

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
dataset=pd.read_csv("/content/WA_Fn-UseC_-HR-Employee-Attrition.csv")
dataset.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	.
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	

5 rows × 35 columns

```
dataset.shape
```

```
(1470, 35)
```

```
dataset.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
```

```
dataset.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000		1470.000000	1470.000000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306		2.721769	65.891156
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335		1.093082	20.329428
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000		1.000000	30.000000
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000		2.000000	48.000000
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000		3.000000	66.000000
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000		4.000000	83.750000
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000		4.000000	100.000000

8 rows × 26 columns

corr=dataset.corr()

corr

```
<ipython-input-6-f22ca9e9dc13>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version
corr=dataset.corr()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement
Age	1.000000	0.010661	-0.001686	0.208034	NaN	-0.010145		0.010146	0.0
DailyRate	0.010661	1.000000	-0.004985	-0.016806	NaN	-0.050990		0.018355	0.0
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	NaN	0.032916		-0.016075	0.0
Education	0.208034	-0.016806	0.021042	1.000000	NaN	0.042070		-0.027128	0.0
EmployeeCount	NaN	NaN	NaN	NaN	NaN	NaN		NaN	NaN
EmployeeNumber	-0.010145	-0.050990	0.032916	0.042070	NaN	1.000000		0.017621	0.0
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	NaN	0.017621		1.000000	-0.0
HourlyRate	0.024287	0.023381	0.031131	0.016775	NaN	0.035179		-0.049857	1.0
JobInvolvement	0.029820	0.046135	0.008783	0.042438	NaN	-0.006888		-0.008278	0.0
JobLevel	0.509604	0.002966	0.005303	0.101589	NaN	-0.018519		0.001212	-0.0
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	NaN	-0.046247		-0.006784	-0.0
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	NaN	-0.014829		-0.006259	-0.0
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	NaN	0.012648		0.037600	-0.0
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	NaN	-0.001251		0.012594	0.0
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	NaN	-0.012944		-0.031701	-0.0
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	NaN	-0.020359		-0.029548	-0.0
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	NaN	-0.069861		0.007665	0.0
StandardHours	NaN	NaN	NaN	NaN	NaN	NaN		NaN	NaN
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	NaN	0.062227		0.003432	0.0
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280	NaN	-0.014365		-0.002693	-0.0
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100	NaN	0.023603		-0.019359	-0.0
WorkLifeBalance	-0.021490	-0.037848	-0.026556	0.009819	NaN	0.010309		0.027627	-0.0
YearsAtCompany	0.311309	-0.034055	0.009508	0.069114	NaN	-0.011240		0.001458	-0.0
YearsInCurrentRole	0.212901	0.009932	0.018845	0.060236	NaN	-0.008416		0.018007	-0.0
YearsSinceLastPromotion	0.216513	-0.033229	0.010029	0.054254	NaN	-0.009019		0.016194	-0.0
YearsWithCurrManager	0.202089	-0.026363	0.014406	0.069065	NaN	-0.009197		-0.004999	-0.0

26 rows × 26 columns

dataset.isnull().any()

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

```

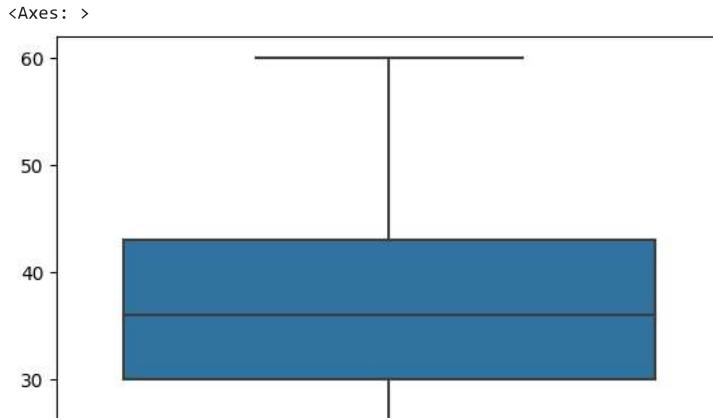
```
dataset.isnull().sum()
```

```

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

```

```
sns.boxplot(dataset.Age)
```



```
dataset.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	.
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

5 rows × 35 columns

```
#years at company=>16
x=dataset.iloc[:,2:]
y=dataset.iloc[:,1:2]
```

```
x.head()
```

	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatis-
0	Travel_Rarely	1102	Sales		1	2	Life Sciences	1	1
1	Travel_Frequently	279	Research & Development		8	1	Life Sciences	1	2
2	Travel_Rarely	1373	Research & Development		2	2	Other	1	4
3	Travel_Frequently	1392	Research & Development		3	4	Life Sciences	1	5
4	Travel_Rarely	591	Research & Development		2	1	Medical	1	7

5 rows × 33 columns

```
y.head()
```

	Attrition	grid
0	Yes	blue
1	No	
2	Yes	
3	No	
4	No	

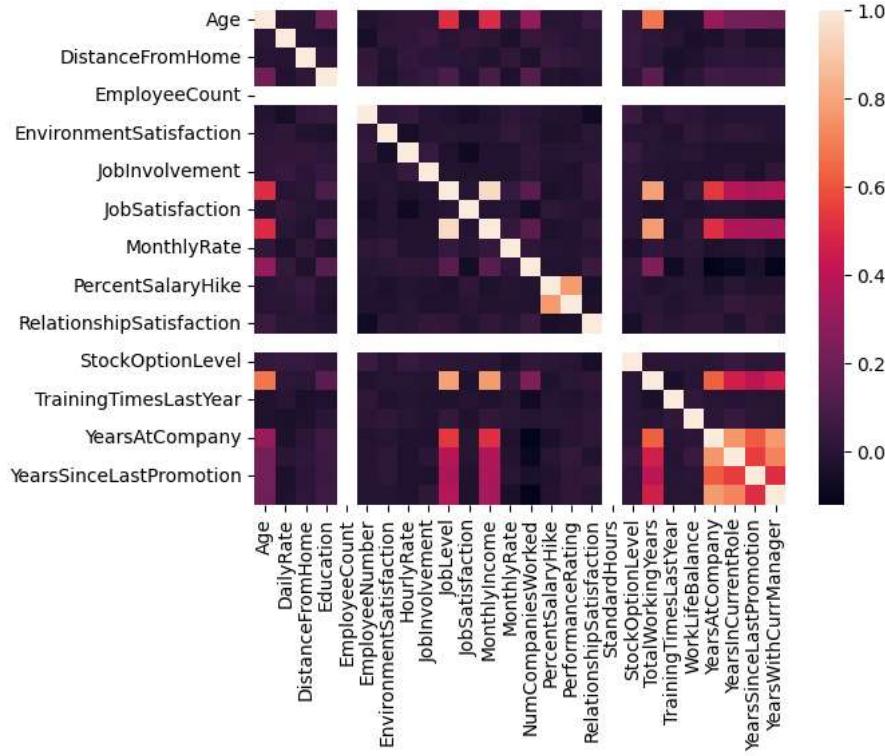
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
y_attrition = le.fit_transform(y_Attrition)
y.head()
```

Attrition	
0	1
1	0
2	1
3	0
4	0

```
sns.heatmap(dataset.corr())
```

```
<ipython-input-16-aa6664222663>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version
  sns.heatmap(dataset.corr())
<Axes: >
```



```
dataset.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	.
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

5 rows × 35 columns

```
x.head()
```

	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatis-
0	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	
1	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	
2	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	
3	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	
4	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	

```
x.BusinessTravel.value_counts()
```

```
Travel_Rarely      1043
Travel_Frequently   277
Non-Travel         150
Name: BusinessTravel, dtype: int64
```

```
from matplotlib.image import BboxTransformTo  
bt=pd.get_dummies(x[["BusinessTravel"]],drop first=True)
```

bt

	Travel_Frequently	Travel_Rarely
0	0	1
1	1	0
2	0	1
3	1	0
4	0	1
...
1465	1	0
1466	0	1
1467	0	1
1468	1	0
1469	0	1

1470 rows × 2 columns

```
x=pd.concat([x,bt],axis=1)
```

x.head()

	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatis-
0	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	
1	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	
2	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	
3	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	
4	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	

5 rows x 35 columns

```
x.drop(["BusinessTravel"],axis=1,inplace=True)
```

x.head()

	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender
0	1102	Sales		1	2	Life Sciences	1	1	2 Female
1	279	Research & Development		8	1	Life Sciences	1	2	3 Male
2	1373	Research & Development		2	2	Other	1	4	4 Male
3	1392	Research & Development		3	4	Life Sciences	1	5	4 Female
4	591	Research & Development		2	1	Medical	1	7	1 Male

5 rows × 34 columns

x.Department.value_counts()

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

```
from matplotlib.image import BboxTransformTo
dep=pd.get_dummies(x["Department"],drop_first=True)
x=pd.concat([x,dep],axis=1)
x.drop(["Department"],axis=1,inplace=True)
```

x.head()

	DailyRate	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement
0	1102	1	2	Life Sciences	1	1	2	Female	94	94
1	279	8	1	Life Sciences	1	2	3	Male	61	61
2	1373	2	2	Other	1	4	4	Male	92	92
3	1392	3	4	Life Sciences	1	5	4	Female	56	56
4	591	2	1	Medical	1	7	1	Male	40	40

5 rows × 35 columns

```
from matplotlib.image import BboxTransformTo
gen=pd.get_dummies(x["Gender"],drop_first=True)
x=pd.concat([x,gen],axis=1)
x.drop(["Gender"],axis=1,inplace=True)
x.head()
```

	DailyRate	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement
0	1102	1	2	Life Sciences	1	1	2	94	94
1	279	8	1	Life Sciences	1	2	3	61	61
2	1373	2	2	Other	1	4	4	92	92
3	1392	3	4	Life Sciences	1	5	4	56	56
4	591	2	1	Medical	1	7	1	40	40

5 rows × 35 columns

```
from matplotlib.image import BboxTransformTo
edu=pd.get_dummies(x["EducationField"],drop_first=True)
x=pd.concat([x,edu],axis=1)
x.drop(["EducationField"],axis=1,inplace=True)
x.head()
```

FromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel
1	2	1	1		2	94	3
8	1	1	2		3	61	2
2	2	1	4		4	92	2
3	4	1	5		4	56	3
2	1	1	7		1	40	3
							1

```
from matplotlib.image import BboxTransformTo
jr=pd.get_dummies(x["JobRole"],drop_first=True)
x=pd.concat([x,jr],axis=1)
x.drop(["JobRole"],axis=1,inplace=True)
x.head()
```

	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel
0	1102	1	2	1	1		2	94	3
1	279	8	1	1	2		3	61	2
2	1373	2	2	1	4		4	92	2
3	1392	3	4	1	5		4	56	3
4	591	2	1	1	7		1	40	3

5 rows × 46 columns

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

```
x_train.shape,x_test.shape,y_train.shape,y_test.shape
((1029, 46), (441, 46), (1029, 1), (441, 1))
```

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
```

```
x_train=sc.fit_transform(x_train)
x_test=sc.fit_transform(x_test)
x_train
array([[-0.58244694, -0.51036458,  0.06512323, ..., -0.90267093,
       -0.69784971, -0.61441846],
       [-1.12354988, -0.51036458,  0.06512323, ..., -0.90267093,
        1.43297329,  1.62755527],
       [-0.04626313,  1.84630474,  1.06529941, ..., -0.90267093,
        1.43297329, -0.61441846],
       ...,
       [ 0.91542436, -0.88247026,  0.06512323, ...,  1.10782342,
       -0.69784971,  1.62755527],
       [-1.3252337 , -0.88247026,  2.0654756 , ...,  1.10782342,
       -0.69784971,  1.62755527],
       [-0.36600577,  0.10981156,  1.06529941, ..., -0.90267093,
       -0.69784971, -0.61441846]])
```

```
x.head()
```

	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel
0	1102	1	2	1	1	2	94	3	2
1	279	8	1	1	2	3	61	2	2
2	1373	2	2	1	4	4	92	2	1

dataset.head()

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	.
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	

5 rows × 35 columns

x_train.head()

	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel
338	570	5	3	1	456	4	30	2	
363	350	5	3	1	485	4	34	3	
759	788	24	4	1	1049	2	36	3	
793	895	15	2	1	1102	1	50	3	
581	921	1	3	1	806	4	38	1	

5 rows × 46 columns

dataset.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

```

```
x.MaritalStatus.value_counts()
```

```

Married      673
Single       470
Divorced     327
Name: MaritalStatus, dtype: int64

```

```

from matplotlib.image import BboxTransformTo
ms=pd.get_dummies(x["MaritalStatus"],drop_first=True)
x=pd.concat([x,ms],axis=1)
x.drop(["MaritalStatus"],axis=1,inplace=True)
x.head()

```

```
x.Over18.value_counts()
```

```

Y      1470
Name: Over18, dtype: int64

```

```

from matplotlib.image import BboxTransformTo
o=pd.get_dummies(x["Over18"],drop_first=True)
x=pd.concat([x,o],axis=1)
x.drop(["Over18"],axis=1,inplace=True)
x.head()

```

	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel
0	1102	1	2	1	1	2	94	3	2
1	279	8	1	1	2	3	61	2	2
2	1373	2	2	1	4	4	92	2	1
3	1392	3	4	1	5	4	56	3	1
4	591	2	1	1	7	1	40	3	1

5 rows × 46 columns

```

from matplotlib.image import BboxTransformTo
ot=pd.get_dummies(x["OverTime"],drop_first=True)
x=pd.concat([x,ot],axis=1)
x.drop(["OverTime"],axis=1,inplace=True)
x.head()

```

	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel
0	1102	1	2	1	1	2	94	3	2
1	279	8	1	1	2	3	61	2	2
2	1373	2	2	1	4	4	92	2	1

```
from sklearn.preprocessing import MinMaxScaler  
ms=MinMaxScaler()  
x_scaled=pd.DataFrame(ms.fit_transform(x),columns=x.columns)
```

2 ROWS X 40 COLUMNS

x scaled

	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel
0	0.715820	0.000000	0.25	0.0	0.000000	0.333333	0.914286	0.666667	0
1	0.126700	0.250000	0.00	0.0	0.000484	0.666667	0.442857	0.333333	0
2	0.909807	0.035714	0.25	0.0	0.001451	1.000000	0.885714	0.333333	0
3	0.923407	0.071429	0.75	0.0	0.001935	1.000000	0.371429	0.666667	0
4	0.350036	0.035714	0.00	0.0	0.002903	0.000000	0.142857	0.666667	0
...
1465	0.559771	0.785714	0.25	0.0	0.996613	0.666667	0.157143	1.000000	0
1466	0.365784	0.178571	0.00	0.0	0.997097	1.000000	0.171429	0.333333	0
1467	0.037938	0.107143	0.50	0.0	0.998065	0.333333	0.814286	1.000000	0
1468	0.659270	0.035714	0.50	0.0	0.998549	1.000000	0.471429	0.333333	0
1469	0.376521	0.250000	0.50	0.0	1.000000	0.333333	0.742857	1.000000	0

1470 rows × 46 columns

```
from sklearn.linear_model import LogisticRegression  
model=LogisticRegression()  
model.fit(x_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143: DataConversionWarning: A column-vector y was passed when a 1d  
y = column_or_1d(y, warn=True)  
v LogisticRegression  
LogisticRegression()
```

```
pred=model.predict(x_test)  
pred
```

```
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,roc_auc_score,roc_curve  
accuracy_score(y_test,pred)
```

```
0.8752834467120182
```

```
confusion_matrix(y_test,pred)
```

```
array([[359,  12],  
       [ 43,  27]])
```

```
print(classification_report(y_test,pred))
```

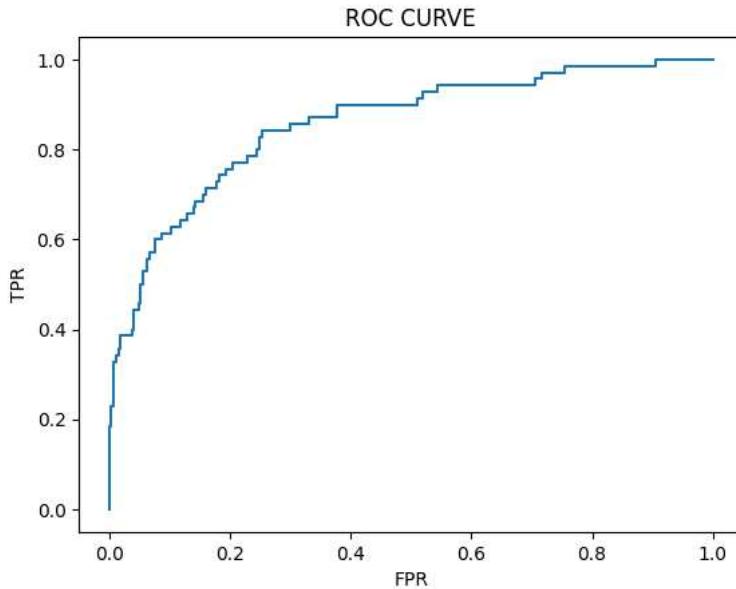
	precision	recall	f1-score	support
0	0.89	0.97	0.93	371
1	0.69	0.39	0.50	70
accuracy			0.88	441
macro avg	0.79	0.68	0.71	441
weighted avg	0.86	0.88	0.86	441

```
probability=model.predict_proba(x_test)[:,1]  
probability
```

```
8.37792343e-02, 3.04262398e-01, 2.01306657e-01, 4.07001454e-03,  
1.37910530e-02, 5.82290333e-02, 2.04020196e-02, 1.09034507e-01,  
1.27635361e-01, 4.91038183e-01, 4.58193701e-01, 2.07786042e-01,
```

```
1.3/153890e-02, 1.6/0136/6e-01, 5./955/836e-02, 8.00865140e-05,
6.66266026e-04])
```

```
fpr,tpr,thresholds = roc_curve(y_test,probability)
plt.plot(fpr,tpr)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC CURVE')
plt.show()
```



```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)

(1029, 46)
(441, 46)
(1029, 1)
(441, 1)
```

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

```
    ▾ LinearRegression
    LinearRegression()
```

```
lr.coef_
array([[-1.18123207e-02,  2.89510662e-02,  2.55596622e-03,
       -1.52308721e-15, -7.32435652e-03, -4.33592856e-02,
       -1.99346035e-03, -3.36102665e-02,  1.09364404e-02,
       -3.39152836e-02, -1.18654092e-02,  6.64793644e-03,
       4.00421131e-02, -1.42727327e-02,  1.56891912e-02,
      -2.69926472e-02, -1.17961196e-16, -7.94222398e-03,
      -5.19164933e-02, -1.27287648e-02, -3.24271240e-02,
      4.69961491e-02, -3.55834876e-02,  3.62961008e-02,
      -4.25420370e-02,  5.72744446e-02,  2.95463022e-02,
      1.10684394e-01,  7.75509759e-02,  1.20667697e-02,
     -1.23544342e-01, -5.43399582e-02, -1.27155088e-01,
      -5.18189998e-02, -4.38545929e-02,  4.34125338e-02,
      4.81933375e-02,  1.01790606e-02, -4.71438734e-03,
      5.66884605e-04,  1.08688419e-02,  4.39215065e-02,
      7.15971521e-02,  7.48167229e-03,  5.19746532e-02,
      9.13856176e-02]])
```

```
y_pred=lr.predict(x_test)
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

0.2243832296734597

```
X=dataset.iloc[:, :1]
```

x.head()

Age
0 41
1 49
2 37
3 33
4 27

```
from sklearn.tree import DecisionTreeClassifier  
dtc=DecisionTreeClassifier()
```

```
dtc.fit(x_train,y_train)
```

```
▼ DecisionTreeClassifier  
DecisionTreeClassifier()
```

```
pred=dtc.predict(x_test)  
pred
```

```
dataset.DailyRate.value_counts().plot(kind="pie", autopct="%1.1f%%")
```

```
<Axes: ylabel='DailyRate'>

from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier()
knn.fit(x_train,y_train)

/usr/local/lib/python3.10/dist-packages/sklearn/neighbors/_classification.py:215: DataConversionWarning: A column-vector y was passed w
    return self._fit(X, y)
  KNeighborsClassifier()
KNeighborsClassifier()
```

28/34

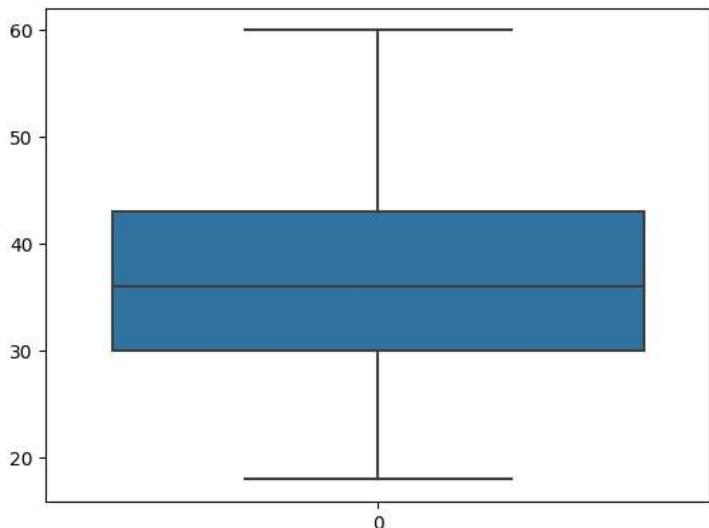
11-12-2021

```
from sklearn.metrics import accuracy_score,classification_report  
accuracy score(y pred,y test)
```

0.8503401360544217

```
sns.boxplot(dataset.Age)
```

<Axes: >



```
sns.pairplot(dataset)
```

