Evaluation Project 2 HR Analytics Project- Understanding the Attrition in HR

Problem Statement:

Every year a lot of companies hire a number of employees. The companies invest time and money in training those employees, not just this but there are training programs within the companies for their existing employees as well. The aim of these programs is to increase the effectiveness of their employees. But where HR Analytics fit in this? and is it just about improving the performance of employees?

HR Analytics:

Human resource analytics (HR analytics) is an area in the field of analytics that refers to applying analytic processes to the human resource department of an organization in the hope of improving employee performance and therefore getting a better return on investment. HR analytics does not just deal with gathering data on employee efficiency. Instead, it aims to provide insight into each process by gathering data and then using it to make relevant decisions about how to improve these processes.

Attrition in HR:

Attrition in human resources refers to the gradual loss of employees overtime. In general, relatively high attrition is problematic for companies. HR professionals often assume a leadership role in designing company compensation programs, work culture, and motivation systems that help the organization retain top employees.

How does Attrition affect companies? and how does HR Analytics help in analyzing attrition? We will discuss the first question here and for the second question, we will write the code and try to understand the process step by step.

Attrition affecting Companies:

A major problem in high employee attrition is its cost to an organization. Job postings, hiring processes, paperwork, and new hire training are some of the common expenses of losing employees and replacing them. Additionally, regular employee turnover prohibits your organization from increasing its collective knowledge base and experience over time. This is especially concerning if your business is customer-facing, as customers often prefer to interact with familiar people. Errors and issues are more likely if you constantly have new workers.

Note: You can find the dataset in the link below.

Downlaod Files:

https://github.com/dsrscientist/IBM_HR_Attrition_Rate_Analytics

- Importing require library for performing EDA, Data Wrangling and data cleaning

```
import pandas as pd # for data wrangling purpose
import numpy as np # Basic computation library
import seaborn as sns # For Visualization
import matplotlib.pyplot as plt # ploting package
%matplotlib inline
import warnings # Filtering warnings
warnings.filterwarnings('ignore')

# Importing IBM HR dataset Csv file:
df=pd.read_csv('WA_Fn-UseC_-HR-Employee-Attrition.csv')

print('No of Rows:',df.shape[0])
print('No of Columns:',df.shape[1])
pd.set_option('display.max_columns', None) # This will enable us to
see truncated columns
df.head()
```

```
No of Rows: 1470
No of Columns: 35
```

```
   Age Attrition     BusinessTravel  DailyRate              Department
\
0   41       Yes      Travel_Rarely       1102                   Sales

1   49        No  Travel_Frequently        279  Research & Development

2   37       Yes      Travel_Rarely       1373  Research & Development

3   33        No  Travel_Frequently       1392  Research & Development

4   27        No      Travel_Rarely        591  Research & Development


   DistanceFromHome  Education EducationField  EmployeeCount
EmployeeNumber  \
0                 1          2  Life Sciences              1
1
1                 8          1  Life Sciences              1
2
2                 2          2          Other              1
4
3                 3          4  Life Sciences              1
5
4                 2          1        Medical              1
7

   EnvironmentSatisfaction  Gender  HourlyRate  JobInvolvement
JobLevel  \
0                        2  Female          94               3
2
1                        3    Male          61               2
```

```
2
2                       4    Male         92               2
1
3                       4  Female         56               3
1
4                       1    Male         40               3
1

                  JobRole  JobSatisfaction MaritalStatus  MonthlyIncome
\
0         Sales Executive                4        Single           5993

1      Research Scientist                2       Married           5130

2   Laboratory Technician                3        Single           2090

3      Research Scientist                3       Married           2909

4   Laboratory Technician                2       Married           3468


   MonthlyRate  NumCompaniesWorked Over18 OverTime  PercentSalaryHike
\
0        19479                   8      Y      Yes                 11

1        24907                   1      Y       No                 23

2         2396                   6      Y      Yes                 15

3        23159                   1      Y      Yes                 11

4        16632                   9      Y       No                 12


   PerformanceRating  RelationshipSatisfaction  StandardHours  \
0                  3                         1             80
1                  4                         4             80
2                  3                         2             80
3                  3                         3             80
4                  3                         4             80

   StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
0                 0                  8                      0
1                 1                 10                      3
2                 0                  7                      3
3                 0                  8                      3
4                 1                  6                      3

   WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
0                1               6                   4
1                3              10                   7
```

```
2                3              0                  0
3                3              8                  7
4                3              2                  2

   YearsSinceLastPromotion  YearsWithCurrManager
0                        0                     5
1                        1                     7
2                        0                     0
3                        3                     0
4                        2                     2
```

```
df.columns
```

```
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')
```

```
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
```

```
# As we have 35 Columns Lets sort Columns by their datatype
df.columns.to_series().groupby(df.dtypes).groups
```

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

Comment :

- In this HR dataset we have 1470 rows and 35 columns.
- Non-null count is same for all Columns, so it seem that it contain No missing value. Still we need to perform Data integrity Check for null values in form of "-","NA" , any duplicate entry or error in Data.
- Out of 35 we have 9 features with Object datatypes and rest are int64 types

- Among all Numeric Variables 'Education','EnvironmentSatisfaction', 'JobInvolvement', 'JobSatisfaction', 'RelationshipSatisfaction', 'PerformanceRating', 'WorkLifeBalance' are ordinal variable. Unique range of all these ordinal Variable need to check.
- Here We have Target Variable 'Attrition'.
- These Ordinal features come with the following label encoding:
  - Education: 1- 'Below College' , 2 -'College', 3 -'Bachelor', 4- 'Master' ,5 -'Doctor'
  - EnvironmentSatisfaction: 1- 'Low', 2- 'Medium', 3 -'High', 4- 'Very High'
  - JobInvolvement: 1 -'Low', 2- 'Medium', 3- 'High', 4- 'Very High'
  - JobSatisfaction: 1- 'Low', 2- 'Medium', 3- 'High', 4 -'Very High'
  - PerformanceRating: 1- 'Low', 2- 'Average', 3 -'Good', 4- 'Excellent', 5- 'Outstanding'
  - RelationshipSatisfaction: 1- 'Low', 2- 'Medium', 3- 'High', 4- 'Very High'
  - WorkLifeBalance: 1- 'Bad', 2- 'Good', 3- 'Better', 4- 'Best'

Statistical Analysis

Before Going for Statistical exploration of data, first check integrity of data & Missing value

Data Integrity Check

Since dataset is large, Let check for any entry which is repeated or duplicated in dataset.

```
df.duplicated().sum()  # This will check the duplicate data for all columns.
```

```
0
```

Missing value check

```
missing_values = df.isnull().sum().sort_values(ascending = False)
percentage_missing_values =(missing_values/len(df))*100
print(pd.concat([missing_values, percentage_missing_values], axis =1, keys =['Missing Values', '% Missing data']))
```

```
                          Missing Values  % Missing data
Age                                    0             0.0
StandardHours                          0             0.0
NumCompaniesWorked                     0             0.0
Over18                                 0             0.0
OverTime                               0             0.0
PercentSalaryHike                      0             0.0
PerformanceRating                      0             0.0
RelationshipSatisfaction               0             0.0
StockOptionLevel                       0             0.0
MonthlyIncome                          0             0.0
TotalWorkingYears                      0             0.0
```

```
TrainingTimesLastYear                          0             0.0
WorkLifeBalance                                0             0.0
YearsAtCompany                                 0             0.0
YearsInCurrentRole                             0             0.0
YearsSinceLastPromotion                        0             0.0
MonthlyRate                                    0             0.0
MaritalStatus                                  0             0.0
Attrition                                      0             0.0
EmployeeCount                                  0             0.0
BusinessTravel                                 0             0.0
DailyRate                                      0             0.0
Department                                     0             0.0
DistanceFromHome                               0             0.0
Education                                      0             0.0
EducationField                                 0             0.0
EmployeeNumber                                 0             0.0
JobSatisfaction                                0             0.0
EnvironmentSatisfaction                        0             0.0
Gender                                         0             0.0
HourlyRate                                     0             0.0
JobInvolvement                                 0             0.0
JobLevel                                       0             0.0
JobRole                                        0             0.0
YearsWithCurrManager                           0             0.0
```

Comment: Luckily for us, there is no missing data! this will make it easier to work with the dataset.

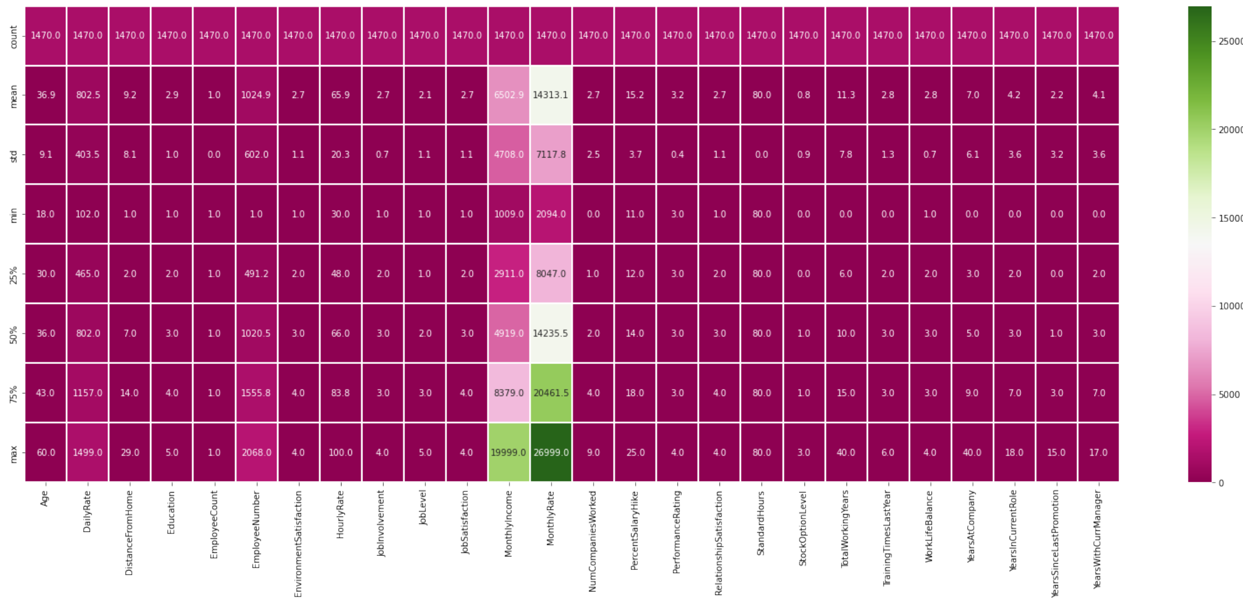
Dataset doesnot contain Any duplicate entry, Missing Values.

So, Yes To Go !!!

Statistical Matrix

```
# Visualizing the statistics of the columns using heatmap.
plt.figure(figsize=(28,10))
sns.heatmap(df.describe(),linewidths = 0.1,fmt='0.1f',annot =
True,cmap='PiYG')
```

```
<AxesSubplot:>
```




```python
df.describe().T.round(3)
```

```
                          count       mean       std     min      25%  \
Age                      1470.0     36.924     9.135    18.0    30.00
DailyRate                1470.0    802.486   403.509   102.0   465.00
DistanceFromHome         1470.0      9.193     8.107     1.0     2.00
Education                1470.0      2.913     1.024     1.0     2.00
EmployeeCount            1470.0      1.000     0.000     1.0     1.00
EmployeeNumber           1470.0   1024.865   602.024     1.0   491.25
EnvironmentSatisfaction  1470.0      2.722     1.093     1.0     2.00
HourlyRate               1470.0     65.891    20.329    30.0    48.00
JobInvolvement           1470.0      2.730     0.712     1.0     2.00
JobLevel                 1470.0      2.064     1.107     1.0     1.00
JobSatisfaction          1470.0      2.729     1.103     1.0     2.00
MonthlyIncome            1470.0   6502.931  4707.957  1009.0  2911.00
MonthlyRate              1470.0  14313.103  7117.786  2094.0  8047.00
NumCompaniesWorked       1470.0      2.693     2.498     0.0     1.00
```

PercentSalaryHike	1470.0	15.210	3.660	11.0	12.00
PerformanceRating	1470.0	3.154	0.361	3.0	3.00
RelationshipSatisfaction	1470.0	2.712	1.081	1.0	2.00
StandardHours	1470.0	80.000	0.000	80.0	80.00
StockOptionLevel	1470.0	0.794	0.852	0.0	0.00
TotalWorkingYears	1470.0	11.280	7.781	0.0	6.00
TrainingTimesLastYear	1470.0	2.799	1.289	0.0	2.00
WorkLifeBalance	1470.0	2.761	0.706	1.0	2.00
YearsAtCompany	1470.0	7.008	6.127	0.0	3.00
YearsInCurrentRole	1470.0	4.229	3.623	0.0	2.00
YearsSinceLastPromotion	1470.0	2.188	3.222	0.0	0.00
YearsWithCurrManager	1470.0	4.123	3.568	0.0	2.00

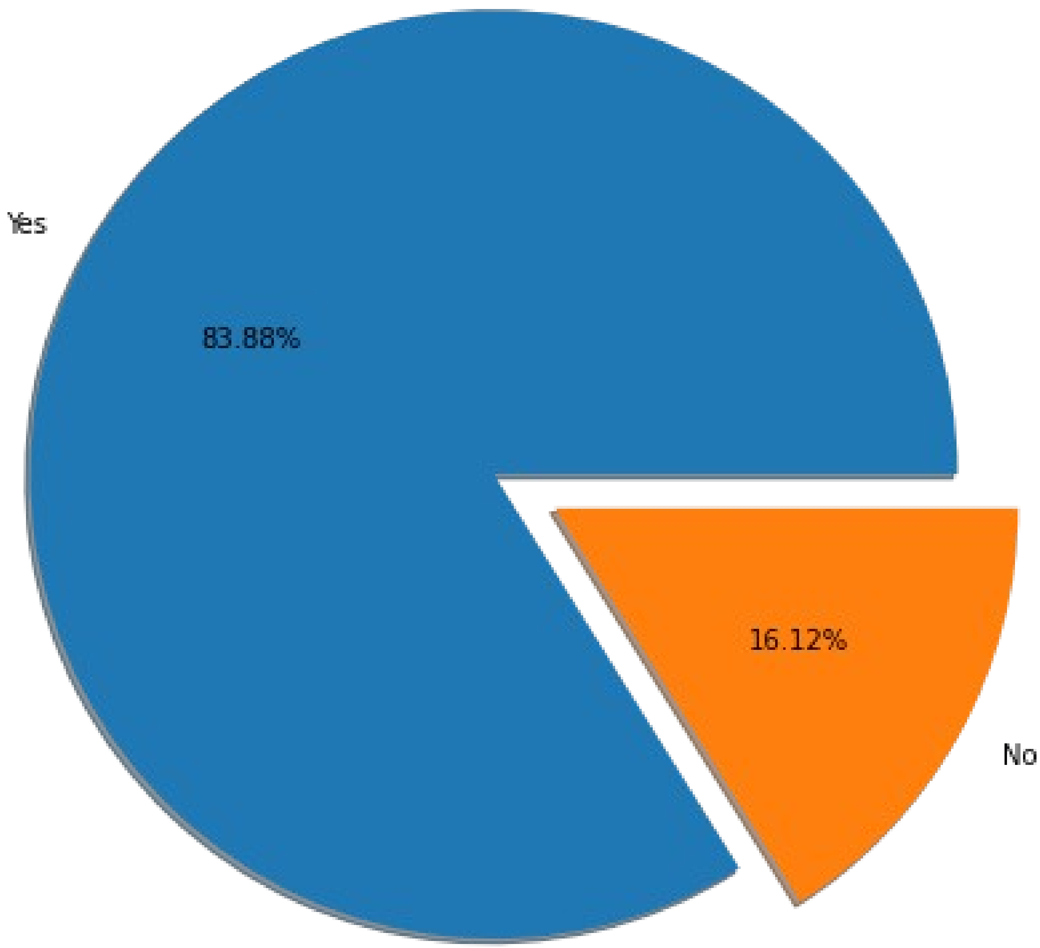
	50%	75%	max
Age	36.0	43.00	60.0
DailyRate	802.0	1157.00	1499.0
DistanceFromHome	7.0	14.00	29.0
Education	3.0	4.00	5.0
EmployeeCount	1.0	1.00	1.0
EmployeeNumber	1020.5	1555.75	2068.0
EnvironmentSatisfaction	3.0	4.00	4.0
HourlyRate	66.0	83.75	100.0
JobInvolvement	3.0	3.00	4.0
JobLevel	2.0	3.00	5.0
JobSatisfaction	3.0	4.00	4.0
MonthlyIncome	4919.0	8379.00	19999.0
MonthlyRate	14235.5	20461.50	26999.0
NumCompaniesWorked	2.0	4.00	9.0
PercentSalaryHike	14.0	18.00	25.0
PerformanceRating	3.0	3.00	4.0
RelationshipSatisfaction	3.0	4.00	4.0
StandardHours	80.0	80.00	80.0
StockOptionLevel	1.0	1.00	3.0
TotalWorkingYears	10.0	15.00	40.0
TrainingTimesLastYear	3.0	3.00	6.0
WorkLifeBalance	3.0	3.00	4.0
YearsAtCompany	5.0	9.00	40.0
YearsInCurrentRole	3.0	7.00	18.0

```
YearsSinceLastPromotion        1.0       3.00      15.0
YearsWithCurrManager           3.0       7.00      17.0
```

Comment :

- Minimum Empolyee Age is 18 and Maximum age of employee 60.
- Average distance from home is 9.1 KM. It means that most of employee travel atleast 18 KM in day from home to office.
- On Average performance Rating of employees is 3.163 with min value 3.0. This Means that performance of most of employee is 'Good'.This implies that Attrition of Employee with 'Outstanding' or 5 rating need to investigate.
- 50% of Employees has worked atleast 2 companies previously.
- For Monthly Income,Monthly Rate by looking at 50% and max column we can say outliers exist in this feature.
- By looking at Mean and Median we see that some of the features are skew in nature.
- For ordinal features statstical terminology of mean, median, std deviation doesnot make sense.
- StandardHours and EmployeeCount contain same value for all stastical parameter. It means they contain one unique value.
- Lets do some Statistical Analysis. Start with target Variable.

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

No      1233
Yes      237
Name: Attrition, dtype: int64

labels = 'Yes','No',
fig, ax = plt.subplots()
ax.pie(df['Attrition'].value_counts(),labels = labels,radius
=2,autopct = '%2.2f%%',explode=[0.1,0.2], shadow=True,)
plt.show()
```



Comment:

83.88% (1237 employees) Employees did not leave the organization while 16.12% (237 employees) did leave the organization making our dataset to be considered imbalanced since more people stay in the organization than they actually leave.

Before arrive at key questions which need to answer about HR Attrition, let try to gain some insight about individual features like distribution of different subcategories, different insight about Human Resource in company like education,job level, working domain.

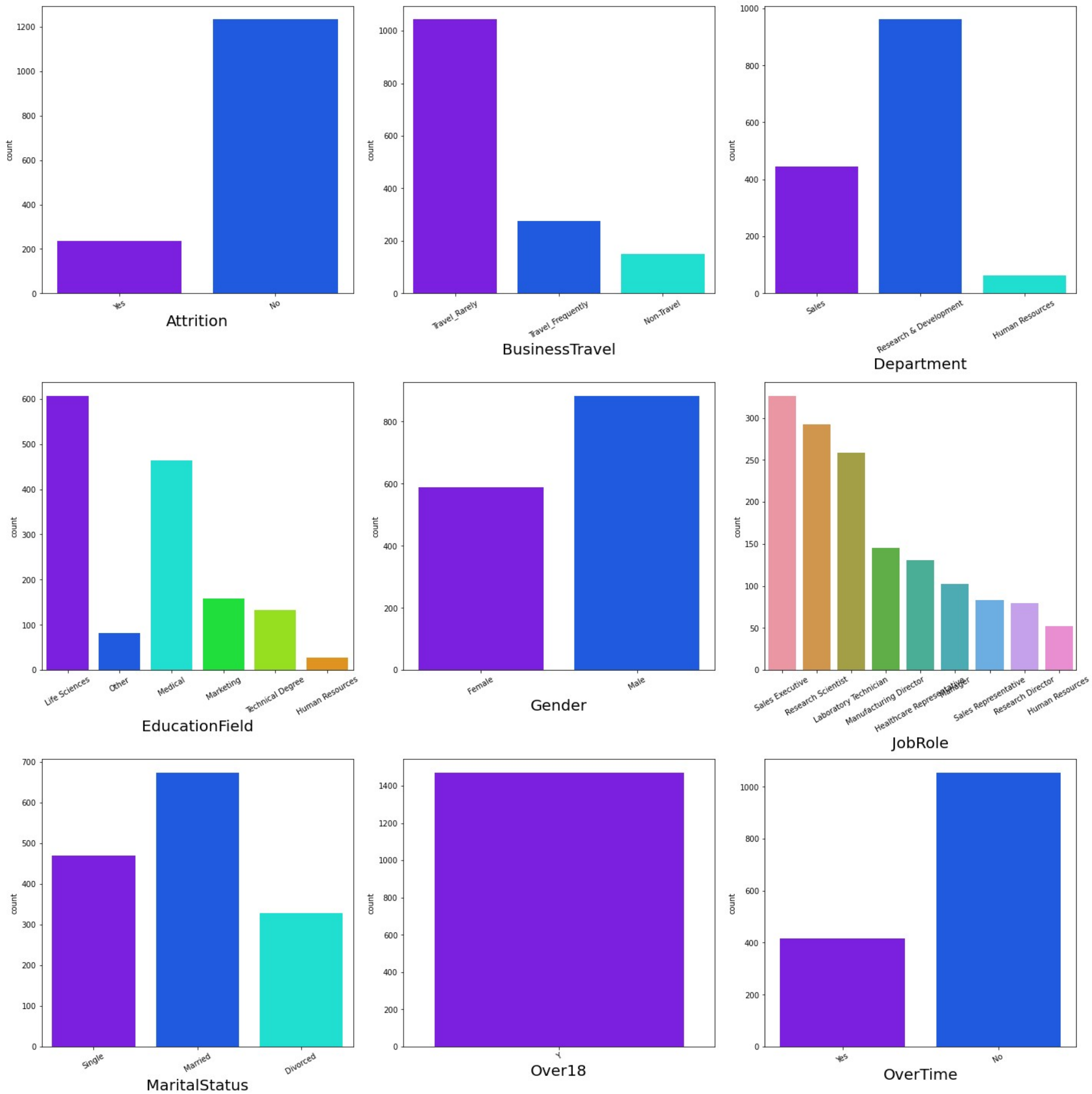
Start with Enlisting Value counts & Sub-categories of different categorial features available

```
Category=['Attrition', 'BusinessTravel', 'Department',
'EducationField',
          'Gender', 'JobRole', 'MaritalStatus', 'Over18', 'OverTime' ]
for i in Category:
    print(i)
    print(df[i].value_counts())
    print("="*100)
```

```
Attrition
No     1233
Yes     237
Name: Attrition, dtype: int64
======================================================================
==============================
BusinessTravel
Travel_Rarely        1043
Travel_Frequently     277
Non-Travel            150
Name: BusinessTravel, dtype: int64
======================================================================
==============================
Department
Research & Development    961
Sales                     446
Human Resources            63
Name: Department, dtype: int64
======================================================================
==============================
EducationField
Life Sciences       606
Medical             464
Marketing           159
Technical Degree    132
Other                82
Human Resources      27
Name: EducationField, dtype: int64
======================================================================
==============================
Gender
Male      882
Female    588
Name: Gender, dtype: int64
======================================================================
==============================
JobRole
Sales Executive              326
Research Scientist           292
Laboratory Technician        259
Manufacturing Director       145
Healthcare Representative    131
Manager                      102
Sales Representative          83
Research Director             80
Human Resources               52
Name: JobRole, dtype: int64
======================================================================
==============================
MaritalStatus
```

```
Married     673
Single      470
Divorced    327
Name: MaritalStatus, dtype: int64
======================================================================
==============================
Over18
Y    1470
Name: Over18, dtype: int64
======================================================================
==============================
OverTime
No     1054
Yes     416
Name: OverTime, dtype: int64
======================================================================
==============================
```

```
sns.set_palette('gist_rainbow_r')
plt.figure(figsize=(20,20), facecolor='white')
plotnumber =1
Category=['Attrition', 'BusinessTravel', 'Department',
'EducationField',
          'Gender', 'JobRole', 'MaritalStatus', 'Over18', 'OverTime' ]
for i in Category:
    if plotnumber <=9:
        ax = plt.subplot(3,3,plotnumber)
        sns.countplot(df[i])
        plt.xlabel(i,fontsize=20)
        plt.xticks(rotation=30)
    plotnumber+=1
plt.tight_layout()
plt.show()
```

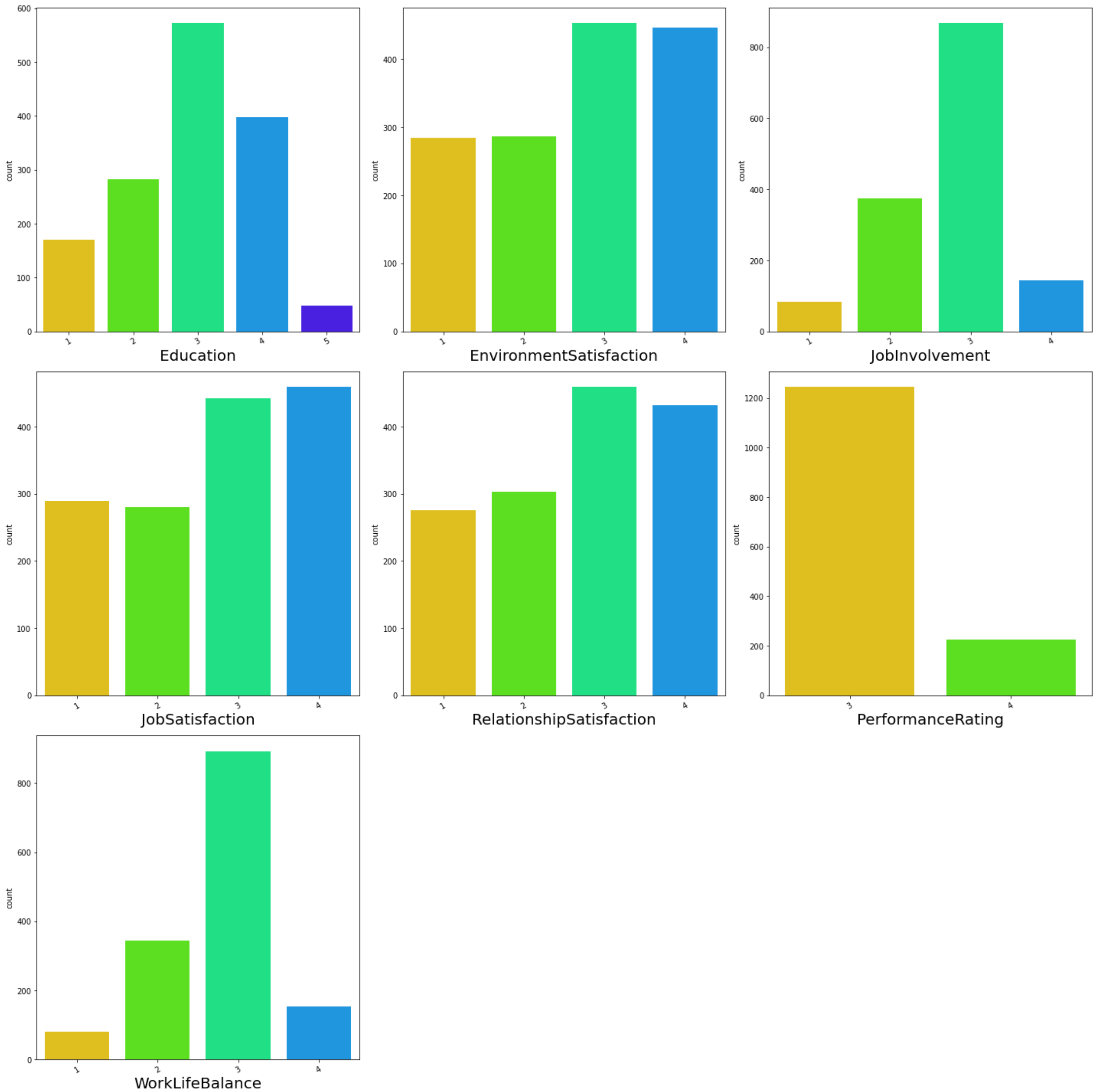


Enlisting Value counts & Sub-categories of different Ordinal features available

```
Ordinal=['Education','EnvironmentSatisfaction',
'JobInvolvement','JobSatisfaction',
          'RelationshipSatisfaction', 'PerformanceRating',
'WorkLifeBalance' ]
for i in Ordinal:
    print(i)
    print(df[i].value_counts())
    print("="*100)

Education
3    572
```

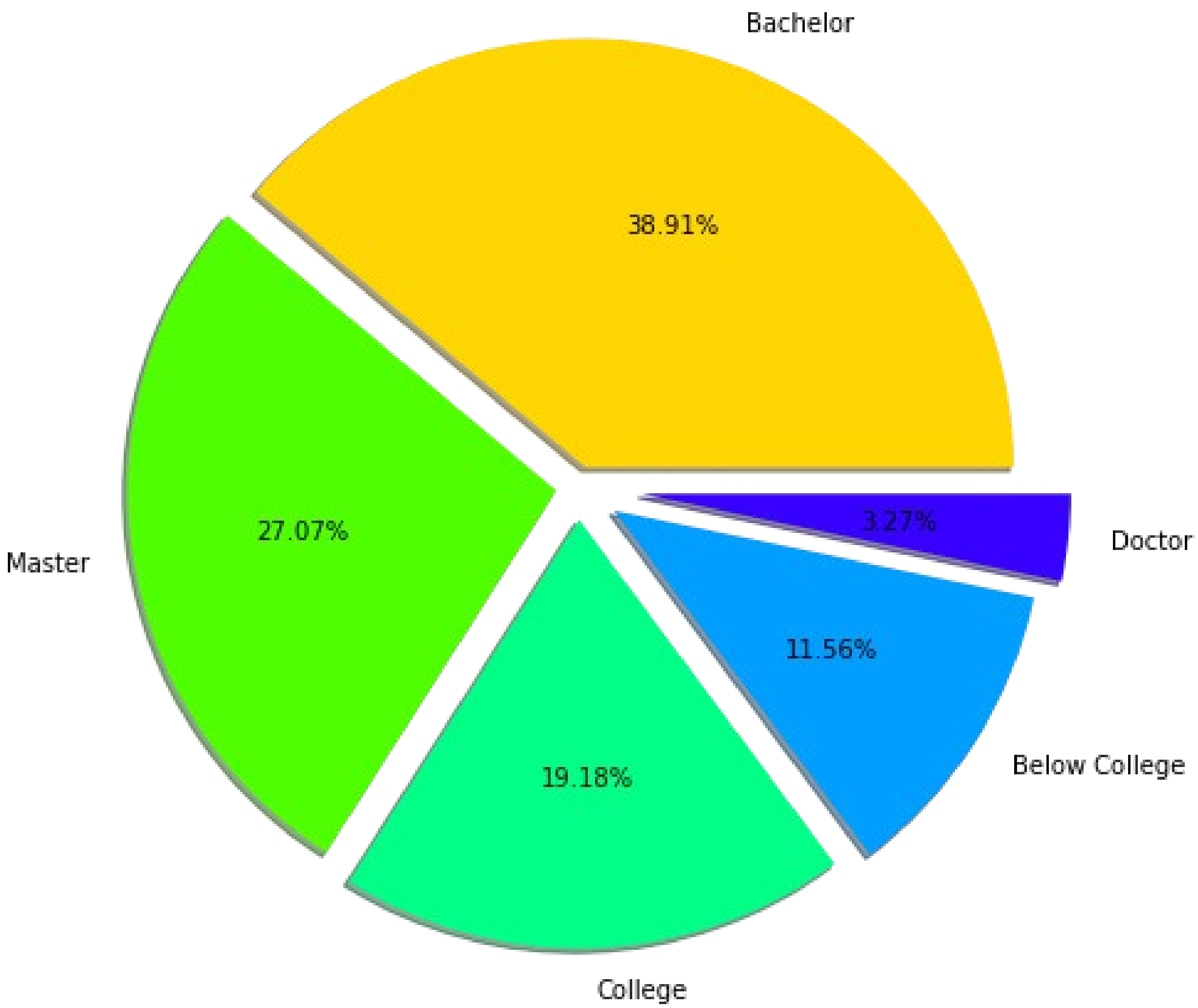
```
4    398
2    282
1    170
5     48
Name: Education, dtype: int64
======================================================================
==============================
EnvironmentSatisfaction
3    453
4    446
2    287
1    284
Name: EnvironmentSatisfaction, dtype: int64
======================================================================
==============================
JobInvolvement
3    868
2    375
4    144
1     83
Name: JobInvolvement, dtype: int64
======================================================================
==============================
JobSatisfaction
4    459
3    442
1    289
2    280
Name: JobSatisfaction, dtype: int64
======================================================================
==============================
RelationshipSatisfaction
3    459
4    432
2    303
1    276
Name: RelationshipSatisfaction, dtype: int64
======================================================================
==============================
PerformanceRating
3    1244
4     226
Name: PerformanceRating, dtype: int64
======================================================================
==============================
WorkLifeBalance
3    893
2    344
4    153
```

```
1      80
Name: WorkLifeBalance, dtype: int64
======================================================================
==============================

sns.set_palette('hsv')
plt.figure(figsize=(20,20), facecolor='white')
plotnumber =1
Ordinal=['Education','EnvironmentSatisfaction',
'JobInvolvement','JobSatisfaction',
          'RelationshipSatisfaction', 'PerformanceRating',
'WorkLifeBalance' ]
for i in Ordinal:
    if plotnumber <=9:
        ax = plt.subplot(3,3,plotnumber)
        sns.countplot(df[i])
        plt.xlabel(i,fontsize=20)
        plt.xticks(rotation=30)
    plotnumber+=1
plt.tight_layout()
plt.show()
```



Education level of Man power available

```
labels='Bachelor','Master','College','Below College','Doctor'
fig, ax = plt.subplots()
ax.pie(df['Education'].value_counts(),labels = labels,radius
=2,autopct = '%3.2f%%',explode=[0.1,0.1,0.15,0.2,0.3], shadow=True,)
plt.show()
```



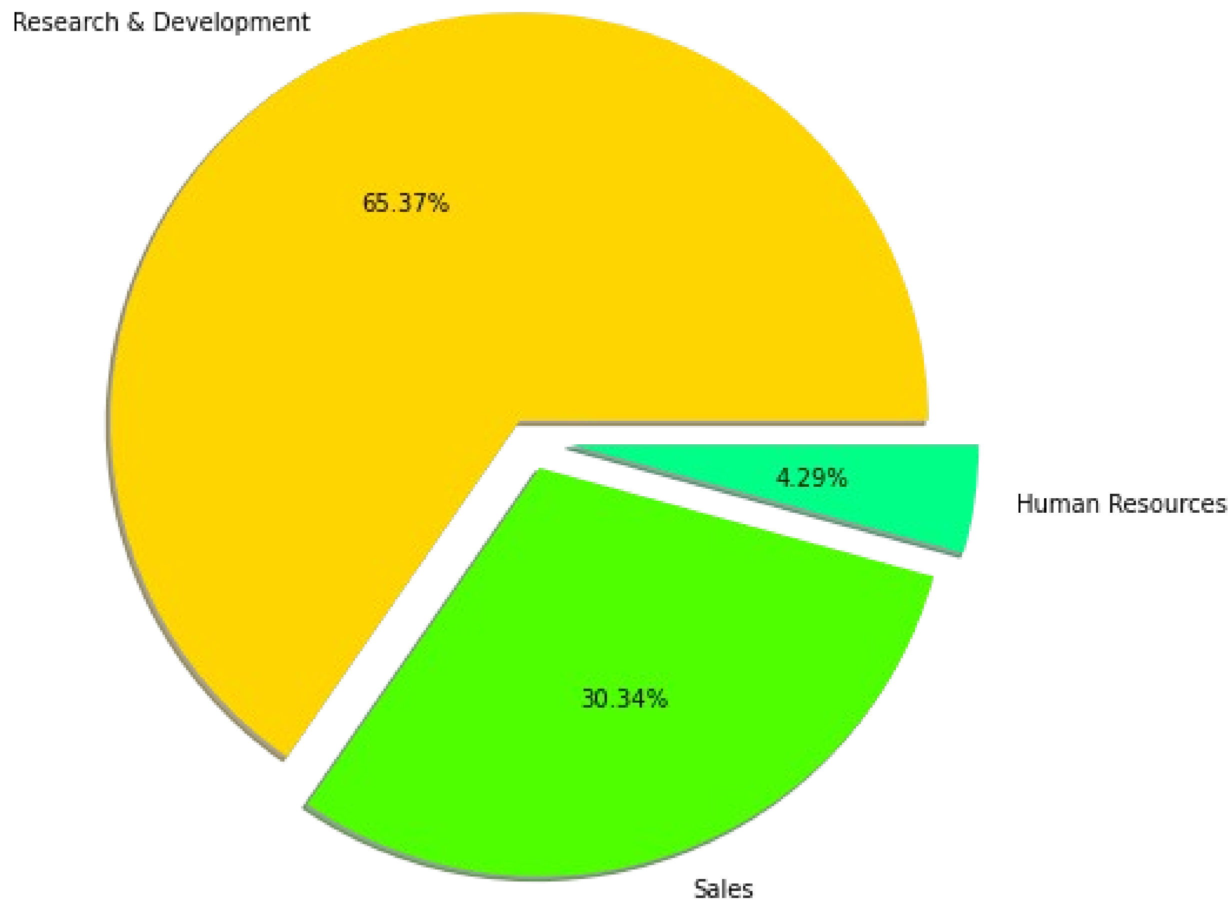
Comment:

- More than 60 % employees educated at Masters & Bachelor. It interesting to find out in which department need this human resources.
- 30 % of Employees are highly educated which involves master and doctor degree.
- 39 % of Employees are graduate.
- Almost 19% Employees are educated upto college & 12% are below college.

Lets try to gain insight on to which department this Human Resource belong and education need of each department through visualization.

```
df['Department'].value_counts()

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

```
labels ='Research & Development','Sales','Human Resources'
fig,ax= plt.subplots()
ax.pie(df['Department'].value_counts(),labels=labels,
radius=2,autopct= '%3.2f%%',explode=[0.1,0.15,0.2],shadow=True)
plt.show()
```




```
pd.crosstab([df.Education],[df.Department],
margins=True).style.background_gradient(cmap='summer_r')
```

```
<pandas.io.formats.style.Styler at 0x26f748eb370>
```

Comment :

- 65.37% of Employees belong to Research & Development Department. Out of Total 961 Employee no of employee educated at Bachelors,Masters, Doctor are 379,255 and 30 respectively.
- Only 63 Employee work in HR department.

```
pd.crosstab([df.Education],[df.Department,df.Attrition],
margins=True).style.background_gradient(cmap='summer_r')
```

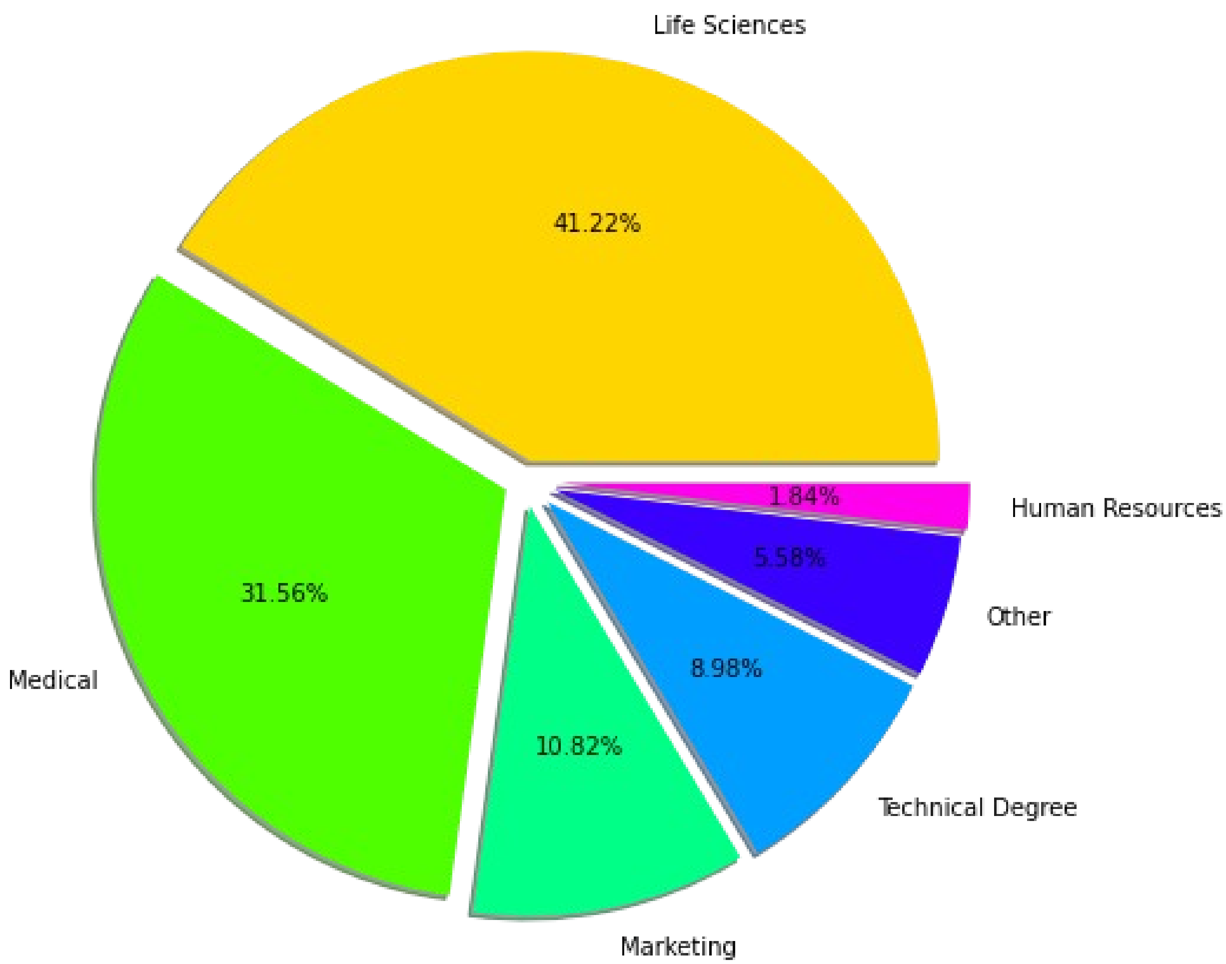
```
<pandas.io.formats.style.Styler at 0x26f745ef9d0>
```

Employee distribution as per education field and level of education

```
df['EducationField'].value_counts()
```

```
Life Sciences       606
Medical             464
Marketing           159
Technical Degree    132
Other                82
Human Resources      27
Name: EducationField, dtype: int64
```

```
labels ='Life Sciences','Medical','Marketing','Technical
Degree','Other','Human Resources'
fig,ax= plt.subplots()
ax.pie(df['EducationField'].value_counts(),labels=labels,
radius=2,autopct= '%3.2f%
%',explode=[0.1,0.1,0.125,0.15,0.15,0.175],shadow=True)
plt.show()
```



```
# Let check distribution of education Vs education Field
pd.crosstab([df.Education],[df.EducationField],
margins=True).style.background_gradient(cmap='summer_r')
```

```
<pandas.io.formats.style.Styler at 0x26f753bddf0>
```

```
# Let check distribution of department Vs education Field
pd.crosstab([df.Department],[df.EducationField],
margins=True).style.background_gradient(cmap='summer_r')
```

```
<pandas.io.formats.style.Styler at 0x26f753c8cd0>
```

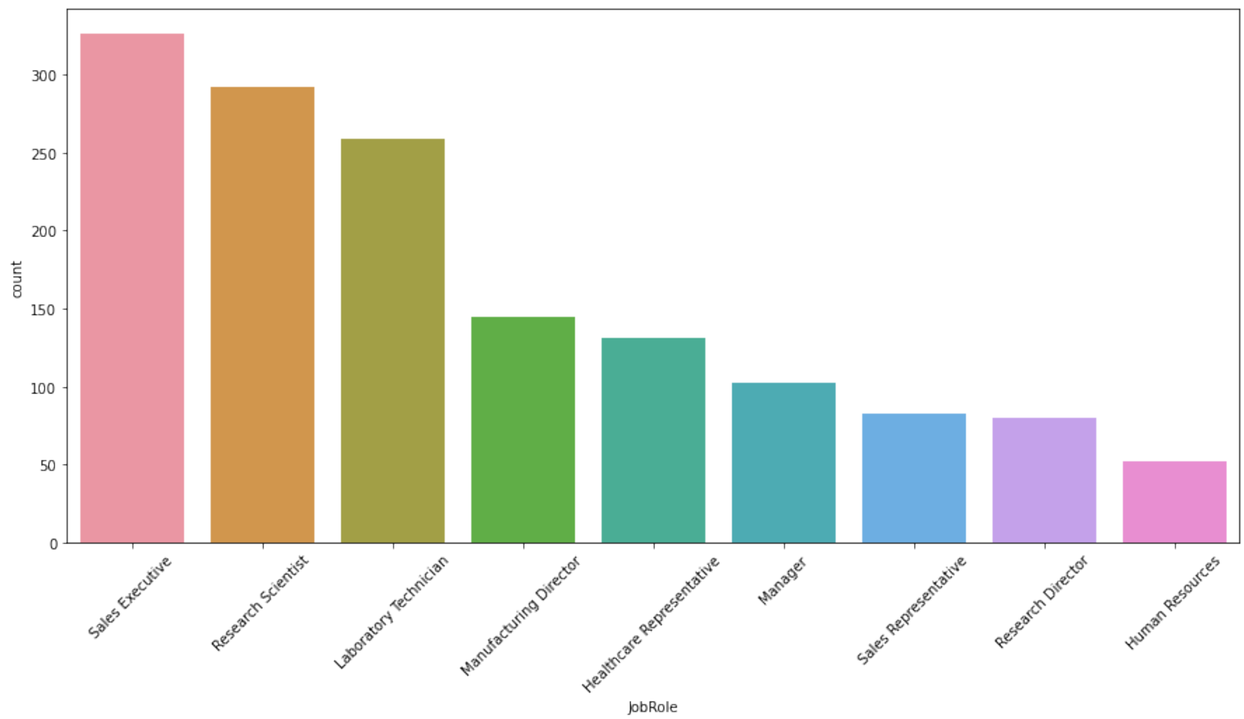
Comment:

- 41.22 % Employee comes from Life science background followed by Medical profession with 31.56%.
- There are only 27 people with HR background and We know that 63 people work in HR Department from previous result. This implies that atleast half employee working in HR department do not have HR background. This important as there is more probability of Employees Retention is when they are working in domain of interest or as per their education background. Dissatisfaction with want we doing can be seen as major reason of leaving job.
- Most of Employees with Techanical degree are Bachelors.
- Most of Employees having Masters and Doctors belong to Life Science and Medical domain.
- R&D department almost everyone comes from profession or technical background except support staff. Factor like Salary Hike, travelling, overtime and Job level are things need to taken in consideration while analysing Attrition of this category.
- There are 159 Employee with Marketing background and all work in Sales Department.
- 50% Employees in sales department have background of Life sciences & Medical. So it will interesting to see attrition rate in these employees.

We will Analysis Attrition over above insight in next section of Job role.

Lets work with Job Role

```
plt.figure(figsize=(15,7))
sns.countplot(df['JobRole'])
plt.xticks(rotation=45)
plt.show()
```



- Before going for Attrition by Job role,

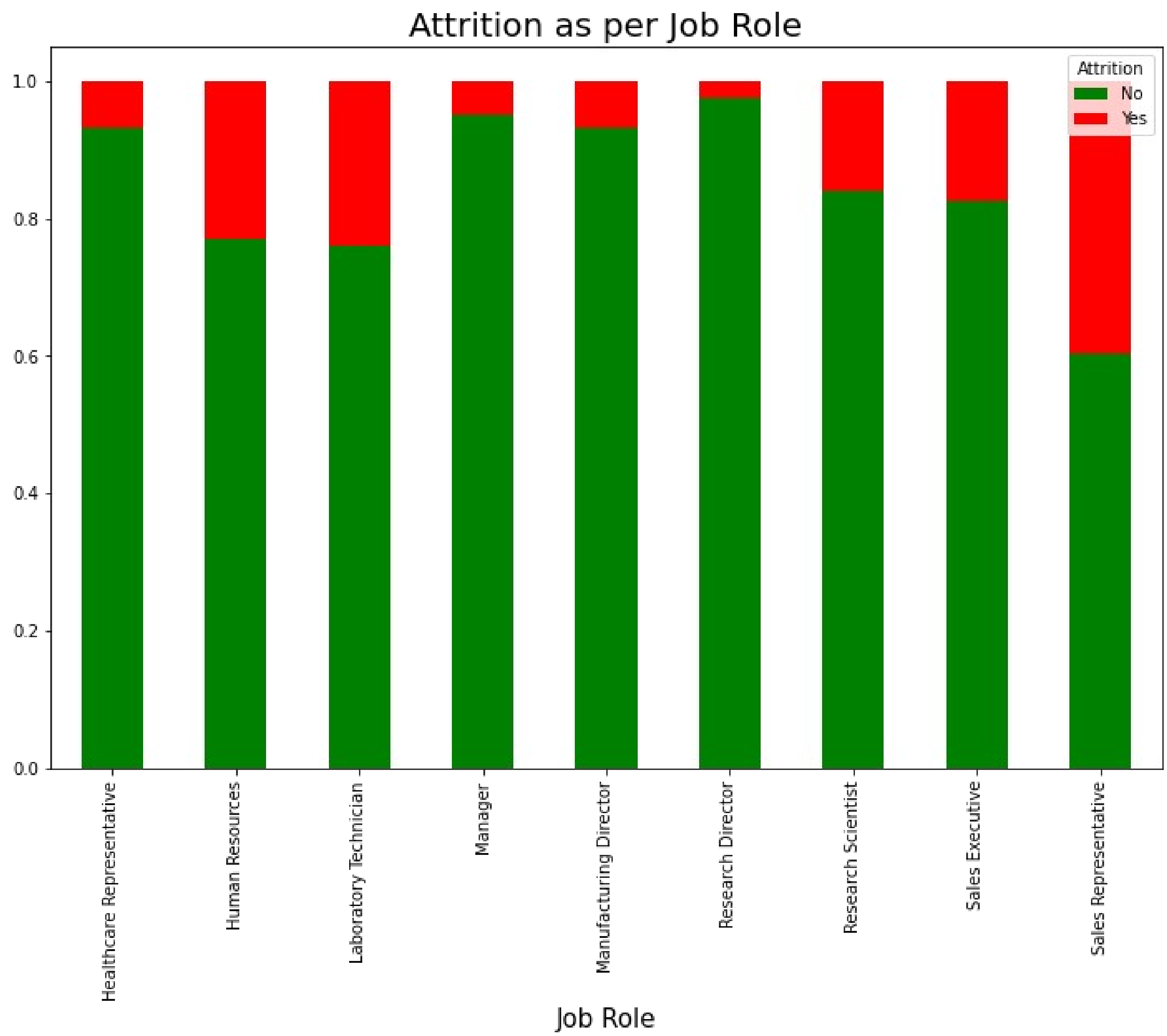
first build matrix of department vs job role which will give us idea about number of employees of different job role across department

```
pd.crosstab([df.JobRole],[df.Department],
margins=True).style.background_gradient(cmap='gist_rainbow_r')

<pandas.io.formats.style.Styler at 0x26f751ed970>
```

Comment:

- There are 3 job role in HR Department, maximum of which are sales Executive with 446 Total Employees.
- Human Resources department has 2 Job role i.e. HR & Manager.
- There 6 different Job role in R&D department with total 961 employees and until now we know that all of them belong to thier respective domain background.

```
plt.figure(figsize=(12,10))
data=pd.crosstab(df['JobRole'], df['Attrition'])
data.div(data.sum(1).astype(float), axis=0).plot(kind='bar',
stacked=True,
                    color=['green', 'red'],figsize=(12,8))
plt.title('Attrition as per Job Role', fontsize=20)
plt.xlabel('Job Role',fontsize=15)
plt.show()

<Figure size 864x720 with 0 Axes>
```



We all can definitely see Red Signal for different Managers & HR of Respective Job Role in above barplot !!!

Bar plot showing % attrition across each job role, let check absolute number matrix of attrition, again this time using crosstab.

```
pd.crosstab([df.JobRole,df.Department],[df.Attrition],
margins=True).style.background_gradient(cmap='gist_rainbow_r')
```

```
<pandas.io.formats.style.Styler at 0x26f751cf550>
```

Comment:

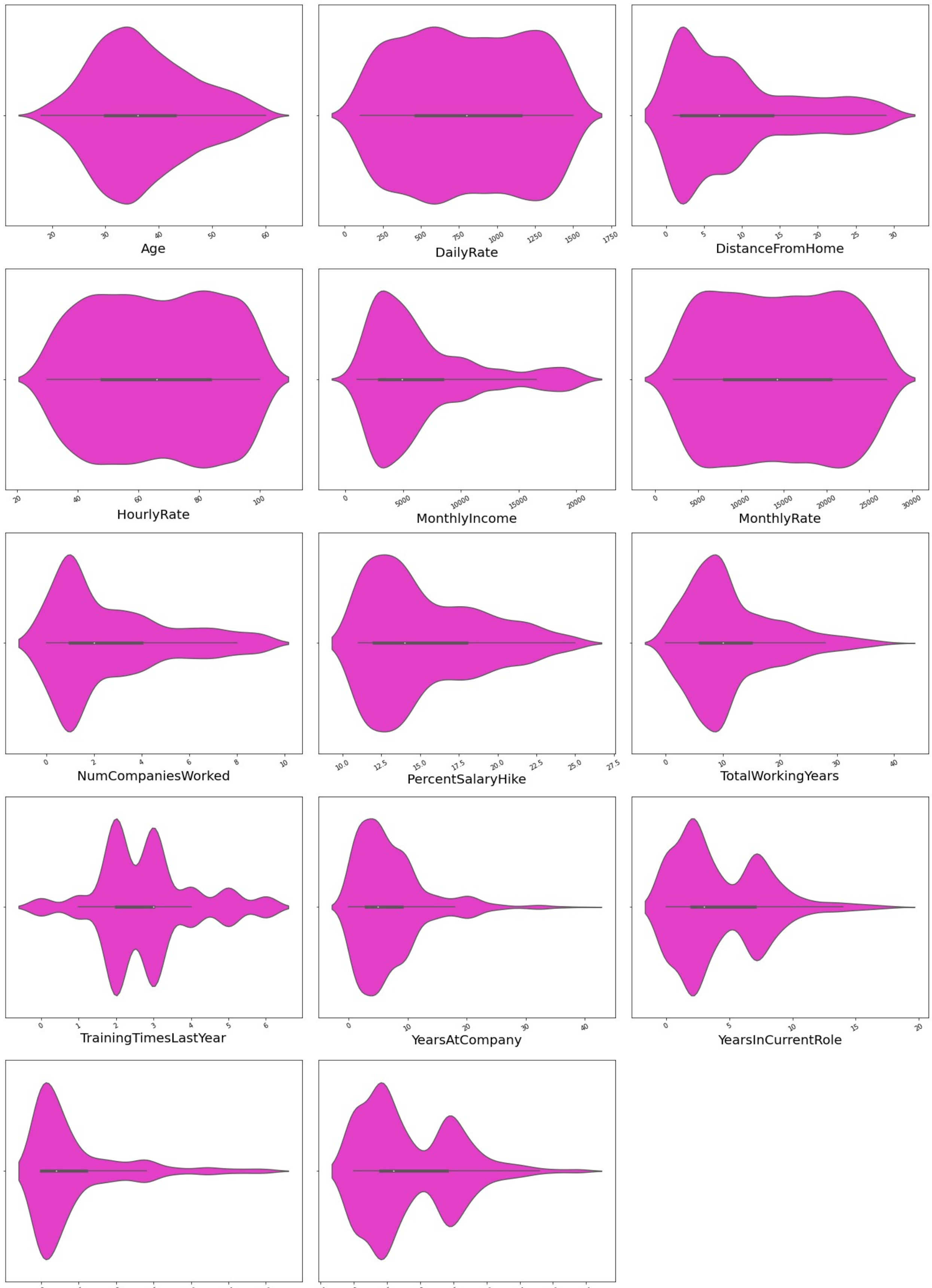
- Percentage of attrition is high in Sales Representative,Laboratory Technician,Human Resources. This all job role comes at bottom in corparate hierachy also Salary is comparatively less compare to other job role.

- Monthly Income, Job stastifation, travelling are feature need to dive into for further insights in these job role.
- At the Top chart 62 Laboratory Technician has resign from job, followed by 57 sales executive and 47 Research Scientist.
- 16 % arttrition rate for Research Scientist, which involve huge investment from company. Company not only loses employee but its knowledge base, expertise & Intellatual property rights in some cases.

```
# Grouping Numeric Features
Numeric=['Age', 'DailyRate', 'DistanceFromHome',
 'EnvironmentSatisfaction', 'HourlyRate', 'JobInvolvement',
'JobLevel', 'JobSatisfaction',
 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
'PercentSalaryHike', 'PerformanceRating',
 'RelationshipSatisfaction','StockOptionLevel', 'TotalWorkingYears',
'TrainingTimesLastYear',
 'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
'YearsSinceLastPromotion', 'YearsWithCurrManager']
```

Violinplot of Numeric Variables

```
# Grouping Numeric Features
Numeric_int=['Age', 'DailyRate', 'DistanceFromHome',
'HourlyRate','MonthlyIncome', 'MonthlyRate',
             'NumCompaniesWorked', 'PercentSalaryHike',
'TotalWorkingYears','TrainingTimesLastYear',
             'YearsAtCompany', 'YearsInCurrentRole',
'YearsSinceLastPromotion', 'YearsWithCurrManager']

sns.set_palette('spring')
plt.figure(figsize=(20,50), facecolor='white')
plotnumber =1

for i in Numeric_int:
    if plotnumber <=25:
        ax = plt.subplot(9,3,plotnumber)
        sns.violinplot(df[i])
        plt.xlabel(i,fontsize=20)
        plt.xticks(rotation=30)
    plotnumber+=1
plt.tight_layout()
plt.show()
```


Age
DailyRate
DistanceFromHome
HourlyRate
MonthlyIncome
MonthlyRate
NumCompaniesWorked
PercentSalaryHike
TotalWorkingYears
TrainingTimesLastYear
YearsAtCompany
YearsInCurrentRole

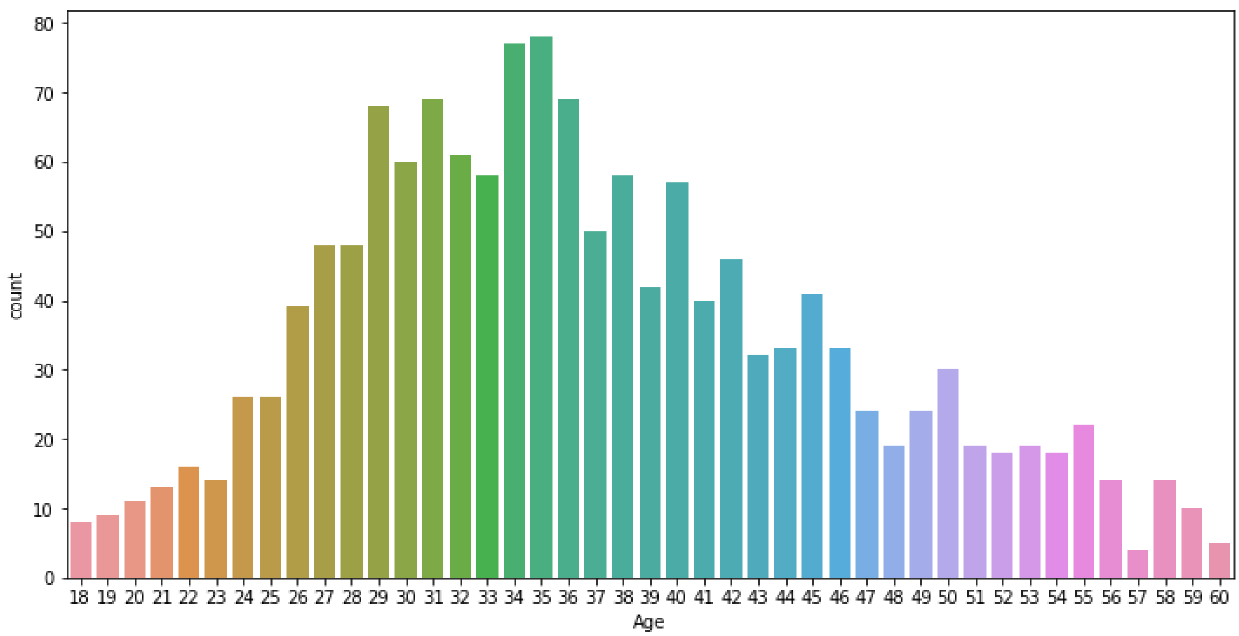
Comment:

- For Majority of people have spend 3 to 10 years at company.
- Most of people staying company upto 2 years after promotion.
- Majority of people are are train 2-3 times in last year.If employees leaves job then it loss investment for company.
- Majority of people stay in same role for maximum 4 yrs.
- Majority of Employees have salary hike of 10 to 15%.

Age Vs Attrition

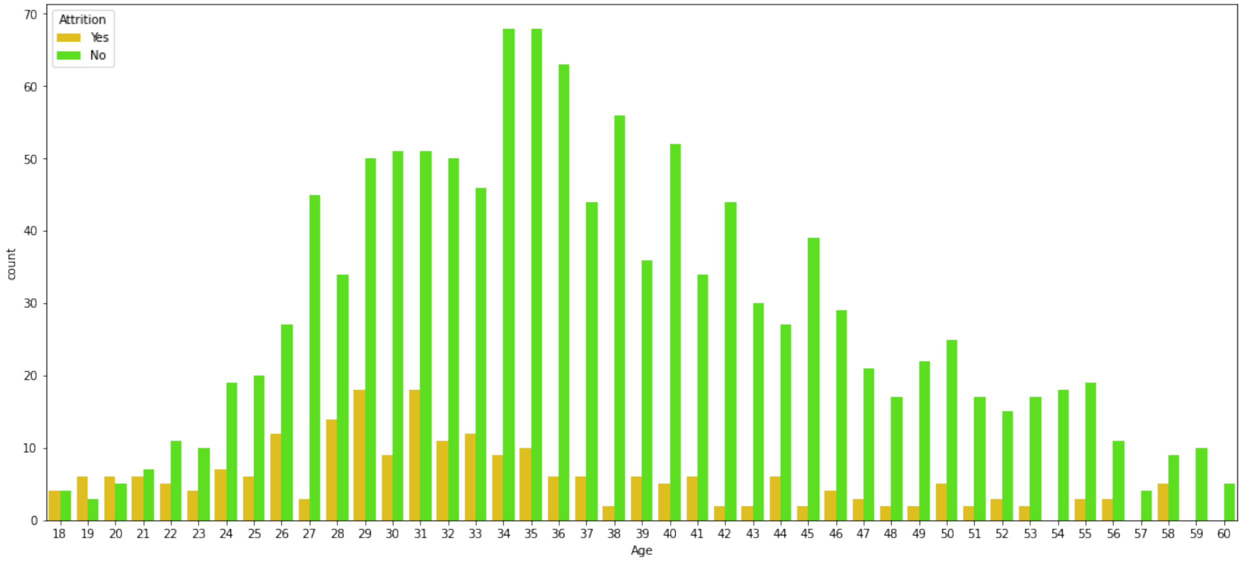
```
plt.subplots(figsize=(12,6))
sns.countplot(df['Age'])
```

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




```
sns.set_palette('hsv')
plt.subplots(figsize=(18,8))
sns.countplot(x='Age', hue='Attrition', data=df)
```

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

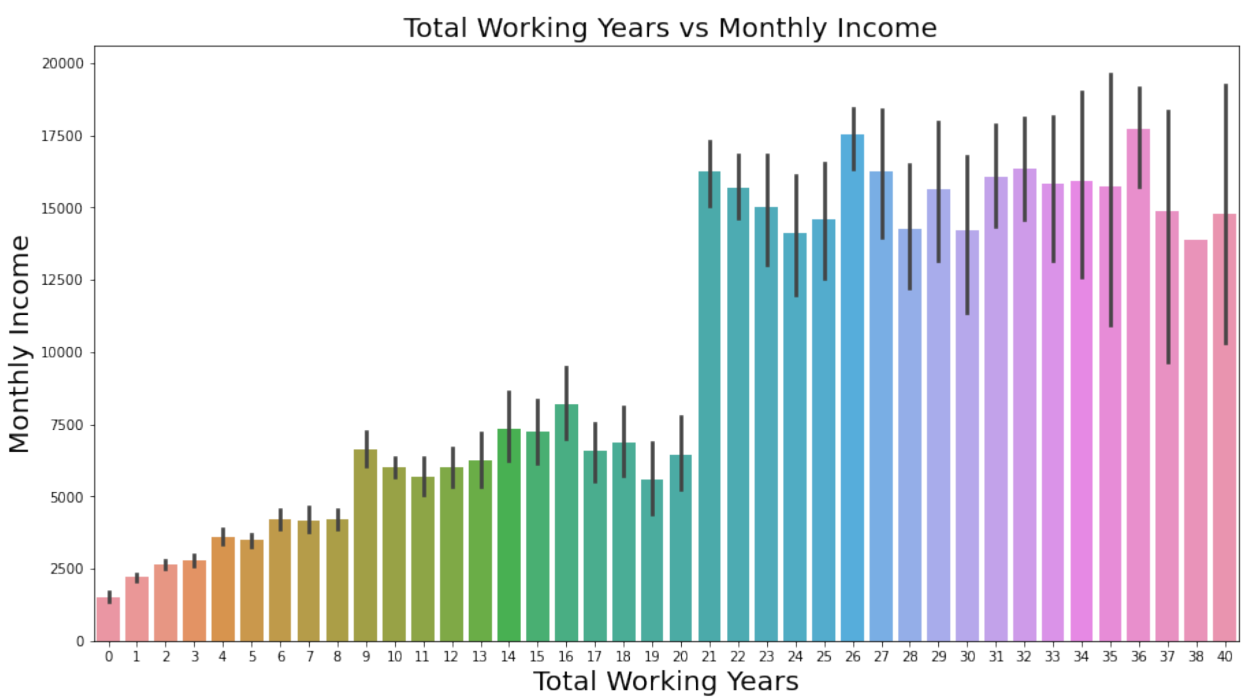


Comment:

1. The Attrition rate is minimum between the Age years of 34 and 35.

2. The Attrition rate is maximum between the Age years of 29 and 31.

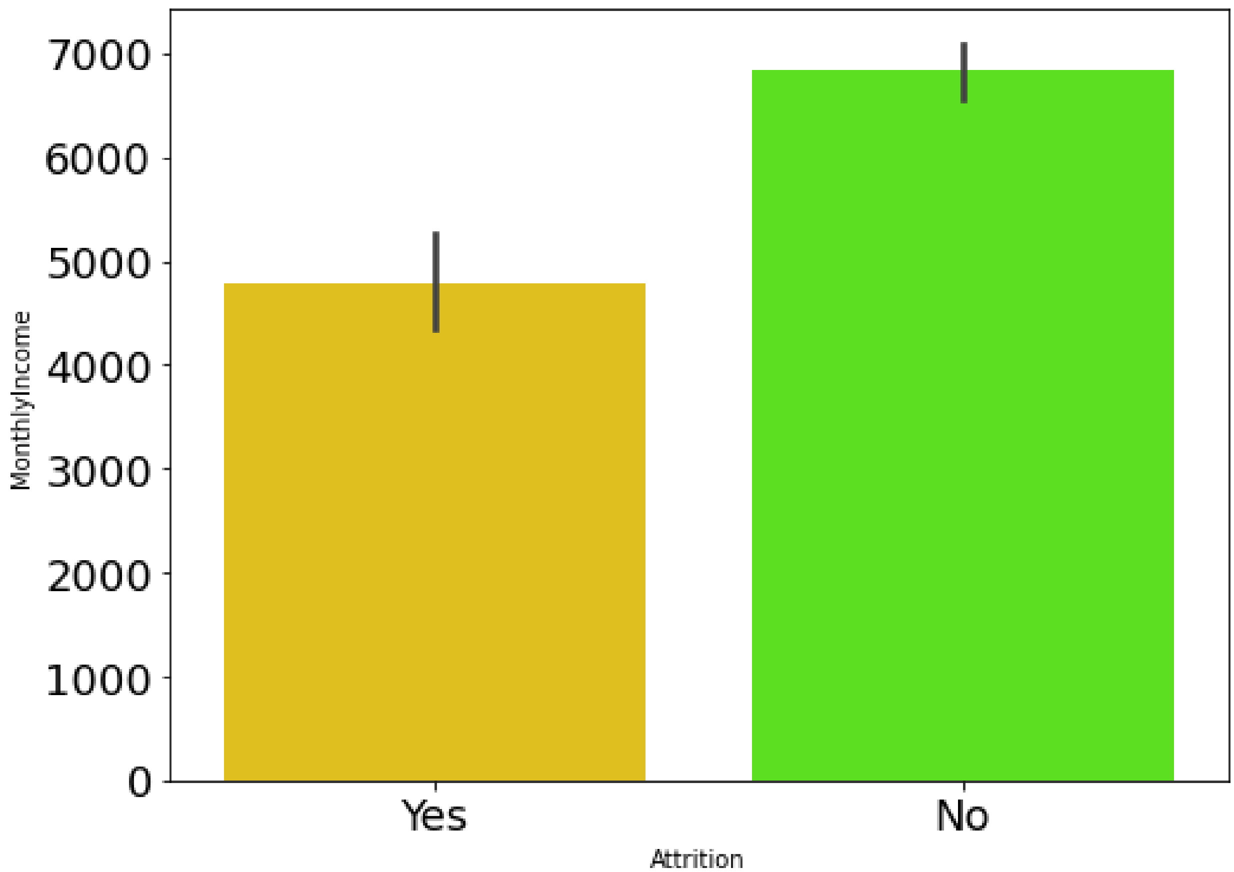
```
plt.figure(figsize=(15,8))
sns.barplot(df['TotalWorkingYears'],df['MonthlyIncome'])
plt.xlabel('Total Working Years',fontsize=20)
plt.ylabel('Monthly Income',fontsize=20)
plt.title(" Total Working Years vs Monthly Income", fontsize=20)
plt.show()
```



Comment:

Monthly Income is highest for the employees with 21 or more number of Total Working Years.

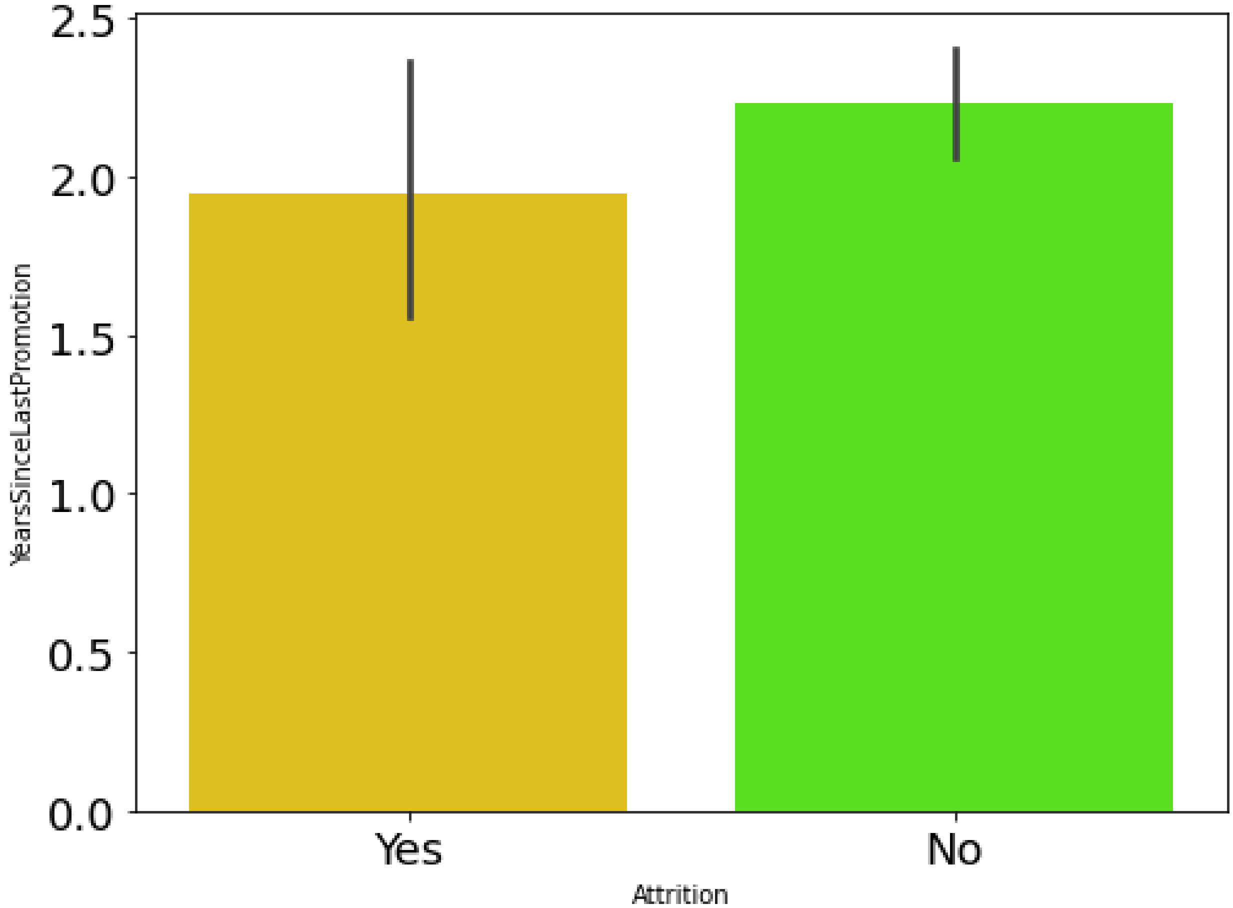
```
plt.figure(figsize=(8,6))
sns.barplot(x='Attrition',y='MonthlyIncome',data=df)
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
plt.show()
```



Comment:

The Attrition rate in the employees is less when the monthly income reaches to 6900.

```
plt.figure(figsize=(8,6))
sns.barplot(x='Attrition',y='YearsSinceLastPromotion',data=df)
plt.xticks(fontsize=18)
plt.yticks(fontsize=18)
plt.show()
```



Comment:

The rate of Attrition is high when the employee did not got promoted since 1.8 years.

```
df=pd.read_csv('WA_Fn-UseC_-HR-Employee-Attrition.csv')
```

Encoding categorical data

```
# Using Label Encoder on target variable
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df["Attrition"] = le.fit_transform(df["Attrition"])
df.head()

   Age  Attrition     BusinessTravel  DailyRate
Department  \
0   41          1      Travel_Rarely       1102
Sales
1   49          0  Travel_Frequently        279  Research &
Development
```

```
2   37          1      Travel_Rarely       1373  Research &
Development
3   33          0  Travel_Frequently       1392  Research &
Development
4   27          0      Travel_Rarely        591  Research &
Development

   DistanceFromHome  Education EducationField  EmployeeCount
EmployeeNumber  \
0                 1          2  Life Sciences              1
1
1                 8          1  Life Sciences              1
2
2                 2          2          Other              1
4
3                 3          4  Life Sciences              1
5
4                 2          1        Medical              1
7

   EnvironmentSatisfaction  Gender  HourlyRate  JobInvolvement
JobLevel  \
0                        2  Female          94               3
2
1                        3    Male          61               2
2
2                        4    Male          92               2
1
3                        4  Female          56               3
1
4                        1    Male          40               3
1

                 JobRole  JobSatisfaction MaritalStatus  MonthlyIncome
\
0        Sales Executive                4        Single           5993

1     Research Scientist                2       Married           5130

2  Laboratory Technician                3        Single           2090

3     Research Scientist                3       Married           2909

4  Laboratory Technician                2       Married           3468


   MonthlyRate  NumCompaniesWorked Over18 OverTime  PercentSalaryHike
\
0        19479                   8      Y      Yes                 11
```

```
1        24907                   1      Y       No                   23
2         2396                   6      Y      Yes                   15
3        23159                   1      Y      Yes                   11
4        16632                   9      Y       No                   12

   PerformanceRating  RelationshipSatisfaction  StandardHours  \
0                  3                         1             80
1                  4                         4             80
2                  3                         2             80
3                  3                         3             80
4                  3                         4             80

   StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
0                 0                  8                      0
1                 1                 10                      3
2                 0                  7                      3
3                 0                  8                      3
4                 1                  6                      3

   WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
0                1               6                   4
1                3              10                   7
2                3               0                   0
3                3               8                   7
4                3               2                   2

   YearsSinceLastPromotion  YearsWithCurrManager
0                        0                     5
1                        1                     7
2                        0                     0
3                        3                     0
4                        2                     2
```

```
# Droping unnecessary columns
df.drop(["EmployeeCount", "EmployeeNumber", "Over18",
"StandardHours"], axis=1, inplace=True)

df.shape
```

```
(1470, 31)
```

```
# Ordinal Encoding for ordinal variables
from sklearn.preprocessing import OrdinalEncoder
oe = OrdinalEncoder()
def ordinal_encode(df, column):
    df[column] = oe.fit_transform(df[column])
    return df
```

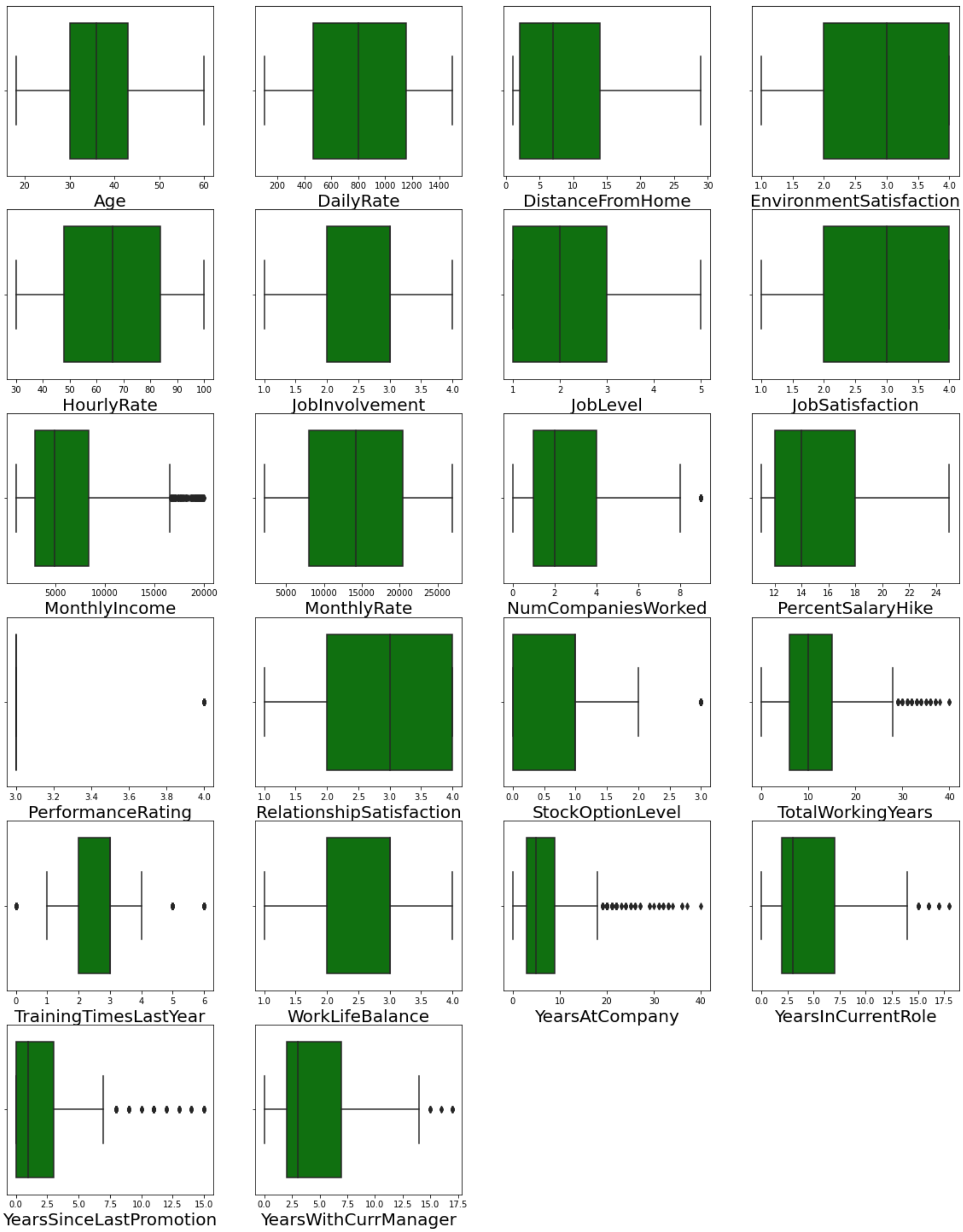
```
oe_col = ['BusinessTravel', 'Department', 'EducationField', 'Gender',
'JobRole', 'MaritalStatus', 'OverTime']
df=ordinal_encode(df, oe_col)
df.head()
```

```
   Age  Attrition  BusinessTravel  DailyRate  Department  
DistanceFromHome  \
0   41          1             2.0       1102         2.0  
1  
1   49          0             1.0        279         1.0  
8  
2   37          1             2.0       1373         1.0  
2  
3   33          0             1.0       1392         1.0  
3  
4   27          0             2.0        591         1.0  
2  

   Education  EducationField  EnvironmentSatisfaction  Gender  
HourlyRate  \
0          2             1.0                        2     0.0  
94  
1          1             1.0                        3     1.0  
61  
2          2             4.0                        4     1.0  
92  
3          4             1.0                        4     0.0  
56  
4          1             3.0                        1     1.0  
40  

   JobInvolvement  JobLevel  JobRole  JobSatisfaction  
MaritalStatus  \
0               3         2      7.0                4            2.0  

1               2         2      6.0                2            1.0  

2               2         1      2.0                3            2.0  

3               3         1      6.0                3            1.0  

4               3         1      2.0                2            1.0  


   MonthlyIncome  MonthlyRate  NumCompaniesWorked  OverTime  \
0           5993        19479                   8       1.0  
1           5130        24907                   1       0.0  
2           2090         2396                   6       1.0  
3           2909        23159                   1       1.0  
```

```
4          3468       16632                    9      0.0

   PercentSalaryHike  PerformanceRating  RelationshipSatisfaction  \
0                 11                  3                         1
1                 23                  4                         4
2                 15                  3                         2
3                 11                  3                         3
4                 12                  3                         4

   StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
0                 0                  8                      0
1                 1                 10                      3
2                 0                  7                      3
3                 0                  8                      3
4                 1                  6                      3

   WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
0                1               6                   4
1                3              10                   7
2                3               0                   0
3                3               8                   7
4                3               2                   2

   YearsSinceLastPromotion  YearsWithCurrManager
0                        0                     5
1                        1                     7
2                        0                     0
3                        3                     0
4                        2                     2
```

Outliers Detection and Removal

```
plt.figure(figsize=(20,30),facecolor='white')
plotnumber=1

for column in Numeric:
    if plotnumber<=28:
        ax=plt.subplot(7,4,plotnumber)
        sns.boxplot(df[column],color='g')
        plt.xlabel(column,fontsize=20)
    plotnumber+=1
plt.show()
```


Age
DailyRate
DistanceFromHome
EnvironmentSatisfaction
HourlyRate
JobInvolvement
JobLevel
JobSatisfaction
MonthlyIncome
MonthlyRate
NumCompaniesWorked
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager

Features containing outliers

"MonthlyIncome", "NumCompaniesWorked", "PerformanceRating", "StockOptionLevel", "TotalWorkingYears", "TrainingTimesLastYear", "YearsAtCompany", "YearsInCurrentRole", "YearsSinceLastPromotion", "YearsWithCurrManager", "Attrition".

```
from scipy.stats import zscore
z = np.abs(zscore(df))
threshold = 3
df1 = df[(z<3).all(axis = 1)]

print ("Shape of the dataframe before removing outliers: ", df.shape)
print ("Shape of the dataframe after removing outliers: ", df1.shape)
print ("Percentage of data loss post outlier removal: ", (df.shape[0]-
df1.shape[0])/df.shape[0]*100)

df=df1.copy() # reassigning the changed dataframe name to our original
dataframe name
```

```
Shape of the dataframe before removing outliers:  (1470, 31)
Shape of the dataframe after removing outliers:  (1387, 31)
Percentage of data loss post outlier removal:  5.646258503401361
```

Data Loss

```
print("\033[1m"+'Percentage Data Loss :'+"\033[0m",((1470-
1387)/1470)*100,'%')
```

```
Percentage Data Loss : 5.646258503401361 %
```

Feature selection and Engineering

1. Skewness of features

```
df.skew()
```

```
Age                         0.472280
Attrition                   1.805983
BusinessTravel             -1.426774
DailyRate                  -0.017078
Department                  0.183919
DistanceFromHome            0.954752
Education                  -0.289024
EducationField              0.544868
EnvironmentSatisfaction    -0.325285
Gender                     -0.417296
HourlyRate                 -0.030481
JobInvolvement             -0.501401
JobLevel                    1.126075
```

```
JobRole                     -0.386843
JobSatisfaction             -0.345612
MaritalStatus               -0.160952
MonthlyIncome                1.544770
MonthlyRate                  0.030596
NumCompaniesWorked           1.037715
OverTime                     0.954751
PercentSalaryHike            0.800592
PerformanceRating            1.931566
RelationshipSatisfaction    -0.295686
StockOptionLevel             0.962332
TotalWorkingYears            1.034487
TrainingTimesLastYear        0.577614
WorkLifeBalance             -0.557100
YearsAtCompany               1.248623
YearsInCurrentRole           0.726675
YearsSinceLastPromotion      1.756335
YearsWithCurrManager         0.694506
dtype: float64
```

```
# Splitting data in target and dependent feature
X = df.drop(['Attrition'], axis =1)
Y = df['Attrition']
```

Transforming skew data using power transform

```
from sklearn.preprocessing import power_transform
df = power_transform(X)
df = pd.DataFrame(df, columns=X.columns)
df.skew()
```

```
Age                         -0.004079
BusinessTravel              -0.960583
DailyRate                   -0.199742
Department                   0.015095
DistanceFromHome            -0.008149
Education                   -0.103747
EducationField              -0.008642
EnvironmentSatisfaction     -0.205472
Gender                      -0.417296
HourlyRate                  -0.105678
JobInvolvement              -0.018801
JobLevel                     0.110769
JobRole                     -0.337641
JobSatisfaction             -0.217730
MaritalStatus               -0.158253
MonthlyIncome                0.027700
MonthlyRate                 -0.176560
NumCompaniesWorked           0.016175
OverTime                     0.954751
```

```
PercentSalaryHike            0.112128
PerformanceRating            0.000000
RelationshipSatisfaction    -0.191406
StockOptionLevel             0.089929
TotalWorkingYears           -0.009666
TrainingTimesLastYear        0.057949
WorkLifeBalance             -0.011133
YearsAtCompany              -0.025230
YearsInCurrentRole          -0.069631
YearsSinceLastPromotion      0.212301
YearsWithCurrManager        -0.070570
dtype: float64
```

Comment :

- For Numeric features skewness is transform within permissible limit.
- For ordinal features & categorical features skew parameter irrevalent.

2. Corrleation

```
df.corr()

                               Age  BusinessTravel  DailyRate
Department  \
Age                       1.000000        0.019607   0.019864    -
0.036344
BusinessTravel            0.019607        1.000000  -0.001984    -
0.003560
DailyRate                 0.019864       -0.001984   1.000000    -
0.003546
Department               -0.036344       -0.003560  -0.003546
1.000000
DistanceFromHome         -0.025855       -0.007041  -0.006034
0.037834
Education                 0.215520       -0.006468  -0.017504
0.012780
EducationField           -0.037564        0.034658   0.040993
0.082525
EnvironmentSatisfaction   0.013967        0.004183   0.034324    -
0.013867
Gender                   -0.037163       -0.011439  -0.003271    -
0.030950
HourlyRate                0.026203        0.026364   0.015156    -
0.000623
JobInvolvement            0.032323        0.018230   0.041841    -
0.025121
JobLevel                  0.442350        0.003401   0.015931
0.200829
JobRole                  -0.116758       -0.002615  -0.013156
0.681597
```

```
JobSatisfaction             0.010038       -0.033026   0.044460
0.030615
MaritalStatus              -0.117182        0.010108  -0.076058
0.052696
MonthlyIncome               0.452513        0.030793   0.029944
0.152234
MonthlyRate                 0.020538       -0.008138  -0.032890
0.023941
NumCompaniesWorked          0.340022        0.034013   0.034923   -
0.033131
OverTime                    0.028332        0.010934   0.020045
0.015121
PercentSalaryHike           0.010488       -0.019175   0.029183   -
0.013541
PerformanceRating          -0.002365       -0.021061   0.000687   -
0.038429
RelationshipSatisfaction    0.037296       -0.036165   0.005771   -
0.037572
StockOptionLevel            0.089449       -0.006092   0.049415   -
0.000630
TotalWorkingYears           0.652405        0.027298   0.042750   -
0.006833
TrainingTimesLastYear      -0.014951        0.006192   0.005118
0.039938
WorkLifeBalance            -0.016180       -0.017977  -0.046550
0.017807
YearsAtCompany              0.207538       -0.024021   0.005391
0.025457
YearsInCurrentRole          0.145404       -0.035610   0.022143
0.057817
YearsSinceLastPromotion     0.114162       -0.033148  -0.035448
0.017699
YearsWithCurrManager        0.142446       -0.032665   0.005908
0.024241

                          DistanceFromHome  Education  EducationField
\
Age                              -0.025855   0.215520       -0.037564

BusinessTravel                   -0.007041  -0.006468        0.034658

DailyRate                        -0.006034  -0.017504        0.040993

Department                        0.037834   0.012780        0.082525

DistanceFromHome                  1.000000   0.002714        0.021074

Education                         0.002714   1.000000       -0.038405

EducationField                    0.021074  -0.038405        1.000000
```

```
EnvironmentSatisfaction          -0.013409  -0.026095         0.042609
Gender                            0.010557  -0.017807         0.005059
HourlyRate                        0.015607   0.011105        -0.004372
JobInvolvement                    0.038096   0.042166        -0.007969
JobLevel                          0.024038   0.103834        -0.026676
JobRole                           0.010044   0.016548         0.050693
JobSatisfaction                  -0.020165  -0.005640        -0.050693
MaritalStatus                    -0.027285  -0.012237         0.013433
MonthlyIncome                     0.000545   0.112084        -0.020033
MonthlyRate                       0.047736  -0.018874        -0.027785
NumCompaniesWorked               -0.010318   0.136101        -0.010403
OverTime                          0.036524  -0.015248         0.010335
PercentSalaryHike                 0.034946  -0.002095         0.000812
PerformanceRating                 0.013212  -0.023157        -0.001393
RelationshipSatisfaction          0.009379  -0.004863        -0.018254
StockOptionLevel                  0.027082   0.025621        -0.012936
TotalWorkingYears                -0.012129   0.150720        -0.001827
TrainingTimesLastYear            -0.015334  -0.023039         0.054321
WorkLifeBalance                  -0.030011   0.010164         0.034788
YearsAtCompany                    0.006570   0.037921         0.004483
YearsInCurrentRole                0.013091   0.051072         0.004372
YearsSinceLastPromotion          -0.003873   0.016076         0.023062
YearsWithCurrManager             -0.002310   0.026651         0.028189

                          EnvironmentSatisfaction    Gender
HourlyRate  \
Age                                      0.013967 -0.037163
0.026203
```

```
BusinessTravel                                0.004183 -0.011439
0.026364
DailyRate                                     0.034324 -0.003271
0.015156
Department                                   -0.013867 -0.030950    -
0.000623
DistanceFromHome                             -0.013409  0.010557
0.015607
Education                                    -0.026095 -0.017807
0.011105
EducationField                                0.042609  0.005059    -
0.004372
EnvironmentSatisfaction                       1.000000 -0.014940    -
0.042512
Gender                                       -0.014940  1.000000
0.005618
HourlyRate                                   -0.042512  0.005618
1.000000
JobInvolvement                               -0.020953  0.014878
0.051979
JobLevel                                      0.010615 -0.058378    -
0.039909
JobRole                                      -0.022464 -0.036436    -
0.023758
JobSatisfaction                              -0.009553  0.038130    -
0.067797
MaritalStatus                                -0.012356 -0.056779    -
0.008966
MonthlyIncome                                -0.011976 -0.052340    -
0.023613
MonthlyRate                                   0.036843 -0.047240    -
0.011438
NumCompaniesWorked                            0.011203 -0.033345
0.019917
OverTime                                      0.058274 -0.051558    -
0.003232
PercentSalaryHike                            -0.027743  0.010984    -
0.015826
PerformanceRating                            -0.024853 -0.010757    -
0.006571
RelationshipSatisfaction                      0.016892  0.041439
0.005207
StockOptionLevel                              0.024345  0.024390
0.041329
TotalWorkingYears                            -0.013356 -0.049776    -
0.012902
TrainingTimesLastYear                        -0.018350 -0.039213    -
0.018396
WorkLifeBalance                               0.030422  0.002726    -
```

```
0.013811
YearsAtCompany                          0.012338 -0.046018   -
0.032827
YearsInCurrentRole                      0.029218 -0.028101   -
0.035899
YearsSinceLastPromotion                 0.038031 -0.016131   -
0.062271
YearsWithCurrManager                    0.006417 -0.027972   -
0.022931

                          JobInvolvement  JobLevel   JobRole
JobSatisfaction  \
Age                             0.032323  0.442350 -0.116758
0.010038
BusinessTravel                  0.018230  0.003401 -0.002615         -
0.033026
DailyRate                       0.041841  0.015931 -0.013156
0.044460
Department                     -0.025121  0.200829  0.681597
0.030615
DistanceFromHome                0.038096  0.024038  0.010044         -
0.020165
Education                       0.042166  0.103834  0.016548         -
0.005640
EducationField                 -0.007969 -0.026676  0.050693         -
0.050693
EnvironmentSatisfaction        -0.020953  0.010615 -0.022464         -
0.009553
Gender                          0.014878 -0.058378 -0.036436
0.038130
HourlyRate                      0.051979 -0.039909 -0.023758         -
0.067797
JobInvolvement                  1.000000 -0.010769  0.004055
0.005571
JobLevel                       -0.010769  1.000000 -0.044347
0.012351
JobRole                         0.004055 -0.044347  1.000000
0.015425
JobSatisfaction                 0.005571  0.012351  0.015425
1.000000
MaritalStatus                  -0.048981 -0.074952  0.063515
0.022030
MonthlyIncome                  -0.010613  0.901390 -0.061166
0.009841
MonthlyRate                    -0.003145  0.059076  0.000150         -
0.008176
NumCompaniesWorked              0.006161  0.172878 -0.064168         -
0.045834
OverTime                        0.002521  0.002946  0.043191
```

```
0.027862
PercentSalaryHike              -0.009987 -0.027772  0.007921
0.017539
PerformanceRating              -0.023995 -0.021943 -0.021340
0.006257
RelationshipSatisfaction        0.038450  0.001790 -0.022498         -
0.011745
StockOptionLevel                0.036543  0.051470 -0.022612
0.006946
TotalWorkingYears               0.013791  0.704215 -0.135182         -
0.000852
TrainingTimesLastYear          -0.012595 -0.010041  0.004225         -
0.018180
WorkLifeBalance                -0.008334  0.048855  0.012521         -
0.024821
YearsAtCompany                  0.023893  0.409496 -0.040080
0.030234
YearsInCurrentRole              0.023724  0.324336  0.007195
0.018021
YearsSinceLastPromotion        -0.006630  0.195445  0.000737
0.026805
YearsWithCurrManager            0.052822  0.315914 -0.016941
0.004270

                          MaritalStatus  MonthlyIncome  MonthlyRate  \
Age                           -0.117182       0.452513     0.020538
BusinessTravel                 0.010108       0.030793    -0.008138
DailyRate                     -0.076058       0.029944    -0.032890
Department                     0.052696       0.152234     0.023941
DistanceFromHome              -0.027285       0.000545     0.047736
Education                     -0.012237       0.112084    -0.018874
EducationField                 0.013433      -0.020033    -0.027785
EnvironmentSatisfaction       -0.012356      -0.011976     0.036843
Gender                        -0.056779      -0.052340    -0.047240
HourlyRate                    -0.008966      -0.023613    -0.011438
JobInvolvement                -0.048981      -0.010613    -0.003145
JobLevel                      -0.074952       0.901390     0.059076
JobRole                        0.063515      -0.061166     0.000150
JobSatisfaction                0.022030       0.009841    -0.008176
MaritalStatus                  1.000000      -0.077609     0.029672
MonthlyIncome                 -0.077609       1.000000     0.050687
MonthlyRate                    0.029672       0.050687     1.000000
NumCompaniesWorked            -0.055310       0.189833     0.019063
OverTime                      -0.014736       0.013261     0.004033
PercentSalaryHike              0.013717      -0.026381    -0.011841
PerformanceRating              0.005881      -0.028388    -0.022556
RelationshipSatisfaction       0.025140      -0.007392    -0.003920
StockOptionLevel              -0.741869       0.052393    -0.046690
TotalWorkingYears             -0.098675       0.716232     0.006912
```

```
TrainingTimesLastYear          0.015749      -0.034842     0.019786
WorkLifeBalance                0.019958       0.032043     0.001548
YearsAtCompany                -0.066987       0.416465    -0.034877
YearsInCurrentRole            -0.054949       0.343024    -0.008670
YearsSinceLastPromotion       -0.008255       0.203459    -0.019588
YearsWithCurrManager          -0.049952       0.320449    -0.033583

                          NumCompaniesWorked  OverTime
PercentSalaryHike  \
Age                                 0.340022  0.028332
0.010488
BusinessTravel                      0.034013  0.010934           -
0.019175
DailyRate                           0.034923  0.020045
0.029183
Department                         -0.033131  0.015121           -
0.013541
DistanceFromHome                   -0.010318  0.036524
0.034946
Education                           0.136101 -0.015248           -
0.002095
EducationField                     -0.010403  0.010335
0.000812
EnvironmentSatisfaction             0.011203  0.058274           -
0.027743
Gender                             -0.033345 -0.051558
0.010984
HourlyRate                          0.019917 -0.003232           -
0.015826
JobInvolvement                      0.006161  0.002521           -
0.009987
JobLevel                            0.172878  0.002946           -
0.027772
JobRole                            -0.064168  0.043191
0.007921
JobSatisfaction                    -0.045834  0.027862
0.017539
MaritalStatus                      -0.055310 -0.014736
0.013717
MonthlyIncome                       0.189833  0.013261           -
0.026381
MonthlyRate                         0.019063  0.004033           -
0.011841
NumCompaniesWorked                  1.000000 -0.008714
0.001821
OverTime                           -0.008714  1.000000           -
0.011486
PercentSalaryHike                   0.001821 -0.011486
1.000000
```

```
PerformanceRating                      -0.006686  0.011654
0.648102
RelationshipSatisfaction                0.041253  0.048337           -
0.032962
StockOptionLevel                        0.038411 -0.011216
0.021635
TotalWorkingYears                       0.322501  0.010003           -
0.021813
TrainingTimesLastYear                  -0.056832 -0.076271           -
0.011397
WorkLifeBalance                         0.020077 -0.035715           -
0.015558
YearsAtCompany                         -0.178518 -0.030449           -
0.053612
YearsInCurrentRole                     -0.136304 -0.031265           -
0.028763
YearsSinceLastPromotion                -0.082304 -0.010815           -
0.055194
YearsWithCurrManager                   -0.154424 -0.030906           -
0.028292

                          PerformanceRating  RelationshipSatisfaction
\
Age                               -0.002365                  0.037296

BusinessTravel                    -0.021061                 -0.036165

DailyRate                          0.000687                  0.005771

Department                        -0.038429                 -0.037572

DistanceFromHome                   0.013212                  0.009379

Education                         -0.023157                 -0.004863

EducationField                    -0.001393                 -0.018254

EnvironmentSatisfaction           -0.024853                  0.016892

Gender                            -0.010757                  0.041439

HourlyRate                        -0.006571                  0.005207

JobInvolvement                    -0.023995                  0.038450

JobLevel                          -0.021943                  0.001790

JobRole                           -0.021340                 -0.022498

JobSatisfaction                    0.006257                 -0.011745
```

MaritalStatus	0.005881	0.025140
MonthlyIncome	-0.028388	-0.007392
MonthlyRate	-0.022556	-0.003920
NumCompaniesWorked	-0.006686	0.041253
OverTime	0.011654	0.048337
PercentSalaryHike	0.648102	-0.032962
PerformanceRating	1.000000	-0.028629
RelationshipSatisfaction	-0.028629	1.000000
StockOptionLevel	0.013927	-0.056524
TotalWorkingYears	0.005734	-0.007637
TrainingTimesLastYear	-0.016277	-0.001554
WorkLifeBalance	0.006043	0.021454
YearsAtCompany	0.010712	-0.019388
YearsInCurrentRole	0.018772	-0.027391
YearsSinceLastPromotion	-0.019851	0.010034
YearsWithCurrManager	0.013494	-0.006855

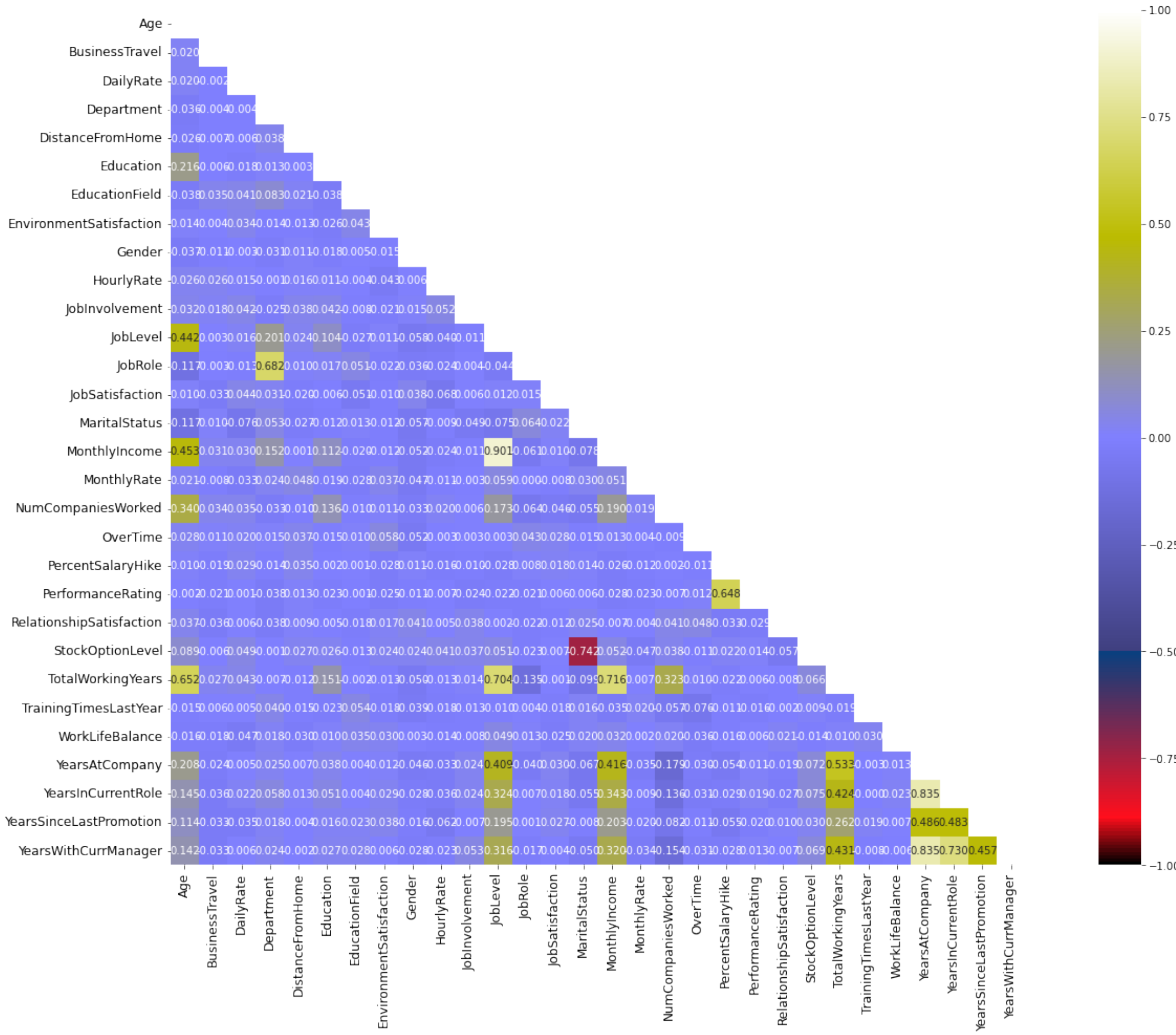
	StockOptionLevel	TotalWorkingYears \
Age	0.089449	0.652405
BusinessTravel	-0.006092	0.027298
DailyRate	0.049415	0.042750
Department	-0.000630	-0.006833
DistanceFromHome	0.027082	-0.012129
Education	0.025621	0.150720
EducationField	-0.012936	-0.001827
EnvironmentSatisfaction	0.024345	-0.013356
Gender	0.024390	-0.049776
HourlyRate	0.041329	-0.012902
JobInvolvement	0.036543	0.013791
JobLevel	0.051470	0.704215
JobRole	-0.022612	-0.135182
JobSatisfaction	0.006946	-0.000852
MaritalStatus	-0.741869	-0.098675
MonthlyIncome	0.052393	0.716232
MonthlyRate	-0.046690	0.006912

```
NumCompaniesWorked               0.038411           0.322501
OverTime                        -0.011216           0.010003
PercentSalaryHike                0.021635          -0.021813
PerformanceRating                0.013927           0.005734
RelationshipSatisfaction        -0.056524          -0.007637
StockOptionLevel                 1.000000           0.066322
TotalWorkingYears                0.066322           1.000000
TrainingTimesLastYear            0.008981          -0.019417
WorkLifeBalance                 -0.013760           0.010194
YearsAtCompany                   0.071910           0.532838
YearsInCurrentRole               0.075323           0.423676
YearsSinceLastPromotion          0.029988           0.262110
YearsWithCurrManager             0.069315           0.430605

                          TrainingTimesLastYear  WorkLifeBalance  \
Age                                   -0.014951        -0.016180
BusinessTravel                         0.006192        -0.017977
DailyRate                              0.005118        -0.046550
Department                             0.039938         0.017807
DistanceFromHome                      -0.015334        -0.030011
Education                             -0.023039         0.010164
EducationField                         0.054321         0.034788
EnvironmentSatisfaction               -0.018350         0.030422
Gender                                -0.039213         0.002726
HourlyRate                            -0.018396        -0.013811
JobInvolvement                        -0.012595        -0.008334
JobLevel                              -0.010041         0.048855
JobRole                                0.004225         0.012521
JobSatisfaction                       -0.018180        -0.024821
MaritalStatus                          0.015749         0.019958
MonthlyIncome                         -0.034842         0.032043
MonthlyRate                            0.019786         0.001548
NumCompaniesWorked                    -0.056832         0.020077
OverTime                              -0.076271        -0.035715
PercentSalaryHike                     -0.011397        -0.015558
PerformanceRating                     -0.016277         0.006043
RelationshipSatisfaction              -0.001554         0.021454
StockOptionLevel                       0.008981        -0.013760
TotalWorkingYears                     -0.019417         0.010194
TrainingTimesLastYear                  1.000000         0.029674
WorkLifeBalance                        0.029674         1.000000
YearsAtCompany                        -0.002893         0.012999
YearsInCurrentRole                    -0.000389         0.023079
YearsSinceLastPromotion                0.018813         0.007065
YearsWithCurrManager                  -0.008266        -0.006078

                          YearsAtCompany  YearsInCurrentRole  \
Age                             0.207538            0.145404
BusinessTravel                 -0.024021           -0.035610
```

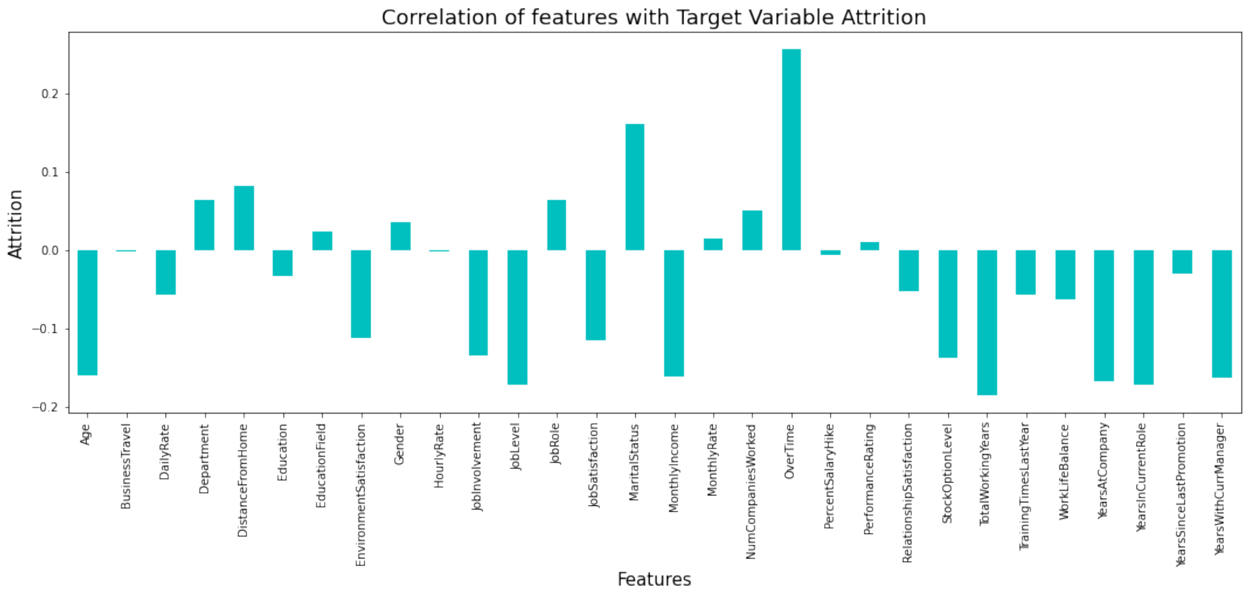
```
DailyRate                         0.005391            0.022143
Department                        0.025457            0.057817
DistanceFromHome                  0.006570            0.013091
Education                         0.037921            0.051072
EducationField                    0.004483            0.004372
EnvironmentSatisfaction           0.012338            0.029218
Gender                           -0.046018           -0.028101
HourlyRate                       -0.032827           -0.035899
JobInvolvement                    0.023893            0.023724
JobLevel                          0.409496            0.324336
JobRole                          -0.040080            0.007195
JobSatisfaction                   0.030234            0.018021
MaritalStatus                    -0.066987           -0.054949
MonthlyIncome                     0.416465            0.343024
MonthlyRate                      -0.034877           -0.008670
NumCompaniesWorked               -0.178518           -0.136304
OverTime                         -0.030449           -0.031265
PercentSalaryHike                -0.053612           -0.028763
PerformanceRating                 0.010712            0.018772
RelationshipSatisfaction         -0.019388           -0.027391
StockOptionLevel                  0.071910            0.075323
TotalWorkingYears                 0.532838            0.423676
TrainingTimesLastYear            -0.002893           -0.000389
WorkLifeBalance                   0.012999            0.023079
YearsAtCompany                    1.000000            0.835352
YearsInCurrentRole                0.835352            1.000000
YearsSinceLastPromotion           0.486029            0.482746
YearsWithCurrManager              0.835219            0.729727

                          YearsSinceLastPromotion
YearsWithCurrManager
Age                                      0.114162
0.142446
BusinessTravel                          -0.033148              -
0.032665
DailyRate                               -0.035448
0.005908
Department                               0.017699
0.024241
DistanceFromHome                        -0.003873              -
0.002310
Education                                0.016076
0.026651
EducationField                           0.023062
0.028189
EnvironmentSatisfaction                  0.038031
0.006417
Gender                                  -0.016131              -
0.027972
HourlyRate                              -0.062271              -
```

```
0.022931
JobInvolvement                                  -0.006630
0.052822
JobLevel                                         0.195445
0.315914
JobRole                                          0.000737              -
0.016941
JobSatisfaction                                  0.026805
0.004270
MaritalStatus                                   -0.008255              -
0.049952
MonthlyIncome                                    0.203459
0.320449
MonthlyRate                                     -0.019588              -
0.033583
NumCompaniesWorked                              -0.082304              -
0.154424
OverTime                                        -0.010815              -
0.030906
PercentSalaryHike                               -0.055194              -
0.028292
PerformanceRating                               -0.019851
0.013494
RelationshipSatisfaction                         0.010034              -
0.006855
StockOptionLevel                                 0.029988
0.069315
TotalWorkingYears                                0.262110
0.430605
TrainingTimesLastYear                            0.018813              -
0.008266
WorkLifeBalance                                  0.007065              -
0.006078
YearsAtCompany                                   0.486029
0.835219
YearsInCurrentRole                               0.482746
0.729727
YearsSinceLastPromotion                          1.000000
0.456672
YearsWithCurrManager                             0.456672
1.000000

upper_triangle = np.triu(df.corr())
plt.figure(figsize=(25,15))
sns.heatmap(df.corr(), vmin=-1, vmax=1, annot=True, square=True,
fmt='0.3f',
            annot_kws={'size':10}, cmap="gist_stern",
mask=upper_triangle)
plt.xticks(fontsize=12)
```

```
plt.yticks(fontsize=12)
plt.show()
```




```
plt.figure(figsize = (18,6))
df1.corr()['Attrition'].drop(['Attrition']).plot(kind='bar',color =
'c')
plt.xlabel('Features',fontsize=15)
plt.ylabel('Attrition',fontsize=15)
plt.title('Correlation of features with Target Variable
Attrition',fontsize = 18)
plt.show()
```



Comment:

- Age, JobLevel, MonthlyIncome is highly positively correlated with TotalWorkingYears.
- JobLevel is highly positively correlated with the MonthlyIncome.
- PercentSalaryHike is highly positively correlated with the column PerformanceRating.

3. Checking Multicollinearity between features using variance_inflation_factor

```
from statsmodels.stats.outliers_influence import
variance_inflation_factor
vif= pd.DataFrame()
vif['VIF']= [variance_inflation_factor(df.values,i) for i in
range(df.shape[1])]
vif['Features']= df.columns
vif
```

	VIF	Features
0	1.930457	Age
1	1.014314	BusinessTravel
2	1.025841	DailyRate
3	2.172093	Department
4	1.017385	DistanceFromHome
5	1.065266	Education
6	1.030480	EducationField
7	1.024396	EnvironmentSatisfaction
8	1.024366	Gender
9	1.024189	HourlyRate
10	1.020167	JobInvolvement
11	5.976707	JobLevel
12	2.023213	JobRole
13	1.023909	JobSatisfaction

```
14  2.298943              MaritalStatus
15  5.842828              MonthlyIncome
16  1.022108                MonthlyRate
17  1.426763         NumCompaniesWorked
18  1.028400                   OverTime
19  1.016867          PercentSalaryHike
20  1.747581          PerformanceRating
21  1.022260   RelationshipSatisfaction
22  2.279101           StockOptionLevel
23  4.093506          TotalWorkingYears
24  1.025519      TrainingTimesLastYear
25  1.017093            WorkLifeBalance
26  6.296064             YearsAtCompany
27  3.513852         YearsInCurrentRole
28  1.373189    YearsSinceLastPromotion
29  3.433437       YearsWithCurrManager
```

Comment :

- We can see that multicollinerity is within permissible limit of 10.

Balancing using SMOTE

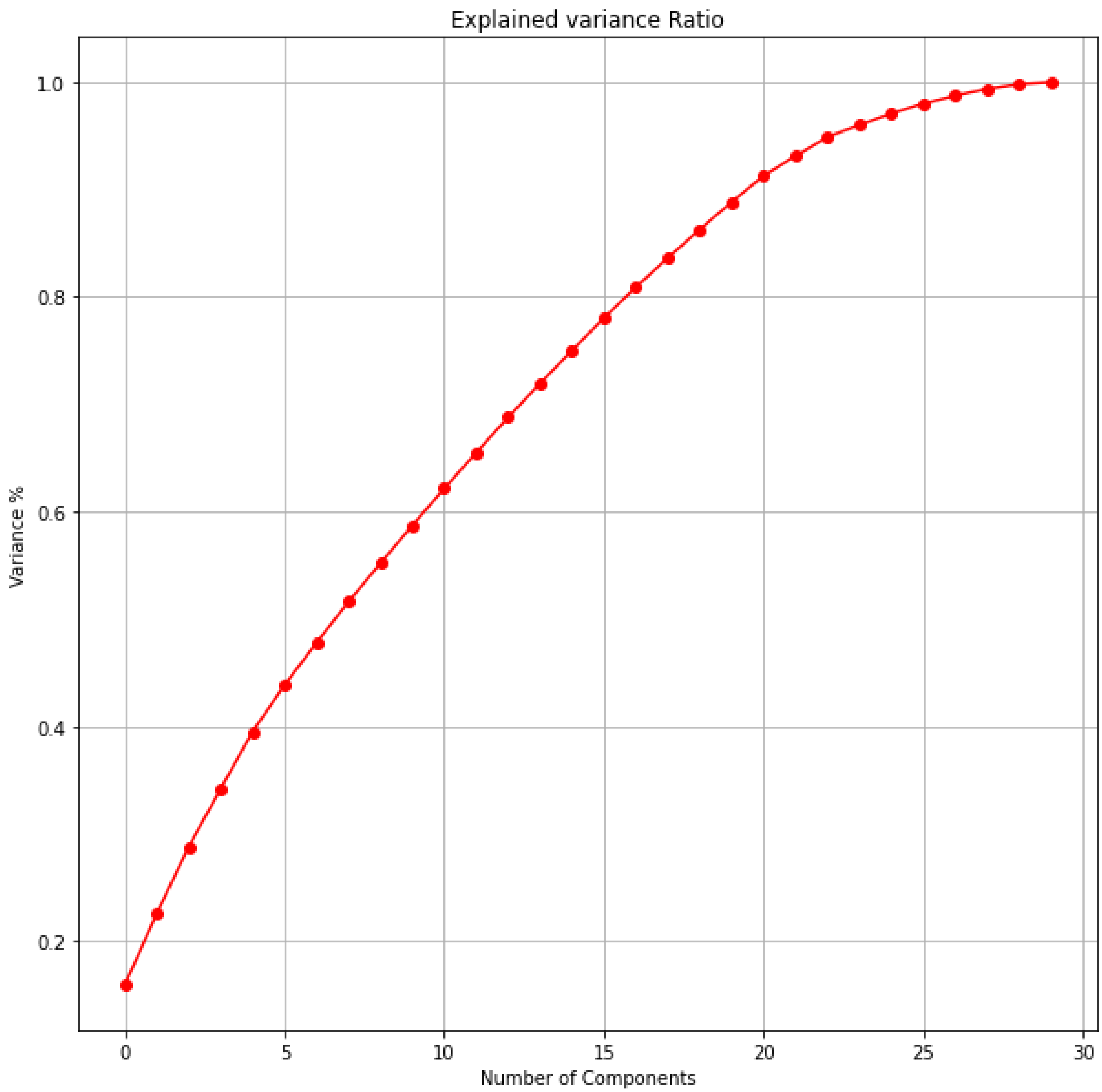
As data is Imbalanced in nature we will need to balance target variable.

```
from imblearn.over_sampling import SMOTE

# Oversampleing using SMOTE Techniques
oversample = SMOTE()
X, Y = oversample.fit_resample(X, Y)

Y.value_counts()

1    1158
0    1158
Name: Attrition, dtype: int64
```

Standard Scaling

```
from sklearn.preprocessing import StandardScaler
scaler= StandardScaler()
X_scale = scaler.fit_transform(X)

from sklearn.decomposition import PCA
pca = PCA()
#plot the graph to find the principal components
x_pca = pca.fit_transform(X_scale)
plt.figure(figsize=(10,10))
plt.plot(np.cumsum(pca.explained_variance_ratio_), 'ro-')
plt.xlabel('Number of Components')
```

```
plt.ylabel('Variance %')
plt.title('Explained variance Ratio')
plt.grid()
```



Comment -

AS per the graph, we can see that 21 principal components attribute for 90% of variation in the data. We shall pick the first 21 components for our prediction

```
pca_new = PCA(n_components=21)
x_new = pca_new.fit_transform(X_scale)
```

```
principle_x=pd.DataFrame(x_new,columns=np.arange(21))
```

Machine Learning Model Building

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score,
confusion_matrix,classification_report,f1_score

from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import BaggingClassifier

X_train, X_test, Y_train, Y_test = train_test_split(principle_x, Y,
random_state=42, test_size=.33)
print('Training feature matrix size:',X_train.shape)
print('Training target vector size:',Y_train.shape)
print('Test feature matrix size:',X_test.shape)
print('Test target vector size:',Y_test.shape)
```

```
Training feature matrix size: (1551, 21)
Training target vector size: (1551,)
Test feature matrix size: (765, 21)
Test target vector size: (765,)
```

Finding best Random state

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,
confusion_matrix,classification_report,f1_score
maxAccu=0
maxRS=0
for i in range(1,250):
    X_train,X_test,Y_train,Y_test =
train_test_split(principle_x,Y,test_size = 0.33, random_state=i)
    log_reg=LogisticRegression()
    log_reg.fit(X_train,Y_train)
    y_pred=log_reg.predict(X_test)
    acc=accuracy_score(Y_test,y_pred)
    if acc>maxAccu:
        maxAccu=acc
```

```
        maxRS=i
print('Best accuracy is', maxAccu ,'on Random_state', maxRS)
```

```
Best accuracy is 0.8849673202614379 on Random_state 75
```

```
X_train, X_test, Y_train, Y_test = train_test_split(principle_x, Y,
random_state=242, test_size=.33)
log_reg=LogisticRegression()
log_reg.fit(X_train,Y_train)
y_pred=log_reg.predict(X_test)
print('\033[1m'+'Logistics Regression Evaluation'+'\033[0m')
print('\n')
print('\033[1m'+'Accuracy Score of Logistics Regression :'+'\033[0m',
accuracy_score(Y_test, y_pred))
print('\n')
print('\033[1m'+'Confusion matrix of Logistics Regression :'+'\033[0m
\n',confusion_matrix(Y_test, y_pred))
print('\n')
print('\033[1m'+'classification Report of Logistics Regression'+'\
033[0m \n',classification_report(Y_test, y_pred))
```

```
Logistics Regression Evaluation


Accuracy Score of Logistics Regression : 0.8745098039215686


Confusion matrix of Logistics Regression :
 [[326  42]
 [ 54 343]]


classification Report of Logistics Regression
               precision    recall  f1-score   support

           0       0.86      0.89      0.87       368
           1       0.89      0.86      0.88       397

    accuracy                           0.87       765
   macro avg       0.87      0.87      0.87       765
weighted avg       0.88      0.87      0.87       765
```

Finding Optimal value of n_neighbors for KNN

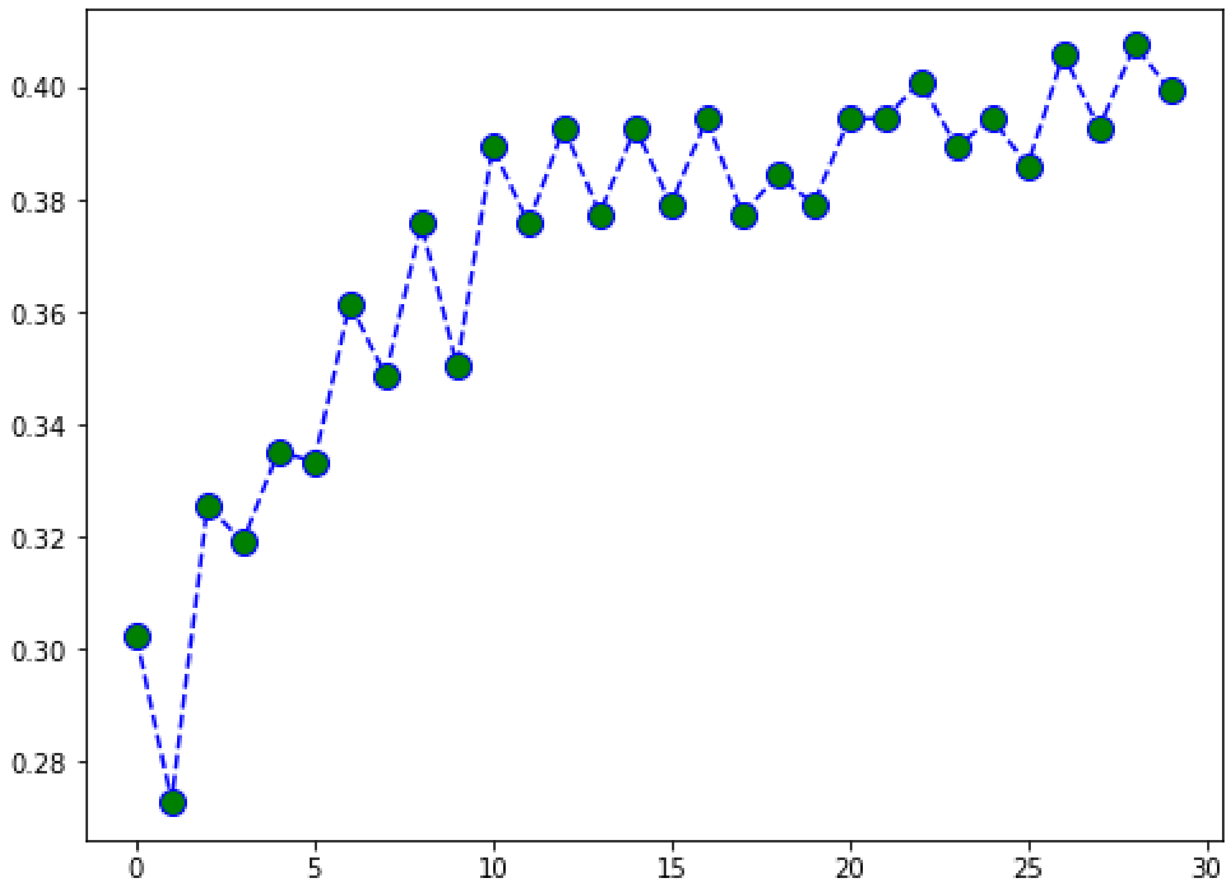
```
from sklearn import neighbors
from math import sqrt
from sklearn.metrics import mean_squared_error
rmse_val = [] #to store rmse values for different k
for K in range(30):
    K = K+1
```

```python
    model = neighbors.KNeighborsClassifier(n_neighbors = K)

    model.fit(X_train,Y_train)  #fit the model
    y_pred=model.predict(X_test) #make prediction on test set
    error = sqrt(mean_squared_error(Y_test,y_pred)) #calculate rmse
    rmse_val.append(error) #store rmse values
    print('RMSE value for k= ' , K , 'is:', error)
```

```
RMSE value for k=  1 is: 0.30249507099101003
RMSE value for k=  2 is: 0.27296484008305655
RMSE value for k=  3 is: 0.32539568672798425
RMSE value for k=  4 is: 0.31931298801289854
RMSE value for k=  5 is: 0.3352883843105734
RMSE value for k=  6 is: 0.3333333333333333
RMSE value for k=  7 is: 0.3615507630310936
RMSE value for k=  8 is: 0.34866692910423897
RMSE value for k=  9 is: 0.37573457465108967
RMSE value for k=  10 is: 0.3505364702758674
RMSE value for k=  11 is: 0.3894020890135344
RMSE value for k=  12 is: 0.37573457465108967
RMSE value for k=  13 is: 0.3927446575232716
RMSE value for k=  14 is: 0.37747007845751024
RMSE value for k=  15 is: 0.3927446575232716
RMSE value for k=  16 is: 0.3791976393296807
RMSE value for k=  17 is: 0.39440531887330776
RMSE value for k=  18 is: 0.37747007845751024
RMSE value for k=  19 is: 0.38433373297259976
RMSE value for k=  20 is: 0.3791976393296807
RMSE value for k=  21 is: 0.39440531887330776
RMSE value for k=  22 is: 0.39440531887330776
RMSE value for k=  23 is: 0.4009791936316524
RMSE value for k=  24 is: 0.3894020890135344
RMSE value for k=  25 is: 0.39440531887330776
RMSE value for k=  26 is: 0.3860305788964616
RMSE value for k=  27 is: 0.4058397249567139
RMSE value for k=  28 is: 0.3927446575232716
RMSE value for k=  29 is: 0.40744701728620475
RMSE value for k=  30 is: 0.39934587037179503
```

```python
#plotting the rmse values against k values -
plt.figure(figsize = (8,6))
plt.plot(range(30), rmse_val, color='blue', linestyle='dashed',
marker='o', markerfacecolor='green', markersize=10)
```

```
[<matplotlib.lines.Line2D at 0x26f7af3a340>]
```



Comment-

At k= 2, we get the minimum RMSE value which approximately 0.30032661958503204, and shoots up on further increasing the k value. We can safely say that k=2 will give us the best result in this case

---

Applying other classification algorithm

```
model=[
        SVC(),
        GaussianNB(),
        DecisionTreeClassifier(),
        KNeighborsClassifier(n_neighbors = 22),
        RandomForestClassifier(),
        AdaBoostClassifier(),
        GradientBoostingClassifier(),
        BaggingClassifier()]

for m in model:
    m.fit(X_train,Y_train)
    y_pred=m.predict(X_test)
```

```
    print('\033[1m'+'Classification ML Algorithm Evaluation
Matrix',m,'is' +'\033[0m')
    print('\n')
    print('\033[1m'+'Accuracy Score :'+'\033[0m\n',
accuracy_score(Y_test, y_pred))
    print('\n')
    print('\033[1m'+'Confusion matrix :'+'\033[0m \
n',confusion_matrix(Y_test, y_pred))
    print('\n')
    print('\033[1m'+'Classification Report :'+'\033[0m \
n',classification_report(Y_test, y_pred))
    print('\n')

print('======================================================================
=============================================')
```

```
Classification ML Algorithm Evaluation Matrix SVC() is


Accuracy Score :
 0.9124183006535947


Confusion matrix :
 [[339  29]
 [ 38 359]]


Classification Report :
               precision    recall  f1-score   support

           0       0.90      0.92      0.91       368
           1       0.93      0.90      0.91       397

    accuracy                           0.91       765
   macro avg       0.91      0.91      0.91       765
weighted avg       0.91      0.91      0.91       765



======================================================================
======================================
Classification ML Algorithm Evaluation Matrix GaussianNB() is


Accuracy Score :
 0.857516339869281


Confusion matrix :
 [[321  47]
```

```
 [ 62 335]]


Classification Report :
               precision    recall  f1-score   support

           0       0.84      0.87      0.85       368
           1       0.88      0.84      0.86       397

    accuracy                           0.86       765
   macro avg       0.86      0.86      0.86       765
weighted avg       0.86      0.86      0.86       765



=======================================================================
======================================
Classification ML Algorithm Evaluation Matrix DecisionTreeClassifier()
is


Accuracy Score :
 0.7895424836601307


Confusion matrix :
 [[276  92]
 [ 69 328]]


Classification Report :
               precision    recall  f1-score   support

           0       0.80      0.75      0.77       368
           1       0.78      0.83      0.80       397

    accuracy                           0.79       765
   macro avg       0.79      0.79      0.79       765
weighted avg       0.79      0.79      0.79       765



=======================================================================
======================================
Classification ML Algorithm Evaluation Matrix
KNeighborsClassifier(n_neighbors=22) is


Accuracy Score :
 0.8444444444444444
```

```
Confusion matrix :
 [[281  87]
 [ 32 365]]


Classification Report :
               precision    recall  f1-score   support

           0       0.90      0.76      0.83       368
           1       0.81      0.92      0.86       397

    accuracy                           0.84       765
   macro avg       0.85      0.84      0.84       765
weighted avg       0.85      0.84      0.84       765



======================================================================
======================================
Classification ML Algorithm Evaluation Matrix RandomForestClassifier()
is


Accuracy Score :
 0.9098039215686274


Confusion matrix :
 [[338  30]
 [ 39 358]]


Classification Report :
               precision    recall  f1-score   support

           0       0.90      0.92      0.91       368
           1       0.92      0.90      0.91       397

    accuracy                           0.91       765
   macro avg       0.91      0.91      0.91       765
weighted avg       0.91      0.91      0.91       765



======================================================================
======================================
Classification ML Algorithm Evaluation Matrix AdaBoostClassifier() is


Accuracy Score :
```

```
 0.8392156862745098


Confusion matrix :
 [[309  59]
 [ 64 333]]


Classification Report :
               precision    recall  f1-score   support

           0       0.83      0.84      0.83       368
           1       0.85      0.84      0.84       397

    accuracy                           0.84       765
   macro avg       0.84      0.84      0.84       765
weighted avg       0.84      0.84      0.84       765



======================================================================
======================================
Classification ML Algorithm Evaluation Matrix
GradientBoostingClassifier() is


Accuracy Score :
 0.8862745098039215


Confusion matrix :
 [[328  40]
 [ 47 350]]


Classification Report :
               precision    recall  f1-score   support

           0       0.87      0.89      0.88       368
           1       0.90      0.88      0.89       397

    accuracy                           0.89       765
   macro avg       0.89      0.89      0.89       765
weighted avg       0.89      0.89      0.89       765



======================================================================
======================================
Classification ML Algorithm Evaluation Matrix BaggingClassifier() is
```

```
Accuracy Score :
 0.8849673202614379


Confusion matrix :
 [[330  38]
 [ 50 347]]


Classification Report :
               precision    recall  f1-score   support

           0       0.87      0.90      0.88       368
           1       0.90      0.87      0.89       397

    accuracy                           0.88       765
   macro avg       0.88      0.89      0.88       765
weighted avg       0.89      0.88      0.89       765



======================================================================
======================================
```

We can see that RandomForestClassifier() gives us good Accuracy and maximum f1 score. so we will continue further investigation with crossvalidation of above model

CrossValidation :

```
from sklearn.model_selection import cross_val_score
model=[LogisticRegression(),
        SVC(),
        GaussianNB(),
        DecisionTreeClassifier(),
        KNeighborsClassifier(n_neighbors = 12),
        RandomForestClassifier(),
        AdaBoostClassifier(),
        GradientBoostingClassifier(),
        BaggingClassifier()]

for m in model:
    score = cross_val_score(m, X, Y, cv =5)
    print('\n')
    print('\033[1m'+'Cross Validation Score', m, ':'+'\033[0m\n')
    print("Score :" ,score)
    print("Mean Score :",score.mean())
    print("Std deviation :",score.std())
    print('\n')
```

```
print('===============================================================
=============================================')
```

```
Cross Validation Score LogisticRegression() :

Score : [0.6487069  0.70194384 0.69546436 0.72354212 0.73218143]
Mean Score : 0.7003677292023534
Std deviation : 0.029135999296551227

======================================================================
======================================

Cross Validation Score SVC() :

Score : [0.57758621 0.6587473  0.58099352 0.61339093 0.61987041]
Mean Score : 0.6101176733447531
Std deviation : 0.029587778818214797

======================================================================
======================================

Cross Validation Score GaussianNB() :

Score : [0.66163793 0.78617711 0.73218143 0.79481641 0.77321814]
Mean Score : 0.7496062039174797
Std deviation : 0.04895090930297518

======================================================================
======================================

Cross Validation Score DecisionTreeClassifier() :

Score : [0.6637931  0.9049676  0.89200864 0.8574514  0.90064795]
Mean Score : 0.8437737394801518
Std deviation : 0.09152713030301529

======================================================================
======================================

Cross Validation Score KNeighborsClassifier(n_neighbors=12) :
```

```
Score : [0.6875     0.73434125 0.73434125 0.76457883 0.73866091]
Mean Score : 0.7318844492440604
Std deviation : 0.024887323106749498


======================================================================
======================================


Cross Validation Score RandomForestClassifier() :

Score : [0.69396552 0.97408207 0.96760259 0.96544276 0.98056156]
Mean Score : 0.9163309004245178
Std deviation : 0.11130849038194443


======================================================================
======================================


Cross Validation Score AdaBoostClassifier() :

Score : [0.60560345 0.93520518 0.93520518 0.92440605 0.96328294]
Mean Score : 0.8727405600655395
Std deviation : 0.13418877714534755


======================================================================
======================================


Cross Validation Score GradientBoostingClassifier() :

Score : [0.58189655 0.97192225 0.95680346 0.94600432 0.97408207]
Mean Score : 0.8861417293513071
Std deviation : 0.1524687106214308


======================================================================
======================================


Cross Validation Score BaggingClassifier() :

Score : [0.67025862 0.95680346 0.94384449 0.93520518 0.95464363]
Mean Score : 0.8921510761897669
Std deviation : 0.11121814288147412


======================================================================
======================================
```

On basis of maximum score in crossvalidation of Random Forest Classifier. we will apply Hyperparameter tuning on Random Forest model

Hyper Parameter Tuning : GridSearchCV

```
from sklearn.model_selection import GridSearchCV

parameter = {  'bootstrap': [True], 'max_depth': [5, 10,20,40,50, None],
              'max_features': ['auto', 'log2'],
              'criterion':['gini','entropy'],
              'n_estimators': [5, 10, 15 ,25,50,100]}

GCV = GridSearchCV(RandomForestClassifier(),parameter,cv=5,n_jobs = -1,verbose=3)
GCV.fit(X_train,Y_train)
```

```
Fitting 5 folds for each of 144 candidates, totalling 720 fits
```

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(), n_jobs=-1,
             param_grid={'bootstrap': [True], 'criterion': ['gini', 'entropy'],
                         'max_depth': [5, 10, 20, 40, 50, None],
                         'max_features': ['auto', 'log2'],
                         'n_estimators': [5, 10, 15, 25, 50, 100]},
             verbose=3)
```

```
GCV.best_params_
```

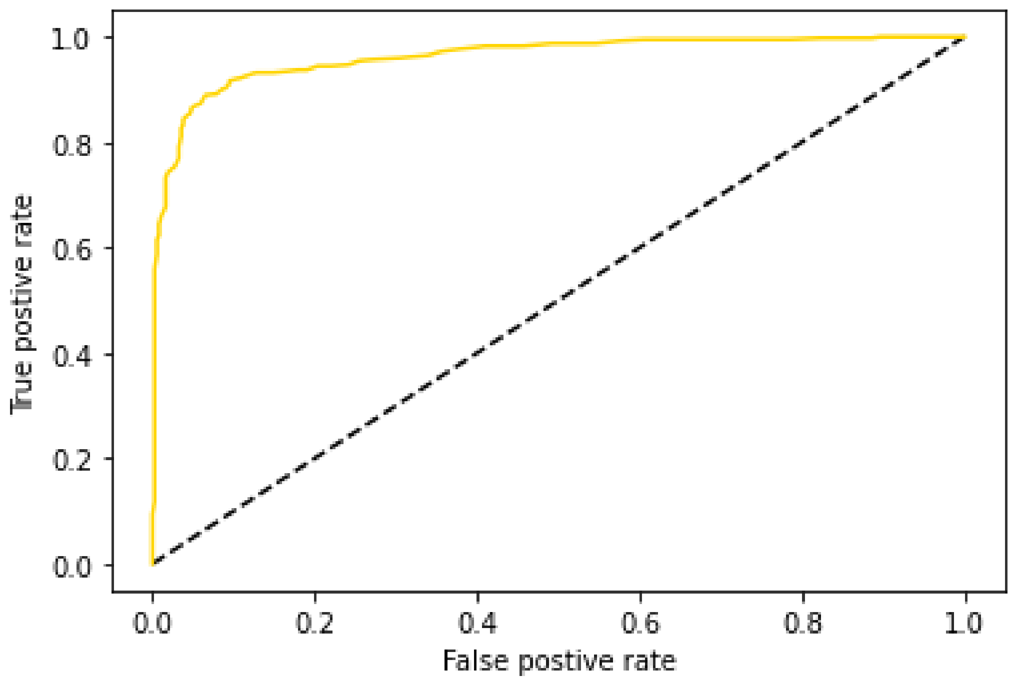
```
{'bootstrap': True,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'log2',
 'n_estimators': 100}
```

Final Model

```
Final_mod = RandomForestClassifier(bootstrap=True,criterion='gini',n_estimators=100, max_depth=None ,max_features='log2')
Final_mod.fit(X_train,Y_train)
y_pred=Final_mod.predict(X_test)
print('\033[1m'+'Accuracy Score :'+'\033[0m\n', accuracy_score(Y_test, y_pred))
```

```
Accuracy Score :
 0.9071895424836601
```

```
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve

y_pred_prob = Final_mod.predict_proba(X_test)[:,1]
fpr, tpr, thresholds = roc_curve(Y_test,y_pred_prob)
plt.plot([0,1],[0,1], 'k--')
plt.plot(fpr, tpr, label='Random Forest Classifier')
plt.xlabel('False postive rate')
plt.ylabel('True postive rate')
plt.show()
auc_score = roc_auc_score(Y_test, Final_mod.predict(X_test))
print('\033[1m'+'Auc Score :'+'\033[0m\n',auc_score)
```




```
Auc Score :
 0.9074033512211149
```

Saving model

```
import joblib
joblib.dump(Final_mod,'HR_Analytics_Final.pkl')
```

```
['HR_Analytics_Final.pkl']
```