Assignment- HR Attrition Data

In this assignment , we need to predict whether a give employee will leave the organization or not. Your target column is Attrition We will create a model with the following steps: Import the relevant packages Download and explore the dataset Perform EDA, Apply dataset for preprocessing Predict the target columns

In [1]:

```
#IMPORTING NECCESSARY PACKAGE
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
dataset = pd.read_csv('HR_Employee_Attrition-1.csv')
```

In [3]:

```
dataset.head()
```

Out[3]:

	EmployeeNumber	Attrition	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	...	RelationshipSatisfaction	StandardHours
0	1	Yes	41	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	...	1	80
1	2	No	49	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	...	4	80
2	3	Yes	37	Travel_Rarely	1373	Research & Development	2	2	Other	1	...	2	80
3	4	No	33	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	...	3	80
4	5	No	27	Travel_Rarely	591	Research & Development	2	1	Medical	1	...	4	80

5 rows × 35 columns

In [4]:

```
dataset.tail()
```

Out[4]:

	EmployeeNumber	Attrition	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	...	RelationshipSatisfaction	StandardHo
2935	2936	No	36	Travel_Frequently	884	Research & Development	23	2	Medical	1	...	3	
2936	2937	No	39	Travel_Rarely	613	Research & Development	6	1	Medical	1	...	1	
2937	2938	No	27	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	...	2	
2938	2939	No	49	Travel_Frequently	1023	Sales	2	3	Medical	1	...	4	
2939	2940	No	34	Travel_Rarely	628	Research & Development	8	3	Medical	1	...	1	

5 rows × 35 columns

In [5]:

```
dataset.shape
```

Out[5]:

```
(2940, 35)
```

In [6]:

```
dataset.columns
```

Out[6]:

```
Index(['EmployeeNumber', 'Attrition', 'Age', 'BusinessTravel', 'DailyRate',
       'Department', 'DistanceFromHome', 'Education', 'EducationField',
       'EmployeeCount', '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 [7]:

```
dataset.describe()
```

Out[7]:

	EmployeeNumber	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	...	Relationshi
count	2940.000000	2940.000000	2940.000000	2940.000000	2940.000000	2940.0	2940.000000	2940.000000	2940.000000	2940.000000	...	
mean	1470.500000	36.923810	802.485714	9.192517	2.912925	1.0	2.721769	65.891156	2.729932	2.063946	...	
std	848.849221	9.133819	403.440447	8.105485	1.023991	0.0	1.092896	20.325969	0.711440	1.106752	...	
min	1.000000	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	30.000000	1.000000	1.000000	...	
25%	735.750000	30.000000	465.000000	2.000000	2.000000	1.0	2.000000	48.000000	2.000000	1.000000	...	
50%	1470.500000	36.000000	802.000000	7.000000	3.000000	1.0	3.000000	66.000000	3.000000	2.000000	...	
75%	2205.250000	43.000000	1157.000000	14.000000	4.000000	1.0	4.000000	84.000000	3.000000	3.000000	...	
max	2940.000000	60.000000	1499.000000	29.000000	5.000000	1.0	4.000000	100.000000	4.000000	5.000000	...	

8 rows × 26 columns

In [8]:

```
dataset.isnull().sum()
```

Out[8]:

```
EmployeeNumber              0
Attrition                   0
Age                         0
BusinessTravel              0
```

```
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               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
```

In [9]:

```
dataset['Attrition'].value_counts()
```

Out[9]:

```
No     2466
Yes     474
Name: Attrition, dtype: int64
```

In [10]:

```
dataset.describe().T
```

Out[10]:

	count	mean	std	min	25%	50%	75%	max
EmployeeNumber	2940.0	1470.500000	848.849221	1.0	735.75	1470.5	2205.25	2940.0
Age	2940.0	36.923810	9.133819	18.0	30.00	36.0	43.00	60.0
DailyRate	2940.0	802.485714	403.440447	102.0	465.00	802.0	1157.00	1499.0
DistanceFromHome	2940.0	9.192517	8.105485	1.0	2.00	7.0	14.00	29.0
Education	2940.0	2.912925	1.023991	1.0	2.00	3.0	4.00	5.0
EmployeeCount	2940.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
EnvironmentSatisfaction	2940.0	2.721769	1.092896	1.0	2.00	3.0	4.00	4.0
HourlyRate	2940.0	65.891156	20.325969	30.0	48.00	66.0	84.00	100.0
JobInvolvement	2940.0	2.729932	0.711440	1.0	2.00	3.0	3.00	4.0
JobLevel	2940.0	2.063946	1.106752	1.0	1.00	2.0	3.00	5.0
JobSatisfaction	2940.0	2.728571	1.102658	1.0	2.00	3.0	4.00	4.0
MonthlyIncome	2940.0	6502.931293	4707.155770	1009.0	2911.00	4919.0	8380.00	19999.0
MonthlyRate	2940.0	14313.103401	7116.575021	2094.0	8045.00	14235.5	20462.00	26999.0
NumCompaniesWorked	2940.0	2.693197	2.497584	0.0	1.00	2.0	4.00	9.0
PercentSalaryHike	2940.0	15.209524	3.659315	11.0	12.00	14.0	18.00	25.0
PerformanceRating	2940.0	3.153741	0.360762	3.0	3.00	3.0	3.00	4.0
RelationshipSatisfaction	2940.0	2.712245	1.081025	1.0	2.00	3.0	4.00	4.0
StandardHours	2940.0	80.000000	0.000000	80.0	80.00	80.0	80.00	80.0
StockOptionLevel	2940.0	0.793878	0.851932	0.0	0.00	1.0	1.00	3.0
TotalWorkingYears	2940.0	11.279592	7.779458	0.0	6.00	10.0	15.00	40.0
TrainingTimesLastYear	2940.0	2.799320	1.289051	0.0	2.00	3.0	3.00	6.0
WorkLifeBalance	2940.0	2.761224	0.706356	1.0	2.00	3.0	3.00	4.0
YearsAtCompany	2940.0	7.008163	6.125483	0.0	3.00	5.0	9.00	40.0
YearsInCurrentRole	2940.0	4.229252	3.622521	0.0	2.00	3.0	7.00	18.0
YearsSinceLastPromotion	2940.0	2.187755	3.221882	0.0	0.00	1.0	3.00	15.0
YearsWithCurrManager	2940.0	4.123129	3.567529	0.0	2.00	3.0	7.00	17.0

In [11]:

```
dataset.nunique()
```

Out[11]:

```
EmployeeNumber              2940
Attrition                      2
Age                           43
BusinessTravel                 3
DailyRate                    886
Department                     3
DistanceFromHome              29
Education                      5
EducationField                 6
EmployeeCount                  1
EnvironmentSatisfaction        4
Gender                         2
HourlyRate                    71
JobInvolvement                 4
JobLevel                       5
JobRole                        9
JobSatisfaction                4
MaritalStatus                  3
```

```
MonthlyIncome               1349
MonthlyRate                 1427
NumCompaniesWorked            10
Over18                         1
OverTime                       2
PercentSalaryHike             15
PerformanceRating              2
RelationshipSatisfaction       4
StandardHours                  1
StockOptionLevel               4
TotalWorkingYears             40
TrainingTimesLastYear          7
WorkLifeBalance                4
YearsAtCompany                37
YearsInCurrentRole            19
YearsSinceLastPromotion       16
YearsWithCurrManager          18
dtype: int64
```

In [12]:

```
dataset.dtypes
```

Out[12]:

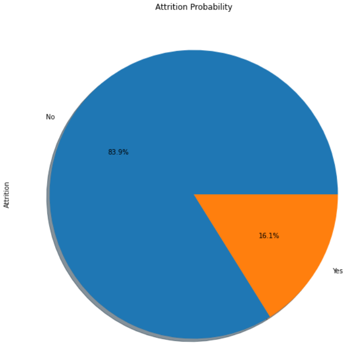
```
EmployeeNumber               int64
Attrition                   object
Age                          int64
BusinessTravel              object
DailyRate                    int64
Department                  object
DistanceFromHome             int64
Education                    int64
EducationField              object
EmployeeCount                int64
EnvironmentSatisfaction      int64
Gender                      object
HourlyRate                   int64
JobInvolvement               int64
JobLevel                     int64
JobRole                     object
JobSatisfaction              int64
MaritalStatus               object
MonthlyIncome                int64
MonthlyRate                  int64
NumCompaniesWorked           int64
Over18                      object
OverTime                    object
PercentSalaryHike            int64
PerformanceRating            int64
RelationshipSatisfaction     int64
StandardHours                int64
StockOptionLevel             int64
TotalWorkingYears            int64
TrainingTimesLastYear        int64
WorkLifeBalance              int64
YearsAtCompany               int64
YearsInCurrentRole           int64
YearsSinceLastPromotion      int64
YearsWithCurrManager         int64
dtype: object
```

In [13]:

```
f,ax=plt.subplots(figsize=(10,10))
ax = dataset['Attrition'].value_counts().plot.pie(explode=[0,0],autopct = '%1.1f%%',shadow=True)
ax.set_title('Attrition Probability')
```

Out[13]:

```
Text(0.5, 1.0, 'Attrition Probability')
```



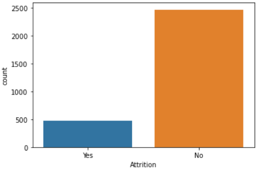
In [14]:

```
sns.countplot(dataset['Attrition'])
```

```
C:\Users\PC\anaconda3p\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.
12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
```

Out[14]:

```
<AxesSubplot:xlabel='Attrition', ylabel='count'>
```

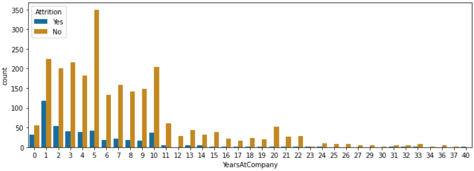


In [15]:

```
plt.subplots(figsize=(12,4))
sns.countplot(x='YearsAtCompany',hue='Attrition',data=dataset,palette ='colorblind')
```

Out[15]:

```
<AxesSubplot:xlabel='YearsAtCompany', ylabel='count'>
```

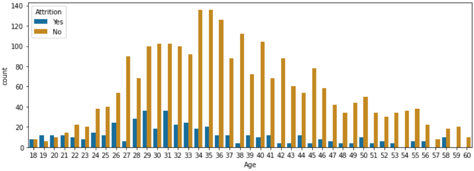


In [16]:

```
#analysing data by attrition on the basis of the age
plt.subplots(figsize=(12,4))
sns.countplot(x='Age',hue='Attrition', data=dataset,palette = 'colorblind')
```

Out[16]:

```
<AxesSubplot:xlabel='Age', ylabel='count'>
```



In [17]:

```
for column in dataset.columns:
    if dataset[column].dtype == object:
        print(str(column)+ ':' + str(dataset[column].unique()))
        print(dataset[column].value_counts())
        print('-------------------------------------------------------------------')
```

```
Attrition:['Yes' 'No']
No     2466
Yes     474
Name: Attrition, dtype: int64
-------------------------------------------------------------------
BusinessTravel:['Travel_Rarely' 'Travel_Frequently' 'Non-Travel']
Travel_Rarely        2086
Travel_Frequently     554
Non-Travel            300
Name: BusinessTravel, dtype: int64
-------------------------------------------------------------------
Department:['Sales' 'Research & Development' 'Human Resources']
Research & Development    1922
Sales                      892
Human Resources            126
Name: Department, dtype: int64
-------------------------------------------------------------------
EducationField:['Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree'
 'Human Resources']
Life Sciences       1212
Medical              928
Marketing            318
Technical Degree     264
Other                164
Human Resources       54
Name: EducationField, dtype: int64
-------------------------------------------------------------------
Gender:['Female' 'Male']
Male      1764
Female    1176
Name: Gender, dtype: int64
-------------------------------------------------------------------
```

```
JobRole:['Sales Executive' 'Research Scientist' 'Laboratory Technician'
 'Manufacturing Director' 'Healthcare Representative' 'Manager'
 'Sales Representative' 'Research Director' 'Human Resources']
Sales Executive              652
Research Scientist           584
Laboratory Technician        518
Manufacturing Director       290
Healthcare Representative    262
Manager                      204
Sales Representative         166
Research Director            160
Human Resources              104
Name: JobRole, dtype: int64
------------------------------------------------------------------
MaritalStatus:['Single' 'Married' 'Divorced']
Married     1346
Single       940
Divorced     654
Name: MaritalStatus, dtype: int64
------------------------------------------------------------------
Over18:['Y']
Y    2940
Name: Over18, dtype: int64
------------------------------------------------------------------
OverTime:['Yes' 'No']
No     2108
Yes     832
Name: OverTime, dtype: int64
------------------------------------------------------------------
```

In [18]:

```
dataset['EmployeeNumber'].unique()
```

Out[18]:

```
array([   1,    2,    3, ..., 2938, 2939, 2940], dtype=int64)
```

In [20]:

```
dataset['StandardHours'].unique()
```

Out[20]:

```
array([80], dtype=int64)
```

In [21]:

```
dataset = dataset.drop('EmployeeCount',axis = 1)
dataset = dataset.drop('EmployeeNumber',axis = 1)
dataset = dataset.drop('StandardHours',axis = 1)
```

In [22]:

```
dataset.head()
```

Out[22]:

	Attrition	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	...	PerformanceRating	RelationshipSatisfactio
0	Yes	41	Travel_Rarely	1102	Sales	1	2	Life Sciences	2	Female	...	3	
1	No	49	Travel_Frequently	279	Research & Development	8	1	Life Sciences	3	Male	...	4	
2	Yes	37	Travel_Rarely	1373	Research & Development	2	2	Other	4	Male	...	3	
3	No	33	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	4	Female	...	3	
4	No	27	Travel_Rarely	591	Research & Development	2	1	Medical	1	Male	...	3	

5 rows × 32 columns

In [25]:

```
dataset
```

Out[25]:

	Attrition	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	...	PerformanceRating	RelationshipSatisfa
0	Yes	41	Travel_Rarely	1102	Sales	1	2	Life Sciences	2	Female	...	3	
1	No	49	Travel_Frequently	279	Research & Development	8	1	Life Sciences	3	Male	...	4	
2	Yes	37	Travel_Rarely	1373	Research & Development	2	2	Other	4	Male	...	3	
3	No	33	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	4	Female	...	3	
4	No	27	Travel_Rarely	591	Research & Development	2	1	Medical	1	Male	...	3	
...	...	...	...	...	...	...	...	...	...	...	...	...	
2935	No	36	Travel_Frequently	884	Research & Development	23	2	Medical	3	Male	...	3	
2936	No	39	Travel_Rarely	613	Research & Development	6	1	Medical	4	Male	...	3	
2937	No	27	Travel_Rarely	155	Research & Development	4	3	Life Sciences	2	Male	...	4	
2938	No	49	Travel_Frequently	1023	Sales	2	3	Medical	4	Male	...	3	
2939	No	34	Travel_Rarely	628	Research & Development	8	3	Medical	2	Male	...	3	

2940 rows × 31 columns

In [26]:

```
dataset.corr()
```

Out[26]:

	Age	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	JobSatisfaction	MonthlyIncome	...	Perfo
Age	1.000000	0.010661	-0.001686	0.208034	0.010146	0.024287	0.029820	0.509604	-0.004892	0.497855	...	
DailyRate	0.010661	1.000000	-0.004985	-0.016806	0.018355	0.023381	0.046135	0.002966	0.030571	0.007707	...	
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	-0.016075	0.031131	0.008783	0.005303	-0.003669	-0.017014	...	
Education	0.208034	-0.016806	0.021042	1.000000	-0.027128	0.016775	0.042438	0.101589	-0.011296	0.094961	...	

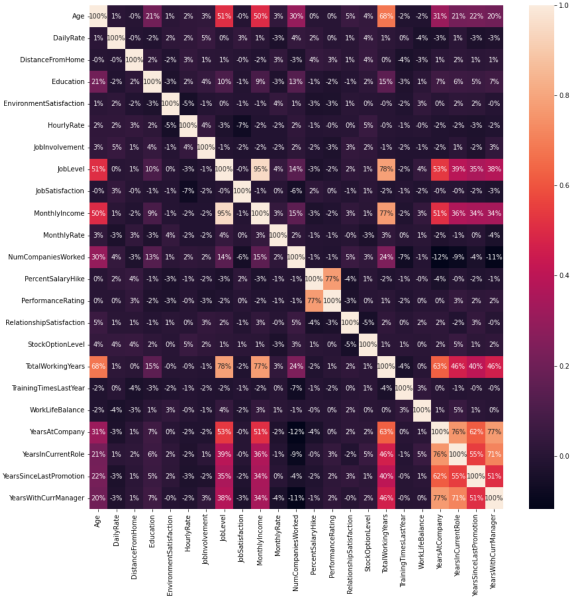
```
 EnvironmentSatisfaction   0.010146   0.018355   -0.016075  -0.027128    1.000000  -0.049857  -0.008278   0.001212  -0.006784  -0.006259  ...
              HourlyRate   0.024287   0.023381    0.031131   0.016775   -0.049857   1.000000   0.042861  -0.027853  -0.071335  -0.015794  ...
          JobInvolvement   0.029820   0.046135    0.008783   0.042438   -0.008278   0.042861   1.000000  -0.012630  -0.021476  -0.015271  ...
                JobLevel   0.509604   0.002966    0.005303   0.101589    0.001212  -0.027853  -0.012630   1.000000  -0.001944   0.950300  ...
         JobSatisfaction  -0.004892   0.030571   -0.003669  -0.011296   -0.006784  -0.071335  -0.021476  -0.001944   1.000000  -0.007157  ...
           MonthlyIncome   0.497855   0.007707   -0.017014   0.094961   -0.006259  -0.015794  -0.015271   0.950300  -0.007157   1.000000  ...
             MonthlyRate   0.028051  -0.032182    0.027473  -0.026084    0.037600  -0.015297  -0.016322   0.039563   0.000644   0.034814  ...
      NumCompaniesWorked   0.299635   0.038153   -0.029251   0.126317    0.012594   0.022157   0.015012   0.142501  -0.055699   0.149515  ...
       PercentSalaryHike   0.003634   0.022704    0.040235  -0.011111   -0.031701  -0.009062  -0.017205  -0.034730   0.020002  -0.027269  ...
       PerformanceRating   0.001904   0.000473    0.027110  -0.024539   -0.029548  -0.002172  -0.029071  -0.021222   0.002297  -0.017120  ...
RelationshipSatisfaction   0.053535   0.007846    0.006557  -0.009118    0.007665   0.001330   0.034297   0.021642  -0.012454   0.025873  ...
        StockOptionLevel   0.037510   0.042143    0.044872   0.018422    0.003432   0.050263   0.021523   0.013984   0.010690   0.005408  ...
       TotalWorkingYears   0.680381   0.014515    0.004628   0.148280   -0.002693  -0.002334  -0.005533   0.782208  -0.020185   0.772893  ...
   TrainingTimesLastYear  -0.019621   0.002453   -0.036942  -0.025100   -0.019359  -0.008548  -0.015338  -0.018191  -0.005779  -0.021736  ...
         WorkLifeBalance  -0.021490  -0.037848   -0.026556   0.009819    0.027627  -0.004607  -0.014617   0.037818  -0.019459   0.030683  ...
          YearsAtCompany   0.311309  -0.034055    0.009508   0.069114    0.001458  -0.019582  -0.021355   0.534739  -0.003803   0.514285  ...
      YearsInCurrentRole   0.212901   0.009932    0.018845   0.060236    0.018007  -0.024106   0.008717   0.389447  -0.002305   0.363818  ...
 YearsSinceLastPromotion   0.216513  -0.033229    0.010029   0.054254    0.016194  -0.026716  -0.024184   0.353885  -0.018214   0.344978  ...
    YearsWithCurrManager   0.202089  -0.026363    0.014406   0.069065   -0.004999  -0.020123   0.025976   0.375281  -0.027656   0.344079  ...

23 rows × 23 columns
```

In [27]:

```
plt.figure(figsize=(14,14))
sns.heatmap(dataset.corr(),annot = True, fmt='.0%')
```

Out[27]:

```
<AxesSubplot:>
```



In [28]:

```
from sklearn.preprocessing import LabelEncoder
for column in dataset.columns:
    if dataset[column].dtype == np.number:
        continue
    dataset[column]= LabelEncoder().fit_transform(dataset[column])
```

```
C:\Users\PC\AppData\Local\Temp/ipykernel_9188/1332740784.py:3: DeprecationWarning: Converting `np.inexact` or `np.floating` to a dtype is deprecate
d. The current result is `float64` which is not strictly correct.
  if dataset[column].dtype == np.number:
```

In [29]:

```
dataset
```

Out[29]:

	Attrition	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	...	PerformanceRating	RelationshipSatisfact
0	1	23	2	624	2	0	1	1	1	0	...	0	
1	0	31	1	113	1	7	0	1	2	1	...	1	
2	1	19	2	805	1	1	1	4	3	1	...	0	
3	0	15	1	820	1	2	3	1	3	0	...	0	
4	0	9	2	312	1	1	0	3	0	1	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
2935	0	18	1	494	1	22	1	3	2	1	...	0	
2936	0	21	2	327	1	5	0	3	3	1	...	0	
2937	0	9	2	39	1	3	2	1	1	1	...	1	
2938	0	31	1	579	2	1	2	3	3	1	...	0	
2939	0	16	2	336	1	7	2	3	1	1	...	0	

2940 rows × 31 columns

In [33]:

```
dataset['Age_Years']=dataset['Age']
dataset=dataset.drop('Age', axis = 1)
```

In [34]:

```
dataset
```

Out[34]:

	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	HourlyRate	...	RelationshipSatisfaction	StockOp
0	1	2	624	2	0	1	1	1	0	64	...	0	
1	0	1	113	1	7	0	1	2	1	31	...	3	
2	1	2	805	1	1	1	4	3	1	62	...	1	
3	0	1	820	1	2	3	1	3	0	26	...	2	
4	0	2	312	1	1	0	3	0	1	10	...	3	
...	...	...	...	...	...	...	...	...	...	...	...	...	
2935	0	1	494	1	22	1	3	2	1	11	...	2	
2936	0	2	327	1	5	0	3	3	1	12	...	0	
2937	0	2	39	1	3	2	1	1	1	57	...	1	
2938	0	1	579	2	1	2	3	3	1	33	...	3	
2939	0	2	336	1	7	2	3	1	1	52	...	0	

2940 rows × 31 columns

In [35]:

```
x = dataset.iloc[:,1:dataset.shape[1]].values
y = dataset.iloc[:,0].values
print(x.shape)
```

```
(2940, 30)
```

In [36]:

```
print(y.shape)
```

```
(2940,)
```

In [37]:

```
print(dataset.shape)
```

```
(2940, 31)
```

In [38]:

```
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=42)
print(x_train.shape,x_test.shape)
```

```
(2352, 30) (588, 30)
```

In [39]:

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators=10, criterion = 'entropy', random_state=0)
forest.fit(x_train,y_train)
```

Out[39]:

```
RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)
```

In [40]:

```
print(x_train[0:5])
```

```
[[   2  380    2    6    2    1    2    1   14    1    2    3    3    0
  1143 1345    9    0   10    1    2    1   10    3    1    5    4    0
     1   13]
 [   2  102    2   20    3    1    3    1   44    3    1    7    3    2
   719 1033    1    0    1    0    3    0   10    3    2   10    7    5
     7   16]
 [   2  492    2    2    3    1    3    1   62    2    2    7    3    0
   962  278    6    0    2    0    1    1   12    2    2   10    9    0
     8   17]
 [   2  788    1   25    3    1    0    1   50    2    1    0    2    1
   722  315    6    0    3    0    1    2   10    2    1    3    2    0
     2   18]
 [   2   68    2    1    0    5    2    1   42    2    0    8    1    1
   185  889    1    0    7    0    0    1    1    3    2    1    0    0
     0   12]]
```

In [42]:

```
score = forest.score(x_train,y_train)
print('randomforest classifier', np.abs(score)*100)
```

```
randomforest classifier 99.44727891156462
```

RANDOMFOREST CLASSIFIER SCORE- 99.44727891156462

In [45]:

```
#TO CHECK MODEL ACCU
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, forest.predict(x_test))

TN = cm[0][0]
TP = cm[1][1]
FN = cm[1][0]
FP = cm[0][1]

print(cm)
print('Model Testing Accuracy is - ',(TP+TN)/(TP+TN+FN+FP) )
```

```
[[489   0]
 [ 29  70]]
Model Testing Accuracy is -  0.9506802721088435
```

In [47]:

```
print('Misclassification Rate -',(FP+FN)/(TP+TN+FP+FN))
print('Precision Rate-',(TP)/(TP+FP))
print('Recall Rate',(TP)/(TP+FN))
```

```
Misclassification Rate - 0.04931972789115646
Precision Rate- 1.0
Recall Rate 0.7070707070707071
```

Precision Rate- 1.0

PREPARED BY : PRATHIBA.M

In [ ]:

```

```