In [78]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [79]:
```
data =pd.read_csv("ibm-hr-analytics-employee-attrition-performance.zip")

data
```

Out[79]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipSatisfaction	StandardHours	StockOptionLevel	TotalWorkingYears	TrainingTi
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	1	80	0	8	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	4	80	1	10	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	2	80	0	7	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	3	80	0	8	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	4	80	1	6	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	...	3	80	1	17	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	...	1	80	1	9	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	...	2	80	1	6	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	...	4	80	0	17	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	...	1	80	0	6	

1470 rows × 35 columns

In [80]:
```
data.head()
```

Out[80]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipSatisfaction	StandardHours	StockOptionLevel	TotalWorkingYears	TrainingTime
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	1	80	0	8	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	4	80	1	10	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	2	80	0	7	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	3	80	0	8	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	4	80	1	6	

5 rows × 35 columns

In [81]:
```
data.tail()
```

Out[81]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipSatisfaction	StandardHours	StockOptionLevel	TotalWorkingYears	TrainingTi
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	...	3	80	1	17	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	...	1	80	1	9	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	...	2	80	1	6	
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	...	4	80	0	17	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	...	1	80	0	6	

5 rows × 35 columns

In [82]:
```
data.isnull().sum()
```

Out[82]:
```
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 [83]:
```
data.columns
```

Out[83]:
```
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')
```

In [84]:
```
data.info()
```

Out[84]:
```
<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 [85]:
```
data.shape
```

Out[85]:
```
(1470, 35)
```

In [86]:
```
data.describe()
```

Out[86]:

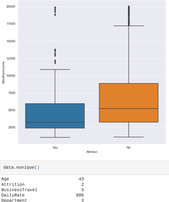
	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	...	RelationshipSatisfaction	StandardHours	StockOptionLevel	Tot
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000		1470.000000	1470.0	1470.000000	
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156	2.729932	2.063946		2.712245	80.0	0.793878	
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428	0.711561	1.106940		1.081209	0.0	0.852077	
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000	1.000000	1.000000		1.000000	80.0	0.000000	
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000	2.000000	1.000000		2.000000	80.0	0.000000	
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000	3.000000	2.000000		3.000000	80.0	1.000000	
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000	3.000000	3.000000		4.000000	80.0	1.000000	
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000	4.000000	5.000000		4.000000	80.0	3.000000	

8 rows × 26 columns

In [87]:
```
sns.set_style('darkgrid')
sns.countplot(x = 'Attrition', data = data)
plt.show()
```

In [88]:
```
sns.lmplot(x = 'Age', y = 'DailyRate', hue = 'Attrition', data = data)
plt.show()
```

In [89]:
```
plt.figure(figsize =(8,6))
sns.boxplot(y = 'MonthlyIncome', x ='Attrition', data = data)
plt.show()
```



In [90]:
```
data.nunique()
```

Out[90]:
```
Age                          43
Attrition                     2
BusinessTravel                3
DailyRate                   886
Department                    3
DistanceFromHome             29
Education                     5
EducationField                6
EmployeeCount                 1
EmployeeNumber             1470
EnvironmentSatisfaction       4
Gender                        2
HourlyRate                   71
JobInvolvement                4
JobLevel                      5
JobRole                       9
JobSatisfaction               4
MaritalStatus                 3
MonthlyIncome              1349
MonthlyRate                1427
NumCompaniesWorked           10
Over18                        1
OverTime                      2
PercentSalaryHike            15
PerformanceRating             2
RelationshipSatisfaction      4
StandardHours                 1
StockOptionLevel              4
TotalWorkingYears            40
TrainingTimesLastYear         7
WorkLifeBalance               4
YearsAtCompany               37
YearsInCurrentRole           19
YearsSinceLastPromotion      16
YearsWithCurrManager         18
dtype: int64
```

In [91]:
```
sns.displot(x = 'Age', hue = 'Attrition', data = data,kde = True)
plt.show()
```

In [92]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.import preprocessing
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
```

In [93]:
```
data.drop(['EmployeeCount'],axis = 1, inplace = True)
data.drop(['StandardHours'],axis = 1,inplace = True)
data.drop(['Over18'],axis = 1,inplace = True)
data.drop(['EmployeeNumber'],axis = 1,inplace = True)
```

In [94]:
```
data.head()
```

Out[94]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction	Gender	...	PerformanceRating	RelationshipSatisfaction	StockOptionLevel	TotalWorkingYears	TrainingTi
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	2	Female	...	3	1	0	8	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	3	Male	...	4	4	1	10	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	4	Male	...	3	2	0	7	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	4	Female	...	3	3	0	8	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	Male	...	3	4	1	6	

5 rows × 31 columns

In [95]:
```
# categorical features

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

In [97]:
```
# Reordering the dataframe, making attrition the first column

data.insert(0, 'Attrition', data.pop('Attrition'))
```

In [98]:
```
# Splitting the data
X = data.iloc[:, 1:data.shape[1]].values
Y = data.iloc[:, 0].values
```

In [99]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=0)
```

In [100]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [101]:
```
# Use random forest classifier
forest = RandomForestClassifier(n_estimators=10, criterion='entropy', random_state=0)
forest.fit(X_train, Y_train)
```

Out[101]:
```
RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)
```

In [102]:
```
# Get accuracy on training data
forest.score(X_train, Y_train)
```

Out[102]:
```
0.9854227405247813
```

In [103]:
```
from sklearn.metrics import confusion_matrix
```

In [104]:
```
# Get accuracy score for the model on the test data
cm = confusion_matrix(Y_test, forest.predict(X_test))

tn = cm[0][0]
fp = cm[0][1]
fn = cm[1][0]
tp = cm[1][1]
accuracy = (tp + tn) / (tn + tp + fn + fp)
print(cm)
print('Accuracy: {:.2f}%'.format(accuracy * 100))
```

```
[[367  4]
 [ 57 13]]
Accuracy: 86.17%
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: