Employee Attrition

DSC 680 - Applied Data Science

Andrews_Bill

Importing the Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

from sklearn import datasets
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_
score, confusion_matrix
from sklearn.preprocessing import MinMaxScaler, OrdinalEncoder, OneHotEncoder
from sklearn.model_selection import train_test_split, cross_val_score
import warnings
warnings.filterwarnings("ignore")
```

Loading Dataset

In [2]:
```
df = pd.read_csv('C:\\Users\\Billa\\OneDrive\\Desktop\\Project DSC 680\\Project 2\\WA_Fn-UseC_-HR-Employee-Attrition.csv')
df.head()
```

Out[2]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educ
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Lif
1	49	No	Travel_Frequently	279	Research & Development	8	1	Lif
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Lif
4	27	No	Travel_Rarely	591	Research & Development	2	1	

5 rows × 35 columns

Exploring the Dataset

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   Age                       1470 non-null   int64 
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64 
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64 
 6   Education                 1470 non-null   int64 
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64 
 9   EmployeeNumber            1470 non-null   int64 
 10  EnvironmentSatisfaction   1470 non-null   int64 
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64 
 13  JobInvolvement            1470 non-null   int64 
 14  JobLevel                  1470 non-null   int64 
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64 
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64 
 19  MonthlyRate               1470 non-null   int64 
 20  NumCompaniesWorked        1470 non-null   int64 
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64 
 24  PerformanceRating         1470 non-null   int64 
 25  RelationshipSatisfaction  1470 non-null   int64 
 26  StandardHours             1470 non-null   int64 
 27  StockOptionLevel          1470 non-null   int64 
 28  TotalWorkingYears         1470 non-null   int64 
 29  TrainingTimesLastYear     1470 non-null   int64 
 30  WorkLifeBalance           1470 non-null   int64 
 31  YearsAtCompany            1470 non-null   int64 
 32  YearsInCurrentRole        1470 non-null   int64 
 33  YearsSinceLastPromotion   1470 non-null   int64 
 34  YearsWithCurrManager      1470 non-null   int64 
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

In [4]:
```
#Get the number of rows and number of columns in the data
df.shape
```

Out[4]:
```
(1470, 35)
```

In [5]:
```
df.sample(20)
```

Out[5]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	E
1343	29	No	Travel_Rarely	592	Research & Development	7	3	
1229	40	No	Travel_Rarely	369	Research & Development	8	2	
1096	40	No	Travel_Rarely	898	Human Resources	6	2	
903	29	No	Travel_Rarely	1329	Research & Development	7	3	
1100	35	No	Travel_Rarely	1402	Sales	28	4	
642	38	No	Travel_Rarely	395	Sales	9	3	
193	43	No	Non-Travel	1344	Research & Development	7	3	
350	42	No	Travel_Rarely	544	Human Resources	2	1	
701	53	No	Travel_Rarely	1376	Sales	2	2	
796	25	Yes	Travel_Rarely	1219	Research & Development	4	1	
30	33	No	Travel_Rarely	924	Research & Development	2	3	
1403	39	No	Travel_Rarely	119	Sales	15	4	
414	24	Yes	Travel_Rarely	1448	Sales	1	1	
218	45	No	Non-Travel	1052	Sales	6	3	
627	52	No	Travel_Frequently	890	Research & Development	25	4	
392	54	No	Travel_Rarely	821	Research & Development	5	2	
731	20	Yes	Travel_Rarely	1097	Research & Development	11	3	
852	29	No	Travel_Rarely	1401	Research & Development	6	1	
792	33	Yes	Travel_Frequently	827	Research & Development	29	4	
716	41	No	Travel_Frequently	840	Research & Development	9	3	

20 rows × 35 columns

In [6]:
```
#View to provide basic statistical details
df.describe()
```

Out[6]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNu
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.00
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.86
std	9.135373	403.509100	8.106864	1.024165	0.0	602.02
min	18.000000	102.000000	1.000000	1.000000	1.0	1.00
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.25
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.50
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.75
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.00

8 rows × 26 columns

In [7]:
```python
#Checking for missing data
df.isnull().sum()
```

Out[7]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

In [8]:
```python
#Get a count of the number of employee attrition
df['Attrition'].value_counts()
```

Out[8]:
```
No     1233
Yes     237
Name: Attrition, dtype: int64
```

EmployeeNumber, StandardHours,Over18 and EmployeeCount columns contain one value which provides no value. They will be removed.

In [9]:
```python
df = df.drop('EmployeeNumber', axis = 1)
df = df.drop('StandardHours', axis = 1)
df = df.drop('EmployeeCount', axis = 1)
df = df.drop('Over18', axis = 1)
```

In [10]:
```
#Get the correlation of the columns
df.corr()
```

Out[10]:

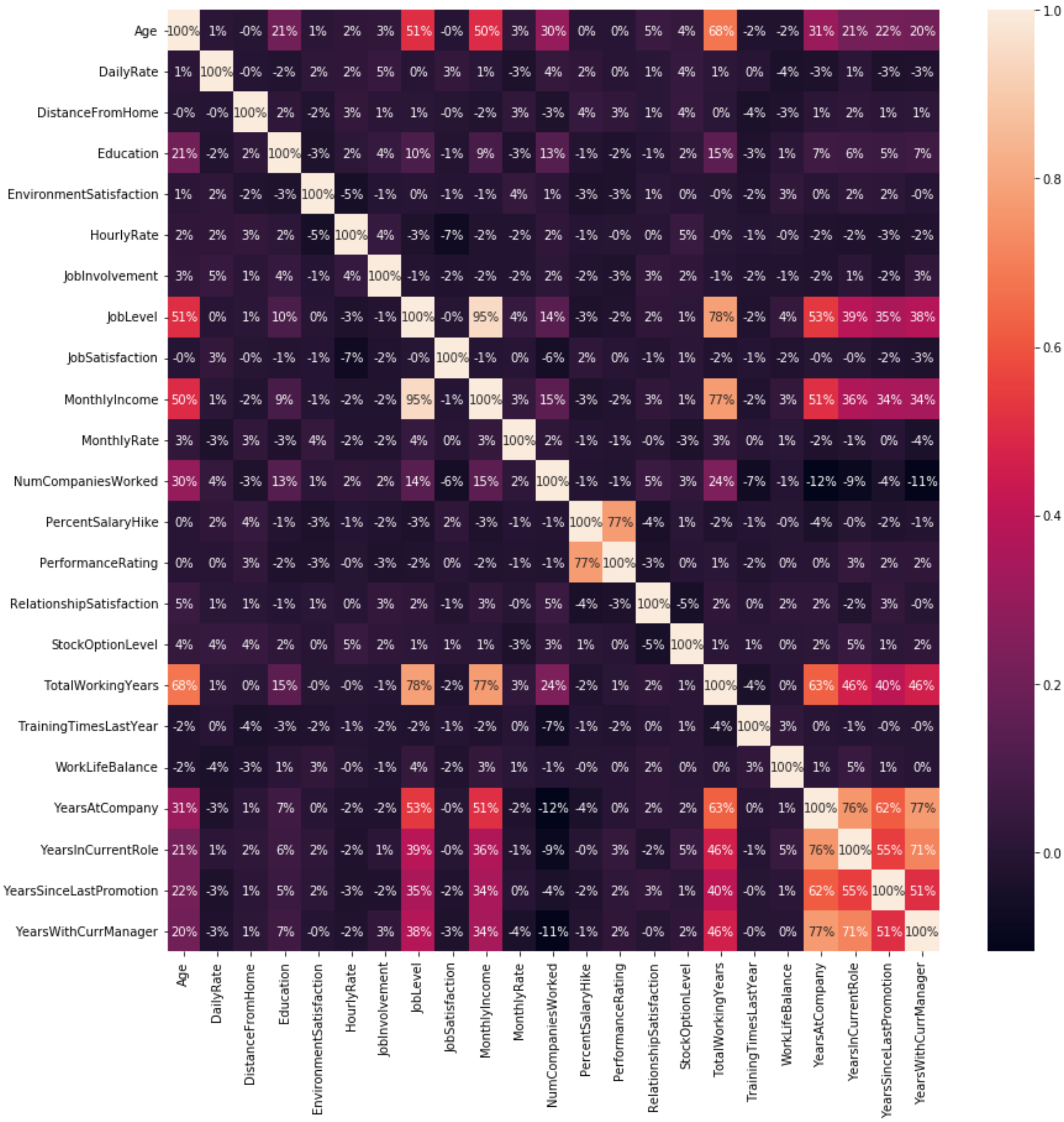
	Age	DailyRate	DistanceFromHome	Education	EnvironmentSatisf
Age	1.000000	0.010661	-0.001686	0.208034	0.C
DailyRate	0.010661	1.000000	-0.004985	-0.016806	0.C
DistanceFromHome	-0.001686	-0.004985	1.000000	0.021042	-0.C
Education	0.208034	-0.016806	0.021042	1.000000	-0.C
EnvironmentSatisfaction	0.010146	0.018355	-0.016075	-0.027128	1.C
HourlyRate	0.024287	0.023381	0.031131	0.016775	-0.C
JobInvolvement	0.029820	0.046135	0.008783	0.042438	-0.C
JobLevel	0.509604	0.002966	0.005303	0.101589	0.C
JobSatisfaction	-0.004892	0.030571	-0.003669	-0.011296	-0.C
MonthlyIncome	0.497855	0.007707	-0.017014	0.094961	-0.C
MonthlyRate	0.028051	-0.032182	0.027473	-0.026084	0.C
NumCompaniesWorked	0.299635	0.038153	-0.029251	0.126317	0.C
PercentSalaryHike	0.003634	0.022704	0.040235	-0.011111	-0.C
PerformanceRating	0.001904	0.000473	0.027110	-0.024539	-0.C
RelationshipSatisfaction	0.053535	0.007846	0.006557	-0.009118	0.C
StockOptionLevel	0.037510	0.042143	0.044872	0.018422	0.C
TotalWorkingYears	0.680381	0.014515	0.004628	0.148280	-0.C
TrainingTimesLastYear	-0.019621	0.002453	-0.036942	-0.025100	-0.C
WorkLifeBalance	-0.021490	-0.037848	-0.026556	0.009819	0.C
YearsAtCompany	0.311309	-0.034055	0.009508	0.069114	0.C
YearsInCurrentRole	0.212901	0.009932	0.018845	0.060236	0.C
YearsSinceLastPromotion	0.216513	-0.033229	0.010029	0.054254	0.C
YearsWithCurrManager	0.202089	-0.026363	0.014406	0.069065	-0.C

23 rows × 23 columns

EDA and Data Visualization

In [11]:
```
#Visualize the correlation
plt.figure(figsize=(14,14))
sns.heatmap(df.corr(), annot=True, fmt='.0%')
```

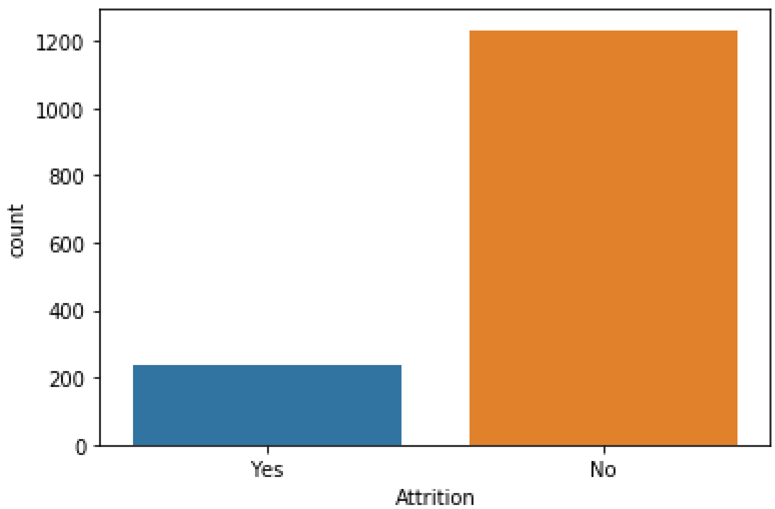
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x27e960ad080>



Correlation between Total Working Years with MonthlyIncome and JobLevel YearsAtCompany with YearsWithCurrManager and YearsInCurrentRole

In [12]:
```
#Visualize this count
sns.countplot(df['Attrition'])
```

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x27e97092048>



In [13]:

```
# Information on dataset
print(f'The dataset has {df.shape[0]} rows for each employee and {df.shape[1]}
attributes\n')
print('='*80)
print('The attributes and their unique values are as below\n')
for i in df.columns:
    print(f'======== {i} ========\n')
    print(df[i].value_counts().sort_values(ascending=False))
    print('- - ' * 20)
```

```
The dataset has 1470 rows for each employee and 31 attributes

================================================================================
===
The attributes and their unique values are as below

======== Age ========

35    78
34    77
31    69
36    69
29    68
32    61
30    60
38    58
33    58
40    57
37    50
27    48
28    48
42    46
39    42
45    41
41    40
26    39
46    33
44    33
43    32
50    30
24    26
25    26
47    24
49    24
55    22
48    19
51    19
53    19
54    18
52    18
22    16
56    14
58    14
23    14
21    13
20    11
59    10
19     9
18     8
60     5
57     4
Name: Age, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== Attrition ========

No     1233
```

```
Yes     237
Name: Attrition, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== BusinessTravel ========

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

691     6
1082    5
329     5
1329    5
530     5
       ..
1193    1
1198    1
1172    1
1202    1
314     1
Name: DailyRate, Length: 886, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== Department ========

Research & Development    961
Sales                     446
Human Resources            63
Name: Department, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== DistanceFromHome ========

2     211
1     208
10     86
9      85
3      84
7      84
8      80
5      65
4      64
6      59
16     32
11     29
24     28
29     27
23     27
18     26
15     26
20     25
25     25
```

```
26     25
28     23
19     22
14     21
12     20
17     20
13     19
22     19
21     18
27     12
Name: DistanceFromHome, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== Education ========

3    572
4    398
2    282
1    170
5     48
Name: Education, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== EducationField ========

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

3    453
4    446
2    287
1    284
Name: EnvironmentSatisfaction, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== Gender ========

Male      882
Female    588
Name: Gender, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== HourlyRate ========

66    29
42    28
98    28
48    28
84    28
```

```
      ..
69    15
53    14
68    14
38    13
34    12
Name: HourlyRate, Length: 71, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== JobInvolvement ========

3    868
2    375
4    144
1     83
Name: JobInvolvement, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== JobLevel ========

1    543
2    534
3    218
4    106
5     69
Name: JobLevel, 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
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== JobSatisfaction ========

4    459
3    442
1    289
2    280
Name: JobSatisfaction, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== MaritalStatus ========

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

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== MonthlyIncome ========

2342     4
2380     3
2404     3
2559     3
6347     3
        ..
5042     1
4256     1
5326     1
13116    1
4450     1
Name: MonthlyIncome, Length: 1349, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== MonthlyRate ========

4223     3
9150     3
6319     2
22102    2
10494    2
        ..
10224    1
10227    1
22495    1
18420    1
16734    1
Name: MonthlyRate, Length: 1427, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== NumCompaniesWorked ========

1    521
0    197
3    159
2    146
4    139
7     74
6     70
5     63
9     52
8     49
Name: NumCompaniesWorked, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== OverTime ========

No     1054
Yes     416
Name: OverTime, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== PercentSalaryHike ========
```

```
11    210
13    209
14    201
12    198
15    101
18     89
17     82
16     78
19     76
22     56
20     55
21     48
23     28
24     21
25     18
Name: PercentSalaryHike, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== PerformanceRating ========

3    1244
4     226
Name: PerformanceRating, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== RelationshipSatisfaction ========

3    459
4    432
2    303
1    276
Name: RelationshipSatisfaction, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== StockOptionLevel ========

0    631
1    596
2    158
3     85
Name: StockOptionLevel, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== TotalWorkingYears ========

10    202
6     125
8     103
9      96
5      88
1      81
7      81
4      63
12     48
3      42
15     40
```

```
16      37
13      36
11      36
21      34
17      33
14      31
2       31
20      30
18      27
19      22
23      22
22      21
24      18
25      14
28      14
26      14
0       11
29      10
32       9
31       9
27       7
30       7
33       7
36       6
34       5
37       4
35       3
40       2
38       1
Name: TotalWorkingYears, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== TrainingTimesLastYear ========

2    547
3    491
4    123
5    119
1     71
6     65
0     54
Name: TrainingTimesLastYear, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== WorkLifeBalance ========

3    893
2    344
4    153
1     80
Name: WorkLifeBalance, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== YearsAtCompany ========

5     196
1     171
```

```
3     128
2     127
10    120
4     110
7      90
9      82
8      80
6      76
0      44
11     32
20     27
13     24
15     20
14     18
22     15
12     14
21     14
18     13
16     12
19     11
17      9
24      6
33      5
25      4
26      4
31      3
32      3
36      2
27      2
29      2
23      2
34      1
37      1
30      1
40      1
Name: YearsAtCompany, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== YearsInCurrentRole ========

2     372
0     244
7     222
3     135
4     104
8      89
9      67
1      57
6      37
5      36
10     29
11     22
13     14
14     11
12     10
15      8
16      7
```

```
17      4
18      2
Name: YearsInCurrentRole, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== YearsSinceLastPromotion ========

0     581
1     357
2     159
7      76
4      61
3      52
5      45
6      32
11     24
8      18
9      17
15     13
12     10
13     10
14      9
10      6
Name: YearsSinceLastPromotion, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
======== YearsWithCurrManager ========

2     344
0     263
7     216
3     142
8     107
4      98
1      76
9      64
5      31
6      29
10     27
11     22
12     18
13     14
17      7
15      5
14      5
16      2
Name: YearsWithCurrManager, dtype: int64
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
```

In [16]:
```
#Create separate dataframe containing attrition employees only
Attrition = df.query("Attrition == 'Yes'")
Attrition
```

Out[16]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	E
0	41	Yes	Travel_Rarely	1102	Sales	1	2	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
14	28	Yes	Travel_Rarely	103	Research & Development	24	3	
21	36	Yes	Travel_Rarely	1218	Sales	9	4	
24	34	Yes	Travel_Rarely	699	Research & Development	6	1	
...	...	...	...	...	...	...	...	
1438	23	Yes	Travel_Frequently	638	Sales	9	3	
1442	29	Yes	Travel_Rarely	1092	Research & Development	1	4	
1444	56	Yes	Travel_Rarely	310	Research & Development	7	2	
1452	50	Yes	Travel_Frequently	878	Sales	1	4	
1461	50	Yes	Travel_Rarely	410	Sales	28	3	

237 rows × 31 columns

In [18]:
```
#Transform non-numeric columns into numerical columns
from sklearn.preprocessing import LabelEncoder

for column in df.columns:
        if df[column].dtype == np.number:
            continue
        df[column] = LabelEncoder().fit_transform(df[column])
```

In [19]:
```
#Show dataframe
df
```

Out[19]:

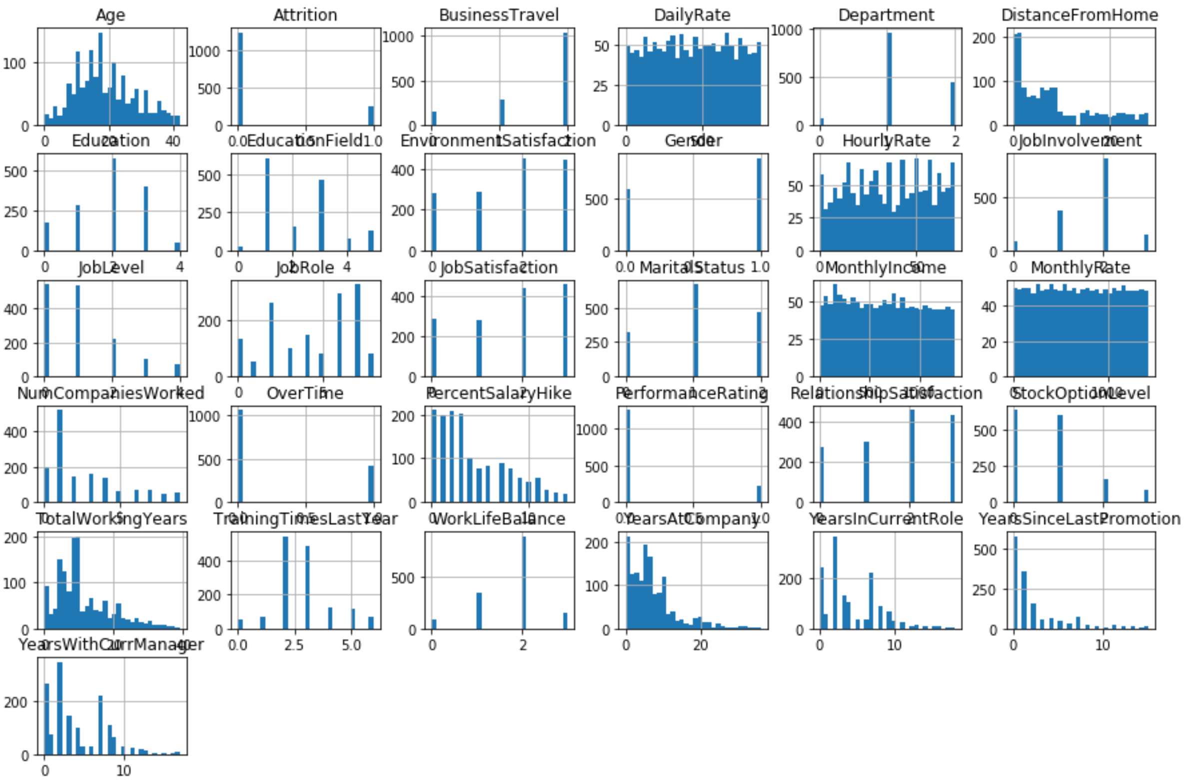
	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Edu
0	23	1	2	624	2	0	1	
1	31	0	1	113	1	7	0	
2	19	1	2	805	1	1	1	
3	15	0	1	820	1	2	3	
4	9	0	2	312	1	1	0	
...	...	...	...	...	...	...	...	
1465	18	0	1	494	1	22	1	
1466	21	0	2	327	1	5	0	
1467	9	0	2	39	1	3	2	
1468	31	0	1	579	2	1	2	
1469	16	0	2	336	1	7	2	

1470 rows × 31 columns

In [20]:

```
#Histogram for all variables
df.hist(bins=30, figsize=(15, 10))
```

```
Out[20]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000027E96885438>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E968AC978>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E968DDF28>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E9691C518>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E9694DAC8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E9698C0B8
         >],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x0000027E969BC668>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E969EDC50>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E969EDC88>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E96A5C7B8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E96A8ED68>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E96ACA358
         >],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x0000027E96AFB908>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E96B2EEB8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E96B6A4A8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E96B9BA58>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E97062048>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E96BE35F8
         >],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x0000027E96C13BA8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E97463198>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E97494748>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E974C3CF8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E975022E8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E97533898
         >],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x0000027E97566E48>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E975A2438>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E975D39E8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E97606F98>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E97641588>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E97673B38
         >],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x0000027E976B1128>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E986B06D8>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E986E5C88>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E98722278>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E98752828>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x0000027E98783DD8
         >]],
               dtype=object)
```



Findings from EDA

When categorical variables are examined, we can see that the majority of employees who left work in the Research & Development department, with the majority being laboratory technicians, sales executives, or research scientists. These employees received excellent performance ratings. It is never good to lose employees with such high levels of performance! Most of them held a Bachelor's degree and were educated in Life Sciences, Medicine, or Marketing. These employees reported high job involvement, high job satisfaction, and a better work-life balance. However, it is concerning that a large number of them reported low satisfaction with the work environment.

Looking at the attrition per age histogram, employees leave less when they get older. Most of the attrition is made up of employees aged 25 to 35. According to the data, the more working years, years at the company, and years in the current role employees have, the less likely they are to leave. In terms of income, most employees who left were among those with a lower monthly income, with monthly income.

Those with a lower percentage salary increase tend to leave more than those with a higher percentage salary increase. Employees who leave tend to be young, with less time working for the company and at the start of their career because most of these employees worked for less than ten years in total.

In [21]:
```
# Splitting Dataset
X = df.drop('Attrition', axis = 1)
y = df.Attrition
```

In [22]:
```
# Splitting data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.3)
print('X train size: ', len(X_train))
print('X test size: ', len(X_test))
print('y train size: ', len(y_train))
print('y test size: ', len(y_test))
```

```
X train size:  1029
X test size:  441
y train size:  1029
y test size:  441
```

The next step is to encode all categorical variables using the Ordinal Encoder and the One Hot Encoder. It is important to remember that the Ordinal Encoder assumes a category ordering. The One Hot Encoder will generate new columns with binary values indicating the presence or absence of each possible value for each categorical attribute in our dataset, where 0 indicates the absence of each value and 1 indicates the presence of each value.

Education and Job Involvement can easily be encoded with the Ordinal Encoder because there is some hierarchy among their values. However, Department, for example, would be better encoded with the One Hot Encoder because no department is lesser or greater than another.

In [23]:
```
# Enoding categorical variables with One Hot Encoder
OHE = OneHotEncoder(handle_unknown = 'ignore', sparse=False)
columns_OHE = ['Department', 'EducationField', 'JobRole', 'MaritalStatus']
X_train_cols = pd.DataFrame(OHE.fit_transform(X_train[columns_OHE]))
X_test_cols = pd.DataFrame(OHE.transform(X_test[columns_OHE]))

# Putting index back
X_train_cols.index = X_train.index
X_test_cols.index = X_test.index

# Removing categorical columns
num_X_train = X_train.drop([col for col in X_train.columns if X_train[col].dty
pe == "object"], axis = 1)
num_X_test = X_test.drop([col for col in X_test.columns if X_test[col].dtype =
= "object"], axis = 1)

# Adding one-hot encoded columns to numerical features
X_train = pd.concat([num_X_train,X_train_cols ],axis = 1)
X_test = pd.concat([num_X_test, X_test_cols], axis = 1)
```

In [24]:
```
X_train
```

Out[24]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationFiel
899	31	2	622	1	3	1	
1374	40	2	323	2	20	2	
937	21	2	192	1	12	3	
369	13	2	190	1	8	3	
863	15	2	33	0	1	2	
...	...	...	...	...	...	...	.
725	17	2	333	1	13	3	
1045	22	2	503	1	1	2	
125	8	2	471	1	5	2	
402	12	2	609	2	11	2	
638	7	2	304	2	3	0	

1029 rows × 51 columns

In [25]:
```
# Rescaling Data
Scaler = MinMaxScaler()
Scaling_Cols = ['TrainingTimesLastYear','YearsAtCompany','TotalWorkingYears',
               'YearsInCurrentRole','YearsSinceLastPromotion','YearsWithCurrMa
nager',
               'PercentSalaryHike','Age','DailyRate','DistanceFromHome','Hourl
yRate',
               'MonthlyIncome','MonthlyRate','NumCompaniesWorked']
X_train[Scaling_Cols] = Scaler.fit_transform(X_train[Scaling_Cols])
X_test[Scaling_Cols] = Scaler.transform(X_test[Scaling_Cols])
X_train
```

Out[25]:

	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educatio
899	0.738095	2	0.704417	1	0.107143	1	
1374	0.952381	2	0.365798	2	0.714286	2	
937	0.500000	2	0.217441	1	0.428571	3	
369	0.309524	2	0.215176	1	0.285714	3	
863	0.357143	2	0.037373	0	0.035714	2	
...	...	...	...	...	...	...	
725	0.404762	2	0.377123	1	0.464286	3	
1045	0.523810	2	0.569649	1	0.035714	2	
125	0.190476	2	0.533409	1	0.178571	2	
402	0.285714	2	0.689694	2	0.392857	2	
638	0.166667	2	0.344281	2	0.107143	0	

1029 rows × 51 columns

In [27]:
```
# Importing Models
from sklearn.ensemble import AdaBoostClassifier, RandomForestClassifier, Gradi
entBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
```

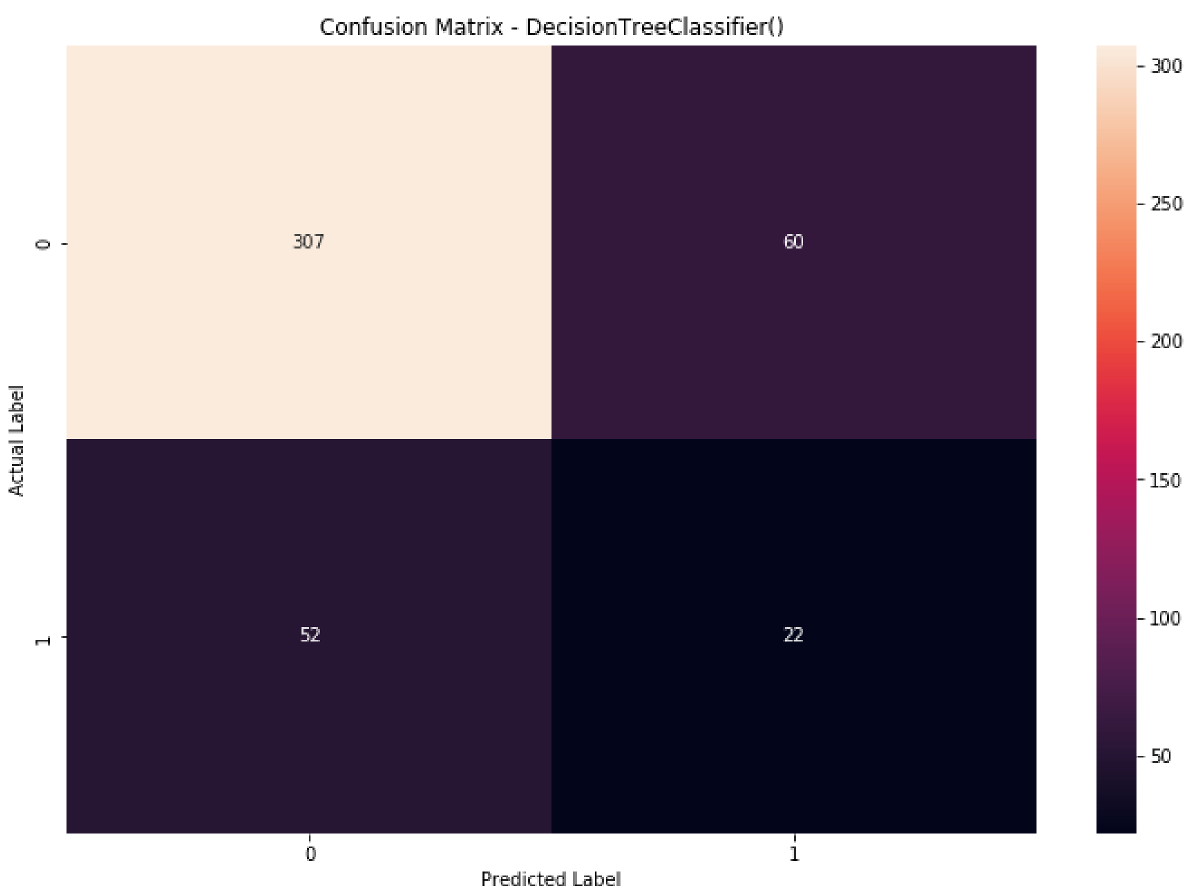
In [28]:
```
randomforest = RandomForestClassifier()
gradientboosting = GradientBoostingClassifier()
decisiontree = DecisionTreeClassifier()
```

In [29]:
```
# Creating a function for predictions
def predict(model):
    model.fit(X_train, y_train)
    y_predict = model.predict(X_test)
    print('Accuracy: %.2f%%' % (accuracy_score(y_test, y_predict) * 100 ))
    print('Precision: %.2f%%' % (precision_score(y_test, y_predict) * 100))
    print('Recall: %.2f%%' % (recall_score(y_test, y_predict) * 100))
    print('F1_Score: %.2f%%' % (f1_score(y_test, y_predict) * 100))

    confusion_matrix_model = confusion_matrix(y_test, y_predict)
    plt.figure(figsize=(12,8))
    ax = plt.subplot()
    sns.heatmap(confusion_matrix_model, annot=True, fmt='g', ax = ax)
    ax.set_xlabel('Predicted Label')
    ax.set_ylabel('Actual Label')
    ax.set_title(f'Confusion Matrix - {model}')
    ax.xaxis.set_ticklabels(['0','1'])
    ax.yaxis.set_ticklabels(['0','1'])
```

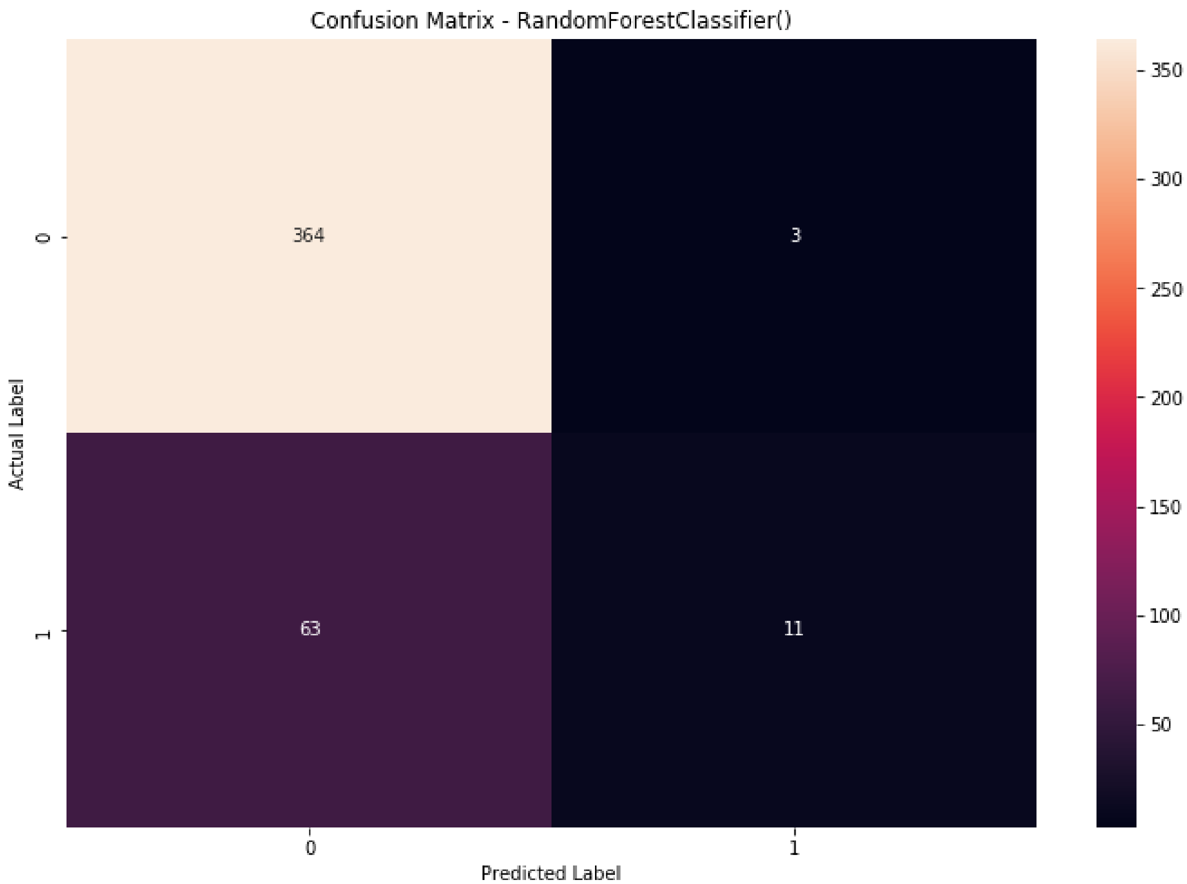
In [30]:
```
predict(decisiontree)
```

```
Accuracy: 74.60%
Precision: 26.83%
Recall: 29.73%
F1_Score: 28.21%
```



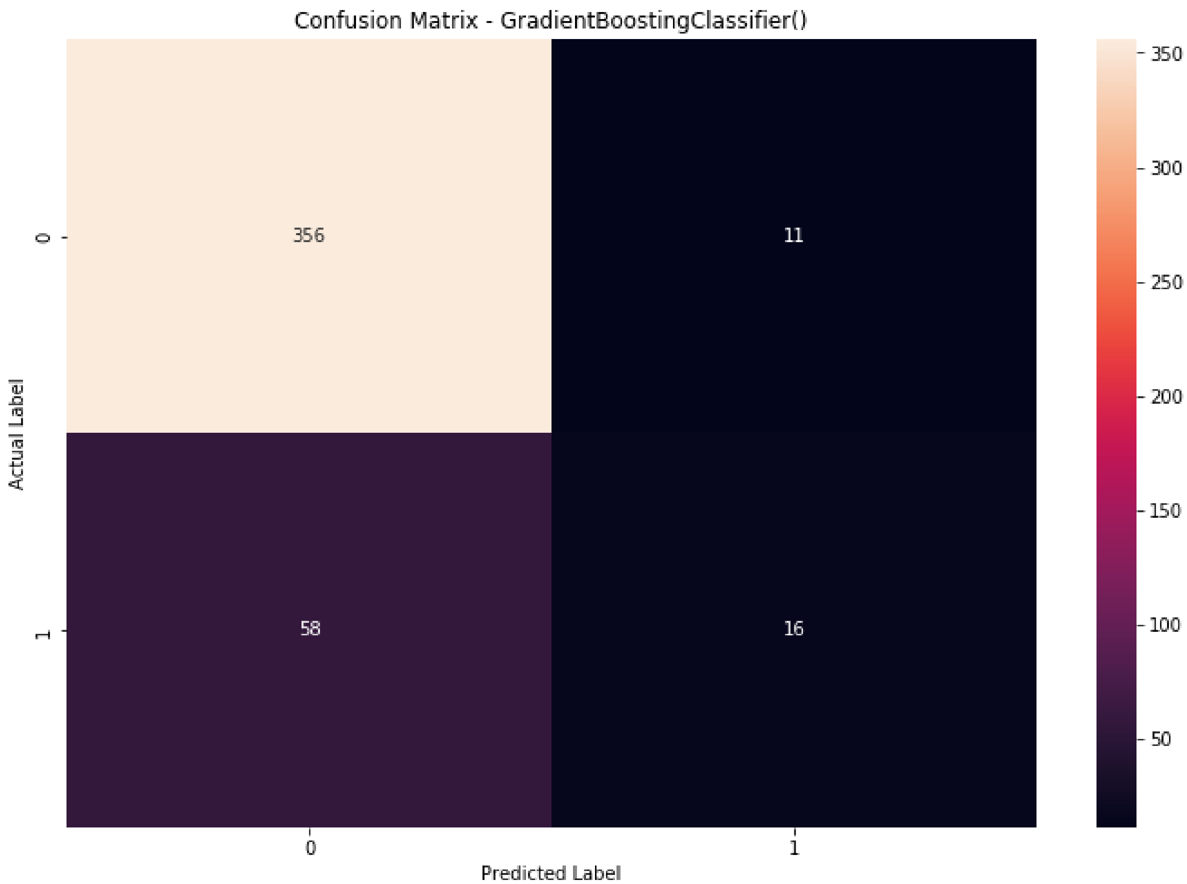
In [31]:
```
predict(randomforest)
```

```
Accuracy: 85.03%
Precision: 78.57%
Recall: 14.86%
F1_Score: 25.00%
```



In [32]:
```
predict(gradientboosting)
```

```
Accuracy: 84.35%
Precision: 59.26%
Recall: 21.62%
F1_Score: 31.68%
```



References

In [ ]:
```
#https://medium.com/analytics-vidhya/predict-employee-attrition-a34e2c5a972d
#https://towardsdatascience.com/people-analytics-with-attrition-predictions-12
adcce9573f
#https://www.analyticsvidhya.com/blog/2021/11/employee-attrition-prediction-a-
comprehensive-guide/#h2_7
```