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

In [2]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.datasets import load_iris
```

In [3]:
```
data = pd.read_csv("C:\Users\ASUS\Downloads\HR-Employee-Attrition.csv")
data
```

Out[3]:

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

1470 rows × 35 columns

In [4]:
```
data.iloc[:6,1]
```

Out[4]:
```
0    Female
1      Male
2      Male
3    Female
4      Male
5      Male
6    Female
       Male
Name: Gender, dtype: object
```

In [5]:
```
data.info()
```

Out[5]:
```
<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  TotalWorkingHours         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
dtype: int64(26), object(9)
memory usage: 402.1+ KB
```

In [6]:
```
data.describe()
```

Out[6]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	...	RelationshipSatisfaction	StandardHours	StockOptionLevel	TotalWorkingHours	TrainingTimesLastYear	WorkLifeBalance	Ye
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	1470.000000	1470.000000	1470.000000	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	11.279592	2.799320	2.761224	7.008163
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	7.780782	1.289271	0.706476	6.126525
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	0.000000	0.000000	1.000000	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	8.000000	2.000000	2.000000	3.000000
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	10.000000	3.000000	3.000000	5.000000
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	15.000000	3.000000	3.000000	9.000000
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	40.000000	6.000000	4.000000	40.000000

8 rows × 26 columns

In [7]:
```
data.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
TotalWorkingHours           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

In [8]:
```
data["StandardHours"].value_counts()
```

Out[8]:
```
80    1470
Name: StandardHours, dtype: int64
```

In [9]:
```
data = data.drop(["EmployeeCount", "Over18", "StandardHours", "EmployeeNumber"], axis=1)
```

In [10]:
```
data.info()
```

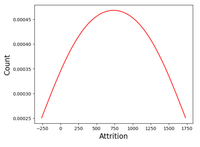
Out[10]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 31 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 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   EnvironmentSatisfaction   1470 non-null   int64
 9   Gender                    1470 non-null   object
 10  HourlyRate                1470 non-null   int64
 11  JobInvolvement            1470 non-null   int64
 12  JobLevel                  1470 non-null   int64
 13  JobRole                   1470 non-null   object
 14  JobSatisfaction           1470 non-null   int64
 15  MaritalStatus             1470 non-null   object
 16  MonthlyIncome             1470 non-null   int64
 17  MonthlyRate               1470 non-null   int64
 18  NumCompaniesWorked        1470 non-null   int64
 19  OverTime                  1470 non-null   object
 20  PercentSalaryHike         1470 non-null   int64
 21  PerformanceRating         1470 non-null   int64
 22  RelationshipSatisfaction  1470 non-null   int64
 23  StockOptionLevel          1470 non-null   int64
 24  TotalWorkingHours         1470 non-null   int64
 25  TrainingTimesLastYear     1470 non-null   int64
 26  WorkLifeBalance           1470 non-null   int64
 27  YearsAtCompany            1470 non-null   int64
 28  YearsInCurrentRole        1470 non-null   int64
 29  YearsSinceLastPromotion   1470 non-null   int64
 30  YearsWithCurrManager      1470 non-null   int64
dtype: int64(23), object(8)
memory usage: 356.1+ KB
```

In [11]:
```
data["Age"].value_counts().sort_index(ascending = True)
```

Out[11]:
```
18    8
19    9
20   11
21   21
22   19
23   19
24   29
25   29
26   31
27   43
28   48
29   42
30   60
31   49
32   52
33   50
34   58
35   76
36   69
37   70
38   56
39   62
40   57
41   42
42   46
43   32
44   33
45   38
46   33
47   36
48   29
49   30
50   30
51   23
52   19
53   16
54   18
55   22
56   14
57   14
58   14
59   15
60   12
Name: Age, dtype: int64
```

In [12]:
```
[data["Attrition"].value_counts()].plot(kind="bearh", color="red")
plt.ylabel("Attrition", fontsize=15)
plt.xlabel("Attrition",fontsize=15)
plt.ylabel("Count",fontsize=15)
```

Out[12]:
```
No    1233
Yes    237
Name: Attrition, dtype: int64
Text(0, 0.5, 'Count')
```



In [13]:
```
for col in data.describe(include="object").columns:
    print(col)
    print(data[col].unique())
    print("_" * 40)
```

Out[13]:
```
Attrition
['Yes' 'No']
________________________________________
BusinessTravel
['Travel_Rarely' 'Travel_Frequently' 'Non-Travel']
________________________________________
Department
['Sales' 'Research & Development' 'Human Resources']
________________________________________
EducationField
['Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree'
 'Human Resources']
________________________________________
Gender
['Female' 'Male']
________________________________________
JobRole
['Sales Executive' 'Research Scientist' 'Laboratory Technician'
 'Manufacturing Director' 'Healthcare Representative' 'Manager'
 'Sales Representative' 'Research Director' 'Human Resources']
________________________________________
MaritalStatus
['Single' 'Married' 'Divorced']
________________________________________
Over18
['Y']
________________________________________
OverTime
['Yes' 'No']
________________________________________
```

In [14]:
```
cat_cols = data.select_dtypes(["object"]).columns
```

In [15]:
```
cat_cols
```

Out[15]:
```
Index(['Attrition', 'BusinessTravel', 'Department', 'EducationField', 'Gender',
       'JobRole', 'MaritalStatus', 'Over18', 'OverTime'],
      dtype='object')
```

In [16]:
```
import matplotlib.pyplot as plt

plotnumber = 1
plt.figure(figsize=(50, 24), facecolor="white")

for col in cat_cols:
    if plotnumber <= 14:  # Adjusted the condition to match the subplot configuration
        ax = plt.subplot(3, 4, plotnumber)
        sns.countplot(data=data, x=data[col], palette="hls")
        plt.title(col + " Distribution")
        plotnumber += 1

plt.show()
```

In [17]:
```
cat_cols_2 = data.drop("Attrition", axis=1).select_dtypes(["object"]).columns
```

In [18]:
```
cat_cols_2
```

Out[18]:
```
Index(['BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRole',
       'MaritalStatus', 'Over18', 'OverTime'],
      dtype='object')
```

In [19]:
```
import matplotlib.pyplot as plt

plotnumber = 1
plt.figure(figsize=(50, 24), facecolor="white")

for col in cat_cols_2:
    if plotnumber <= 14:  # Adjusted the condition to match the subplot configuration
        ax = plt.subplot(3, 4, plotnumber)
        sns.countplot(data=data, x=data[col], hue="Attrition")
        plt.title("Attrition Distribution by " + col)
        ax = plt.gca()
        total_height = len(data["Attrition"])
        for p in ax.patches:
            percentage = '{:.1f}%'.format(100*p.get_height()/total_height)
            x = p.get_x() + p.get_width() / 2
            y = p.get_height()
            ax.annotate(percentage, (x,y), ha='center')
        plotnumber += 1

plt.show()
```