

PROJECT DEVELOPMENT PHASE

SPRINT 1

Date	04 NOVEMBER 2022
Team ID	PNT2022TMID03886
Project Name	Corporate Employee Attrition Analysis

SOURCE CODE:

It is with the understanding of the given set of datasets and fetching the data and cleaning the data with the provided three set of datasets.

These are the results for the various datasets provided:

1. General Data CSV file:

- To fetch the dataset,

```
df1=pd.read_csv('general_data.csv')
```

To view the data by df1

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	...	NumCompaniesWorked	Over18
0	51	No	Travel_Rarely	Sales	6	2	Life Sciences	1	1	Female	...	1.0	Y
1	31	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences	1	2	Female	...	0.0	Y
2	32	No	Travel_Frequently	Research & Development	17	4	Other	1	3	Male	...	1.0	Y
3	38	No	Non-Travel	Research & Development	2	5	Life Sciences	1	4	Male	...	3.0	Y
4	32	No	Travel_Rarely	Research & Development	10	1	Medical	1	5	Male	...	4.0	Y
...
4405	42	No	Travel_Rarely	Research & Development	5	4	Medical	1	4406	Female	...	3.0	Y
4406	29	No	Travel_Rarely	Research & Development	2	4	Medical	1	4407	Male	...	2.0	Y
4407	25	No	Travel_Rarely	Research & Development	25	2	Life Sciences	1	4408	Male	...	0.0	Y
4408	42	No	Travel_Rarely	Sales	18	2	Medical	1	4409	Male	...	0.0	Y
4409	40	No	Travel_Rarely	Research & Development	28	3	Medical	1	4410	Male	...	0.0	Y

4410 rows x 24 columns

To view the various datatypes of the respected columns given in the dataset that is been loaded to the df1.

```
df1.dtypes
```

Age	int64
Attrition	object
BusinessTravel	object
Department	object
DistanceFromHome	int64
Education	int64
EducationField	object
EmployeeCount	int64
EmployeeID	int64
Gender	object
JobLevel	int64
JobRole	object
MaritalStatus	object
MonthlyIncome	int64
NumCompaniesWorked	float64
Over18	object
PercentSalaryHike	int64
StandardHours	int64
StockOptionLevel	int64
TotalWorkingYears	float64
TrainingTimesLastYear	int64
YearsAtCompany	int64
YearsSinceLastPromotion	int64
YearsWithCurrManager	int64
dtype:	object

To describe the information of the dataset that is fetched.

```
df1.describe()
```

	Age	DistanceFromHome	Education	EmployeeCount	EmployeeID	JobLevel	MonthlyIncome	NumCompaniesWorked	PercentSalaryHike	StandardHours	StockOptionLevel
count	4410.000000	4410.000000	4410.000000	4410.0	4410.000000	4410.000000	4410.000000	4391.000000	4410.000000	4410.0	4410.0
mean	36.923810	9.192517	2.912925	1.0	2205.500000	2.063946	65029.312925	2.694830	15.209524	8.0	0.0
std	9.133301	8.105026	1.023933	0.0	1273.201673	1.106689	47068.888559	2.498887	3.659108	0.0	0.0
min	18.000000	1.000000	1.000000	1.0	1.000000	1.000000	10090.000000	0.000000	11.000000	8.0	0.0
25%	30.000000	2.000000	2.000000	1.0	1103.250000	1.000000	29110.000000	1.000000	12.000000	8.0	0.0
50%	36.000000	7.000000	3.000000	1.0	2205.500000	2.000000	49190.000000	2.000000	14.000000	8.0	0.0
75%	43.000000	14.000000	4.000000	1.0	3307.750000	3.000000	83800.000000	4.000000	18.000000	8.0	0.0
max	60.000000	29.000000	5.000000	1.0	4410.000000	5.000000	199990.000000	9.000000	25.000000	8.0	0.0

To check the null values if present:

```
df1.isnull().sum()
```

Age	0
Attrition	0
BusinessTravel	0
Department	0
DistanceFromHome	0
Education	0
EducationField	0
EmployeeCount	0
EmployeeID	0
Gender	0
JobLevel	0
JobRole	0
MaritalStatus	0
MonthlyIncome	0
NumCompaniesWorked	19
Over18	0
PercentSalaryHike	0
StandardHours	0
StockOptionLevel	0
TotalWorkingYears	9
TrainingTimesLastYear	0
YearsAtCompany	0
YearsSinceLastPromotion	0
YearsWithCurrManager	0
dtype: int64	

Here there are some null values in the number of Companies worked that are cleaned and the data after cleaning is referred as;

```
[14] df1['NumCompaniesWorked']=df1['NumCompaniesWorked'].fillna(df1['NumCompaniesWorked'].mean())

[15] df1['TotalWorkingYears']=df1['TotalWorkingYears'].fillna(df1['TotalWorkingYears'].mean())

[16] df1.isnull().sum()

Age                                0
Attrition                          0
BusinessTravel                     0
Department                         0
DistanceFromHome                   0
Education                          0
EducationField                     0
EmployeeCount                      0
EmployeeID                         0
Gender                             0
JobLevel                           0
JobRole                             0
MaritalStatus                      0
MonthlyIncome                      0
NumCompaniesWorked                 0
Over18                             0
PercentSalaryHike                  0
StandardHours                      0
StockOptionLevel                   0
TotalWorkingYears                  0
TrainingTimesLastYear              0
YearsAtCompany                     0
YearsSinceLastPromotion             0
YearsWithCurrManager               0
dtype: int64
```

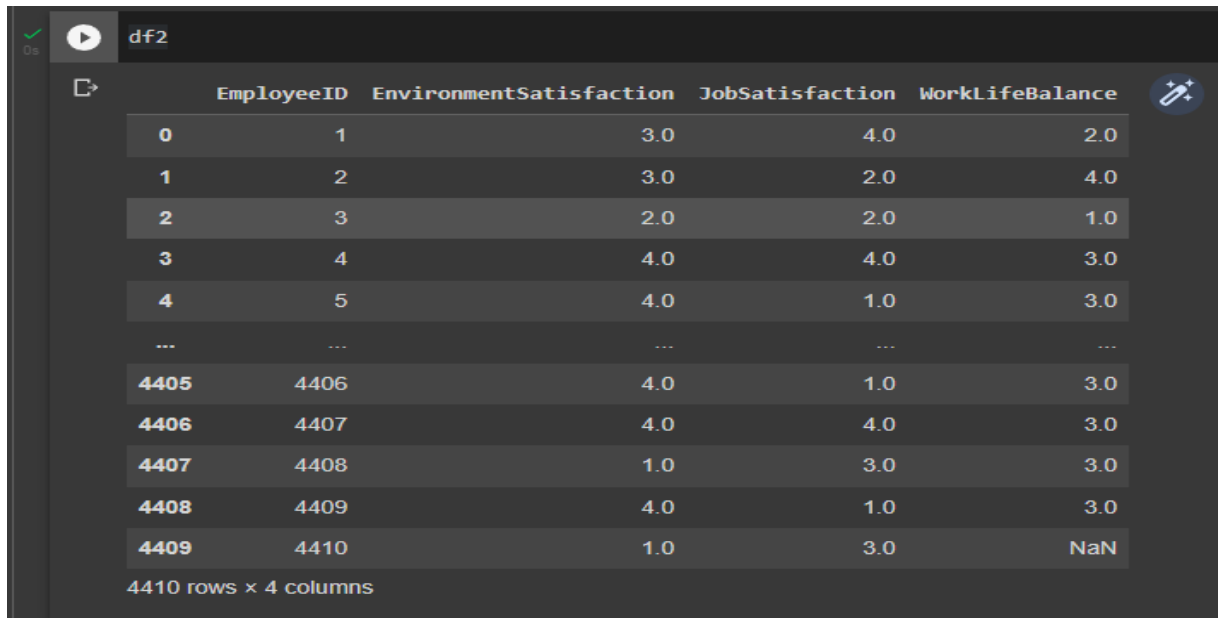
After the correction there is no null values left and the data is cleaned.

2. Employee Survey Data CSV file:

To fetch the dataset,

```
df2 = pd.read_csv('employee_survey_data.csv')
```

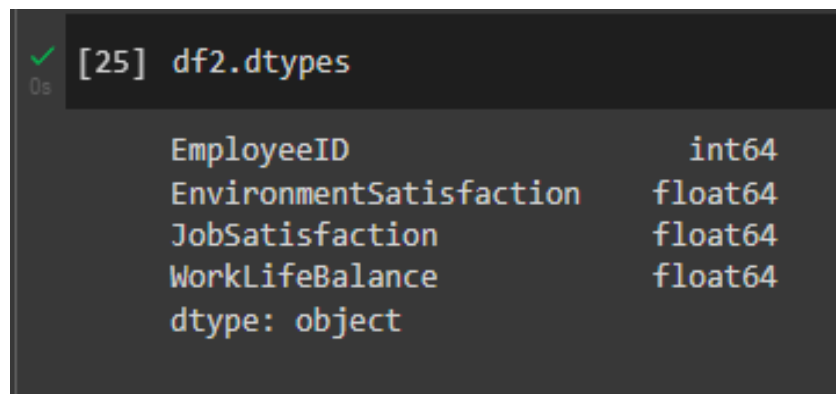
To view the data by df2



	EmployeeID	EnvironmentSatisfaction	JobSatisfaction	WorkLifeBalance
0	1	3.0	4.0	2.0
1	2	3.0	2.0	4.0
2	3	2.0	2.0	1.0
3	4	4.0	4.0	3.0
4	5	4.0	1.0	3.0
...
4405	4406	4.0	1.0	3.0
4406	4407	4.0	4.0	3.0
4407	4408	1.0	3.0	3.0
4408	4409	4.0	1.0	3.0
4409	4410	1.0	3.0	NaN

4410 rows x 4 columns

To view the various datatypes of the respected columns given in the dataset that is been loaded to the df1.



```
[25] df2.dtypes
```

EmployeeID	int64
EnvironmentSatisfaction	float64
JobSatisfaction	float64
WorkLifeBalance	float64
dtype:	object

To describe the information of the dataset that is fetched.



[23] df2.describe()

	EmployeeID	EnvironmentSatisfaction	JobSatisfaction	WorkLifeBalance
count	4410.000000	4385.000000	4390.000000	4372.000000
mean	2205.500000	2.723603	2.728246	2.761436
std	1273.201673	1.092756	1.101253	0.706245
min	1.000000	1.000000	1.000000	1.000000
25%	1103.250000	2.000000	2.000000	2.000000
50%	2205.500000	3.000000	3.000000	3.000000
75%	3307.750000	4.000000	4.000000	3.000000
max	4410.000000	4.000000	4.000000	4.000000

To check the null values if present:

```
df2.isnull().sum()
EmployeeID      0
EnvironmentSatisfaction  25
JobSatisfaction  20
WorkLifeBalance  38
dtype: int64
```

Here there are some null values in the number of Companies worked that are cleaned and the data after cleaning is referred as;

```
[27] df2['EnvironmentSatisfaction']=df2['EnvironmentSatisfaction'].fillna(df2['EnvironmentSatisfaction'].mean())
[28] df2['JobSatisfaction']=df2['JobSatisfaction'].fillna(df2['JobSatisfaction'].mean())
[29] df2['WorkLifeBalance']=df2['WorkLifeBalance'].fillna(df2['WorkLifeBalance'].mean())
[30] df2.isnull().sum()
EmployeeID      0
EnvironmentSatisfaction  0
JobSatisfaction  0
WorkLifeBalance  0
dtype: int64
```

After the correction there is no null values left and the data is cleaned.

3. Manager Survey Data CSV file:

To fetch the dataset,

```
df3=pd.read_csv('manager_survey_data.csv')
```

To view the data by df3

```
df3
EmployeeID  JobInvolvement  PerformanceRating
0           1                3                    3
1           2                2                    4
2           3                3                    3
3           4                2                    3
4           5                3                    3
...         ...            ...                    ...
4405        4406                3                    3
4406        4407                2                    3
4407        4408                3                    4
4408        4409                2                    3
4409        4410                4                    3
4410 rows x 3 columns
```

To view the various datatypes of the respected columns given in the dataset that is been loaded to the df1.

```
✓ 0s df3.dtypes
EmployeeID      int64
JobInvolvement  int64
PerformanceRating int64
dtype: object
```

To describe the information of the dataset that is fetched.

```
✓ 0s df3.describe()
```

	EmployeeID	JobInvolvement	PerformanceRating
count	4410.000000	4410.000000	4410.000000
mean	2205.500000	2.729932	3.153741
std	1273.201673	0.711400	0.360742
min	1.000000	1.000000	3.000000
25%	1103.250000	2.000000	3.000000
50%	2205.500000	3.000000	3.000000
75%	3307.750000	3.000000	3.000000
max	4410.000000	4.000000	4.000000

To check the null values if present:

```
✓ 0s [39] df3.isnull().sum()
EmployeeID      0
JobInvolvement  0
PerformanceRating 0
dtype: int64
```

As there is no null values in the columns of the dataset the data fetched is already cleaned.