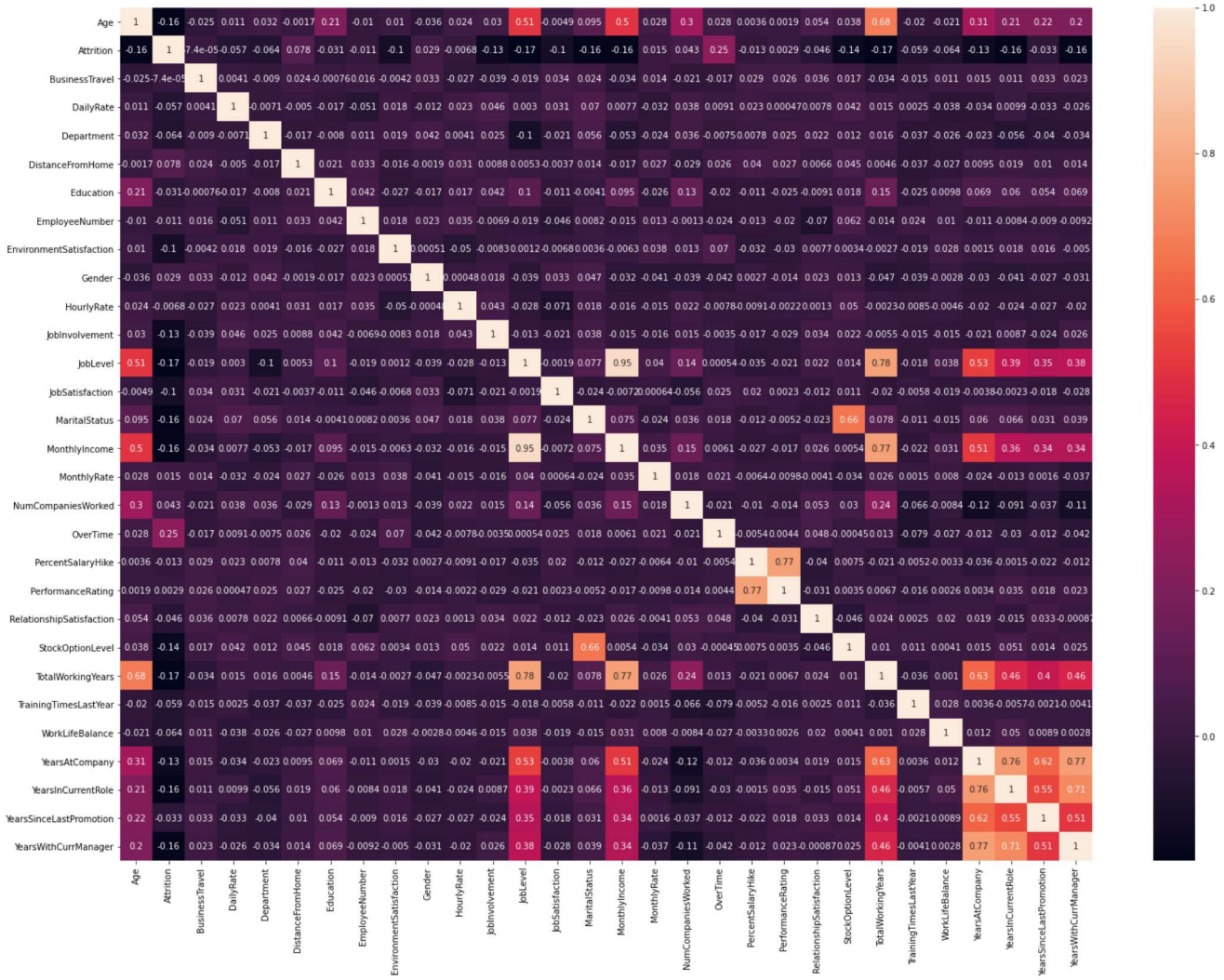
In [38]:
```
df.drop(columns = 'StandardHours',inplace = True)
```

In [39]:
```
df.corr().isnull().sum()
```

Out[39]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

In [40]:
```
plt.figure(figsize = (25,18))
sns.heatmap(df.corr(),annot = True)
```

Out[40]: <AxesSubplot:>



In [41]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 32 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Age                       1470 non-null   int64 
 1   Attrition                 1470 non-null   int64 
 2   BusinessTravel            1470 non-null   int64 
 3   DailyRate                 1470 non-null   int64 
 4   Department                1470 non-null   int64 
 5   DistanceFromHome          1470 non-null   int64 
 6   Education                 1470 non-null   int64 
 7   EducationField            1470 non-null   object
 8   EmployeeNumber            1470 non-null   int64 
 9   EnvironmentSatisfaction   1470 non-null   int64 
 10  Gender                    1470 non-null   int64 
 11  HourlyRate                1470 non-null   int64 
 12  JobInvolvement            1470 non-null   int64 
 13  JobLevel                  1470 non-null   int64 
 14  JobRole                   1470 non-null   object
 15  JobSatisfaction           1470 non-null   int64 
 16  MaritalStatus             1470 non-null   int64 
 17  MonthlyIncome             1470 non-null   int64 
 18  MonthlyRate               1470 non-null   int64 
 19  NumCompaniesWorked        1470 non-null   int64 
 20  OverTime                  1470 non-null   int64 
 21  PercentSalaryHike         1470 non-null   int64 
 22  PerformanceRating         1470 non-null   int64 
 23  RelationshipSatisfaction  1470 non-null   int64 
 24  StockOptionLevel          1470 non-null   int64 
 25  TotalWorkingYears         1470 non-null   int64 
 26  TrainingTimesLastYear     1470 non-null   int64 
 27  WorkLifeBalance           1470 non-null   int64 
 28  YearsAtCompany            1470 non-null   int64 
 29  YearsInCurrentRole        1470 non-null   int64 
 30  YearsSinceLastPromotion   1470 non-null   int64 
 31  YearsWithCurrManager      1470 non-null   int64 
dtypes: int64(30), object(2)
memory usage: 367.6+ KB
```

In [42]:
```
dummy = df.drop(columns = ['JobRole','EducationField'])
```

In [43]:
```
dummy.corr()
```

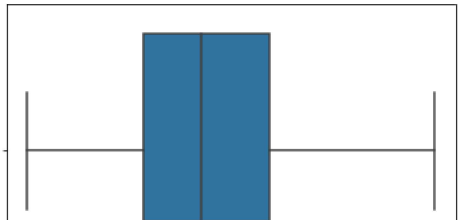
Out[43]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EmployeeNumber	EnvironmentSatisfactio
Age	1.000000	-0.159205	-0.024751	0.010661	0.031882	-0.001686	0.208034	-0.010145	0.01014
Attrition	-0.159205	1.000000	-0.000074	-0.056652	-0.063991	0.077924	-0.031373	-0.010577	-0.10336
BusinessTravel	-0.024751	-0.000074	1.000000	0.004086	-0.009044	0.024469	-0.000757	0.015578	-0.00417
DailyRate	0.010661	-0.056652	0.004086	1.000000	-0.007109	-0.004985	-0.016806	-0.050990	0.01835
Department	0.031882	-0.063991	-0.009044	-0.007109	1.000000	-0.017225	-0.007996	0.010895	0.01939
DistanceFromHome	-0.001686	0.077924	0.024469	-0.004985	-0.017225	1.000000	0.021042	0.032916	-0.01607
Education	0.208034	-0.031373	-0.000757	-0.016806	-0.007996	0.021042	1.000000	0.042070	-0.02712
EmployeeNumber	-0.010145	-0.010577	0.015578	-0.050990	0.010895	0.032916	0.042070	1.000000	0.01762
EnvironmentSatisfaction	0.010146	-0.103369	-0.004174	0.018355	0.019395	-0.016075	-0.027128	0.017621	1.00000
Gender	-0.036311	0.029453	0.032981	-0.011716	0.041583	-0.001851	-0.016547	0.022556	0.00050
HourlyRate	0.024287	-0.006846	-0.026528	0.023381	0.004144	0.031131	0.016775	0.035179	-0.04985
JobInvolvement	0.029820	-0.130016	-0.039062	0.046135	0.024586	0.008783	0.042438	-0.006888	-0.00827
JobLevel	0.509604	-0.169105	-0.019311	0.002966	-0.101963	0.005303	0.101589	-0.018519	0.00121
JobSatisfaction	-0.004892	-0.103481	0.033962	0.030571	-0.021001	-0.003669	-0.011296	-0.046247	-0.00678
MaritalStatus	0.095029	-0.162070	0.024001	0.069586	0.056073	0.014437	-0.004053	0.008155	0.00359
MonthlyIncome	0.497855	-0.159840	-0.034319	0.007707	-0.053130	-0.017014	0.094961	-0.014829	-0.00625
MonthlyRate	0.028051	0.015170	0.014107	-0.032182	-0.023642	0.027473	-0.026084	0.012648	0.03760
NumCompaniesWorked	0.299635	0.043494	-0.020875	0.038153	0.035882	-0.029251	0.126317	-0.001251	0.01259
OverTime	0.028062	0.246118	-0.016543	0.009135	-0.007481	0.025514	-0.020322	-0.024037	0.07013
PercentSalaryHike	0.003634	-0.013478	0.029377	0.022704	0.007840	0.040235	-0.011111	-0.012944	-0.03170
PerformanceRating	0.001904	0.002889	0.026341	0.000473	0.024604	0.027110	-0.024539	-0.020359	-0.02954
RelationshipSatisfaction	0.053535	-0.045872	0.035986	0.007846	0.022414	0.006557	-0.009118	-0.069861	0.00766
StockOptionLevel	0.037510	-0.137145	0.016727	0.042143	0.012193	0.044872	0.018422	0.062227	0.00343
TotalWorkingYears	0.680381	-0.171063	-0.034226	0.014515	0.015762	0.004628	0.148280	-0.014365	-0.00269
TrainingTimesLastYear	-0.019621	-0.059478	-0.015240	0.002453	-0.036875	-0.036942	-0.025100	0.023603	-0.01935
WorkLifeBalance	-0.021490	-0.063939	0.011256	-0.037848	-0.026383	-0.026556	0.009819	0.010309	0.02762
YearsAtCompany	0.311309	-0.134392	0.014575	-0.034055	-0.022920	0.009508	0.069114	-0.011240	0.00145
YearsInCurrentRole	0.212901	-0.160545	0.011497	0.009932	-0.056315	0.018845	0.060236	-0.008416	0.01800
YearsSinceLastPromotion	0.216513	-0.033019	0.032591	-0.033229	-0.040061	0.010029	0.054254	-0.009019	0.01619
YearsWithCurrManager	0.202089	-0.156199	0.022636	-0.026363	-0.034282	0.014406	0.069065	-0.009197	-0.00499

30 rows × 30 columns

In [44]:
```
cols = dummy.columns
for i in cols:
    plt.subplots()
    sns.boxplot(df[i])
```

```
  warnings.warn(
C:\Users\sriharsha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a key
word arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\sriharsha\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a key
word arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an exp
licit keyword will result in an error or misinterpretation.
  warnings.warn(
```



In [45]:
```
performanceRating
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-45-7edb9aa4aab8> in <module>
----> 1 performanceRating

NameError: name 'performanceRating' is not defined
```

In [46]:
```
df.TotalWorkingYears.unique()
```

Out[46]:
```
array([ 8, 10,  7,  6, 12,  1, 17,  5,  3, 31, 13,  0, 26, 24, 22,  9, 19,
        2, 23, 14, 15,  4, 29, 28, 21, 25, 20, 11, 16, 37, 38, 30, 40, 18,
       36, 34, 32, 33, 35, 27], dtype=int64)
```

In [47]:
```
cols
```

Out[47]:
```
Index(['Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EmployeeNumber',
       'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
       'JobLevel', 'JobSatisfaction', 'MaritalStatus', 'MonthlyIncome',
       'MonthlyRate', 'NumCompaniesWorked', 'OverTime', 'PercentSalaryHike',
       'PerformanceRating', 'RelationshipSatisfaction', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

In [48]:
```
for i in cols:
    plt.subplots()
    sns.distplot(df[i])
```

```
C:\Users\sriharsha\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated funct
ion and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\sriharsha\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated funct
ion and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\sriharsha\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated funct
ion and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\sriharsha\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated funct
ion and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\sriharsha\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated funct
ion and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
```

In [95]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 32 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Age                       1470 non-null   int64 
 1   Attrition                 1470 non-null   int64 
 2   BusinessTravel            1470 non-null   int64 
 3   DailyRate                 1470 non-null   int64 
 4   Department                1470 non-null   int64 
 5   DistanceFromHome          1470 non-null   int64 
 6   Education                 1470 non-null   int64 
 7   EducationField            1470 non-null   object
 8   EmployeeNumber            1470 non-null   int64 
 9   EnvironmentSatisfaction   1470 non-null   int64 
 10  Gender                    1470 non-null   int64 
 11  HourlyRate                1470 non-null   int64 
 12  JobInvolvement            1470 non-null   int64 
 13  JobLevel                  1470 non-null   int64 
 14  JobRole                   1470 non-null   object
 15  JobSatisfaction           1470 non-null   int64 
 16  MaritalStatus             1470 non-null   int64 
 17  MonthlyIncome             1470 non-null   int64 
 18  MonthlyRate               1470 non-null   int64 
 19  NumCompaniesWorked        1470 non-null   int64 
 20  OverTime                  1470 non-null   int64 
 21  PercentSalaryHike         1470 non-null   int64 
 22  PerformanceRating         1470 non-null   int64 
 23  RelationshipSatisfaction  1470 non-null   int64 
 24  StockOptionLevel          1470 non-null   int64 
 25  TotalWorkingYears         1470 non-null   int64 
 26  TrainingTimesLastYear     1470 non-null   int64 
 27  WorkLifeBalance           1470 non-null   int64 
 28  YearsAtCompany            1470 non-null   int64 
 29  YearsInCurrentRole        1470 non-null   int64 
 30  YearsSinceLastPromotion   1470 non-null   int64 
 31  YearsWithCurrManager      1470 non-null   int64 
dtypes: int64(30), object(2)
memory usage: 367.6+ KB
```

In [118]:
```
dummy = df[['Age', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EmployeeNumber',
       'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
       'JobLevel', 'JobSatisfaction', 'MaritalStatus',
       'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'OverTime',
       'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
       'YearsSinceLastPromotion', 'YearsWithCurrManager']]
```

In [119]:
```
dummy.corr()
```

Out[119]:

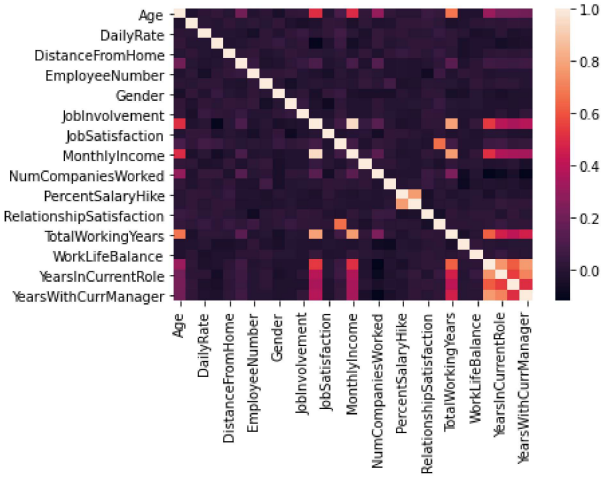
	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EmployeeNumber	EnvironmentSatisfaction	Gende
Age	1.000000	-0.024751	0.010661	0.031882	-0.001686	0.208034	-0.010145	0.010146	-0.03631
BusinessTravel	-0.024751	1.000000	0.004086	-0.009044	0.024469	-0.000757	0.015578	-0.004174	0.03298
DailyRate	0.010661	0.004086	1.000000	-0.007109	-0.004985	-0.016806	-0.050990	0.018355	-0.01171
Department	0.031882	-0.009044	-0.007109	1.000000	-0.017225	-0.007996	0.010895	0.019395	0.04158
DistanceFromHome	-0.001686	0.024469	-0.004985	-0.017225	1.000000	0.021042	0.032916	-0.016075	-0.00185
Education	0.208034	-0.000757	-0.016806	-0.007996	0.021042	1.000000	0.042070	-0.027128	-0.01654
EmployeeNumber	-0.010145	0.015578	-0.050990	0.010895	0.032916	0.042070	1.000000	0.017621	0.02255
EnvironmentSatisfaction	0.010146	-0.004174	0.018355	0.019395	-0.016075	-0.027128	0.017621	1.000000	0.00050
Gender	-0.036311	0.032981	-0.011716	0.041583	-0.001851	-0.016547	0.022556	0.000508	1.00000
HourlyRate	0.024287	-0.026528	0.023381	0.004144	0.031131	0.016775	0.035179	-0.049857	-0.00047
JobInvolvement	0.029820	-0.039062	0.046135	0.024586	0.008783	0.042438	-0.006888	-0.008278	0.01796
JobLevel	0.509604	-0.019311	0.002966	-0.101963	0.005303	0.101589	-0.018519	0.001212	-0.03940
JobSatisfaction	-0.004892	0.033962	0.030571	-0.021001	-0.003669	-0.011296	-0.046247	-0.006784	0.03325
MaritalStatus	0.095029	0.024001	0.069586	0.056073	0.014437	-0.004053	0.008155	0.003593	0.04718
MonthlyIncome	0.497855	-0.034319	0.007707	-0.053130	-0.017014	0.094961	-0.014829	-0.006259	-0.03185
MonthlyRate	0.028051	0.014107	-0.032182	-0.023642	0.027473	-0.026084	0.012648	0.037600	-0.04148
NumCompaniesWorked	0.299635	-0.020875	0.038153	0.035882	-0.029251	0.126317	-0.001251	0.012594	-0.03914
OverTime	0.028062	-0.016543	0.009135	-0.007481	0.025514	-0.020322	-0.024037	0.070132	-0.04192
PercentSalaryHike	0.003634	0.029377	0.022704	0.007840	0.040235	-0.011111	-0.012944	-0.031701	0.00273
PerformanceRating	0.001904	0.026341	0.000473	0.024604	0.027110	-0.024539	-0.020359	-0.029548	-0.01385
RelationshipSatisfaction	0.053535	0.035986	0.007846	0.022414	0.006557	-0.009118	-0.069861	0.007665	0.02286
StockOptionLevel	0.037510	0.016727	0.042143	0.012193	0.044872	0.018422	0.062227	0.003432	0.01271
TotalWorkingYears	0.680381	-0.034226	0.014515	0.015762	0.004628	0.148280	-0.014365	-0.002693	-0.04688
TrainingTimesLastYear	-0.019621	-0.015240	0.002453	-0.036875	-0.036942	-0.025100	0.023603	-0.019359	-0.03878
WorkLifeBalance	-0.021490	0.011256	-0.037848	-0.026383	-0.026556	0.009819	0.010309	0.027627	-0.00275
YearsAtCompany	0.311309	0.014575	-0.034055	-0.022920	0.009508	0.069114	-0.011240	0.001458	-0.02974
YearsInCurrentRole	0.212901	0.011497	0.009932	-0.056315	0.018845	0.060236	-0.008416	0.018007	-0.04148
YearsSinceLastPromotion	0.216513	0.032591	-0.033229	-0.040061	0.010029	0.054254	-0.009019	0.016194	-0.02698
YearsWithCurrManager	0.202089	0.022636	-0.026363	-0.034282	0.014406	0.069065	-0.009197	-0.004999	-0.03059

29 rows × 29 columns

In [121]:
```
from sklearn.preprocessing import normalize
nr = normalize(dummy,norm = 'l2')
```

In [123]:
```
sns.heatmap(dummy.corr())
```

Out[123]:
```
<AxesSubplot:>
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [125]:
```
x = dummy[['Age', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EmployeeNumber',
       'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement',
       'JobLevel', 'JobSatisfaction', 'MaritalStatus',
       'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked', 'OverTime',
       'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
       'YearsSinceLastPromotion', 'YearsWithCurrManager']]
y = df['Attrition']
```

In [126]:
```
from sklearn.model_selection import train_test_split
trainx,testx,trainy,testy = train_test_split(x,y)
```

Logistic Regression

In [127]:
```
from sklearn.linear_model import LogisticRegression
lg = LogisticRegression(random_state=42)
```

In [128]:
```
lg.fit(trainx,trainy)
```

```
C:\Users\sriharsha\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs failed to conve
rge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modules/line
ar_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
```

Out[128]:
```
LogisticRegression(random_state=42)
```

In [129]:
```
yp = lg.predict(testx)
```

In [130]:
```
from sklearn.metrics import accuracy_score
ac = accuracy_score(testy,yp)
ac*100
```

Out[130]:
```
87.5
```

In [131]:
```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier()
```

In [132]:
```
dtc.fit(trainx,trainy)
```

Out[132]:
```
DecisionTreeClassifier()
```

In [133]:
```
ypdtc = dtc.predict(testx)
```

In [134]:
```
accuracy_score(testy,ypdtc)*100
```

Out[134]:
```
80.70652173913044
```

USING SVM as SVC

In [135]:
```
from sklearn.svm import SVC
s = SVC()
```

In [136]:
```
s.fit(trainx,trainy)
```

Out[136]:
```
SVC()
```

In [137]:
```
ypsvc = s.predict(testx)
```

In [138]:
```
accuracy_score(testy,ypsvc)*100
```

Out[138]:
```
86.41304347826086
```

In [139]:
```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
```

In [140]:
```
rf.fit(trainx,trainy)
```

Out[140]:
```
RandomForestClassifier()
```

In [141]:
```
yprf = rf.predict(testx)
```

In [142]:
```
accuracy_score(testy,yprf)*100
```

Out[142]:
```
88.58695652173914
```

In [ ]:

In [ ]:

In [ ]: