EmployeeAttrition

March 10, 2025

1 Predicting Employee Attrition

dataset: Kaggle IBM HR Analytics Employee Attrition & Performance

1.1 Read csv file

[1]:
```
# Import pandas library
import pandas as pd
```

[2]:
```
# read csv file into pandas dataframe
df = pd.read_csv("/content/WA_Fn-UseC_-HR-Employee-Attrition.csv")
```

[3]:
```
# Display dataframe. 1470 rows x 35 columns
df
```

[3]:

	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	
…	…	…	…	…	…	
1465	36	No	Travel_Frequently	884	Research & Development	
1466	39	No	Travel_Rarely	613	Research & Development	
1467	27	No	Travel_Rarely	155	Research & Development	
1468	49	No	Travel_Frequently	1023	Sales	
1469	34	No	Travel_Rarely	628	Research & Development	

	DistanceFromHome	Education	EducationField	EmployeeCount	\
0	1	2	Life Sciences	1	
1	8	1	Life Sciences	1	
2	2	2	Other	1	
3	3	4	Life Sciences	1	
4	2	1	Medical	1	
…	…	…	…	…	
1465	23	2	Medical	1	
1466	6	1	Medical	1	
1467	4	3	Life Sciences	1	

```
1468                2          3        Medical              1
1469                8          3        Medical              1

      EmployeeNumber  …  RelationshipSatisfaction StandardHours  \
0                  1  …                         1            80
1                  2  …                         4            80
2                  4  …                         2            80
3                  5  …                         3            80
4                  7  …                         4            80
…                …  …                         …           …
1465            2061  …                         3            80
1466            2062  …                         1            80
1467            2064  …                         2            80
1468            2065  …                         4            80
1469            2068  …                         1            80

      StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
0                    0                  8                      0
1                    1                 10                      3
2                    0                  7                      3
3                    0                  8                      3
4                    1                  6                      3
…                  …                  …                    …
1465                 1                 17                      3
1466                 1                  9                      5
1467                 1                  6                      0
1468                 0                 17                      3
1469                 0                  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
…                …               …                …
1465               3               5                  2
1466               3               7                  7
1467               3               6                  2
1468               2               9                  6
1469               4               4                  3

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

```
4                           2                    2
…                         …                    …
1465                        0                    3
1466                        1                    7
1467                        0                    3
1468                        0                    8
1469                        1                    2

[1470 rows x 35 columns]
```

[4]:
```
# change number of columns displayed default
pd.options.display.max_columns = 500
```

[5]:
```
# Display dataframe
df
```

[5]:
```
      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
…    …       …                 …        …                       …
1465   36        No  Travel_Frequently        884  Research & Development
1466   39        No      Travel_Rarely        613  Research & Development
1467   27        No      Travel_Rarely        155  Research & Development
1468   49        No  Travel_Frequently       1023                   Sales
1469   34        No      Travel_Rarely        628  Research & Development

      DistanceFromHome  Education EducationField  EmployeeCount  \
0                    1          2  Life Sciences              1
1                    8          1  Life Sciences              1
2                    2          2          Other              1
3                    3          4  Life Sciences              1
4                    2          1        Medical              1
…                  …        …              …            …
1465                23          2        Medical              1
1466                 6          1        Medical              1
1467                 4          3  Life Sciences              1
1468                 2          3        Medical              1
1469                 8          3        Medical              1

      EmployeeNumber  EnvironmentSatisfaction  Gender  HourlyRate  \
0                  1                        2  Female          94
1                  2                        3    Male          61
2                  4                        4    Male          92
3                  5                        4  Female          56
```

```
4                    7                         1    Male          40
…                  …                       …     …         …
1465              2061                         3    Male          41
1466              2062                         4    Male          42
1467              2064                         2    Male          87
1468              2065                         4    Male          63
1469              2068                         2    Male          82

      JobInvolvement  JobLevel                    JobRole  JobSatisfaction  \
0                  3         2            Sales Executive                4
1                  2         2         Research Scientist                2
2                  2         1      Laboratory Technician                3
3                  3         1         Research Scientist                3
4                  3         1      Laboratory Technician                2
…                …       …                          …              …
1465               4         2      Laboratory Technician                4
1466               2         3  Healthcare Representative                1
1467               4         2     Manufacturing Director                2
1468               2         2            Sales Executive                2
1469               4         2      Laboratory Technician                3

     MaritalStatus  MonthlyIncome  MonthlyRate  NumCompaniesWorked Over18  \
0           Single           5993        19479                   8      Y
1          Married           5130        24907                   1      Y
2           Single           2090         2396                   6      Y
3          Married           2909        23159                   1      Y
4          Married           3468        16632                   9      Y
…            …             …           …                  …    …
1465       Married           2571        12290                   4      Y
1466       Married           9991        21457                   4      Y
1467       Married           6142         5174                   1      Y
1468       Married           5390        13243                   2      Y
1469       Married           4404        10228                   2      Y

     OverTime  PercentSalaryHike  PerformanceRating  RelationshipSatisfaction  \
0         Yes                 11                  3                         1
1          No                 23                  4                         4
2         Yes                 15                  3                         2
3         Yes                 11                  3                         3
4          No                 12                  3                         4
…       …                 …                 …                       …
1465       No                 17                  3                         3
1466       No                 15                  3                         1
1467      Yes                 20                  4                         2
1468       No                 14                  3                         4
1469       No                 12                  3                         1
```

```
      StandardHours  StockOptionLevel  TotalWorkingYears  \
0                80                 0                  8
1                80                 1                 10
2                80                 0                  7
3                80                 0                  8
4                80                 1                  6
…               …                  …                  …
1465             80                 1                 17
1466             80                 1                  9
1467             80                 1                  6
1468             80                 0                 17
1469             80                 0                  6

      TrainingTimesLastYear  WorkLifeBalance  YearsAtCompany  \
0                         0                1               6
1                         3                3              10
2                         3                3               0
3                         3                3               8
4                         3                3               2
…                        …                …               …
1465                      3                3               5
1466                      5                3               7
1467                      0                3               6
1468                      3                2               9
1469                      3                4               4

      YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                      4                        0                     5
1                      7                        1                     7
2                      0                        0                     0
3                      7                        3                     0
4                      2                        2                     2
…                     …                        …                     …
1465                   2                        0                     3
1466                   7                        1                     7
1467                   2                        0                     3
1468                   6                        0                     8
1469                   3                        1                     2

[1470 rows x 35 columns]
```

1.2 Exploratory Data Analysis

[6]:
```
# Display columns and types
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
```

[7]:
```
# Gender Value Counts
df.Gender.value_counts()
```

[7]:
```
Gender
Male      882
Female    588
Name: count, dtype: int64
```

[8]:
```
# Attrition Value Counts
df.Attrition.value_counts()
```

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

[9]:
```
# OverTime Value Counts
df.OverTime.value_counts()
```

[9]:
```
OverTime
No     1054
Yes     416
Name: count, dtype: int64
```

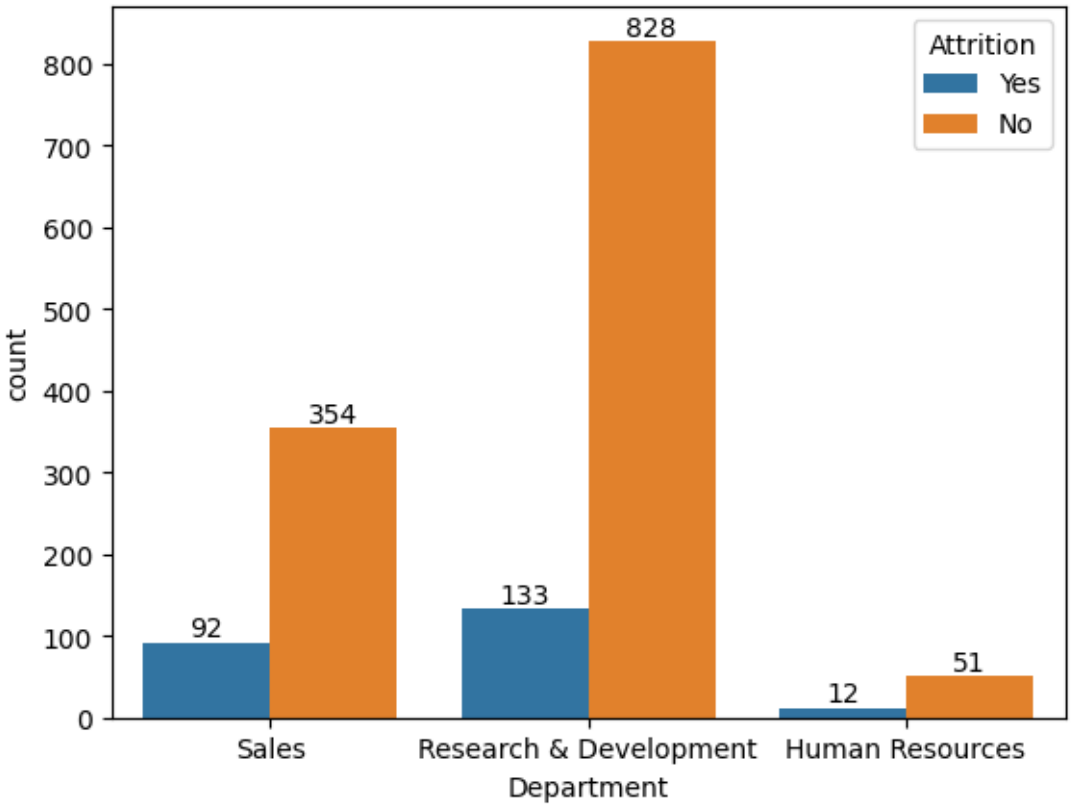
[10]:
```
# Over18 Value Counts
df.Over18.value_counts()
```

[10]:
```
Over18
Y    1470
Name: count, dtype: int64
```

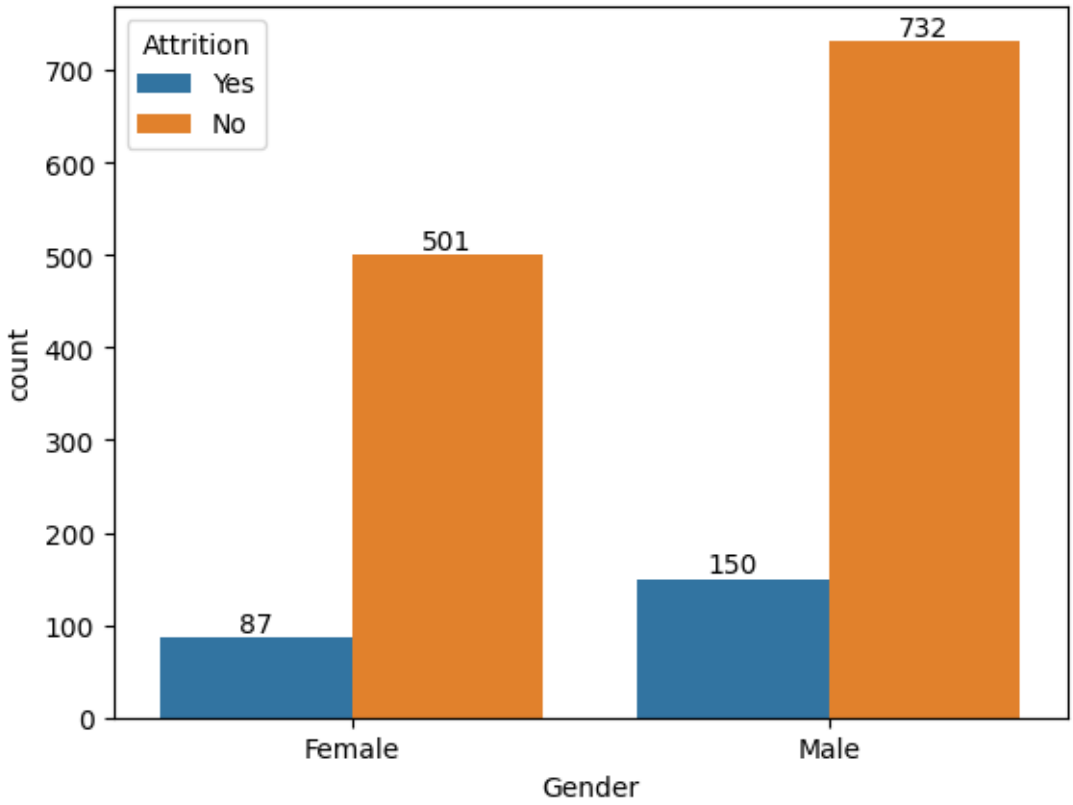
1.3 Data Visualization

[11]:
```
# Load matplotlib
# Histogram
import matplotlib.pyplot as plt

df.hist(figsize=(20,15))
plt.tight_layout()
plt.show()
```

[12]:
```
# Load seaborn
# Attrition By Department
import seaborn as sns
import matplotlib.pyplot as plt
ax=sns.countplot(df, x="Department", hue="Attrition")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.show()
```

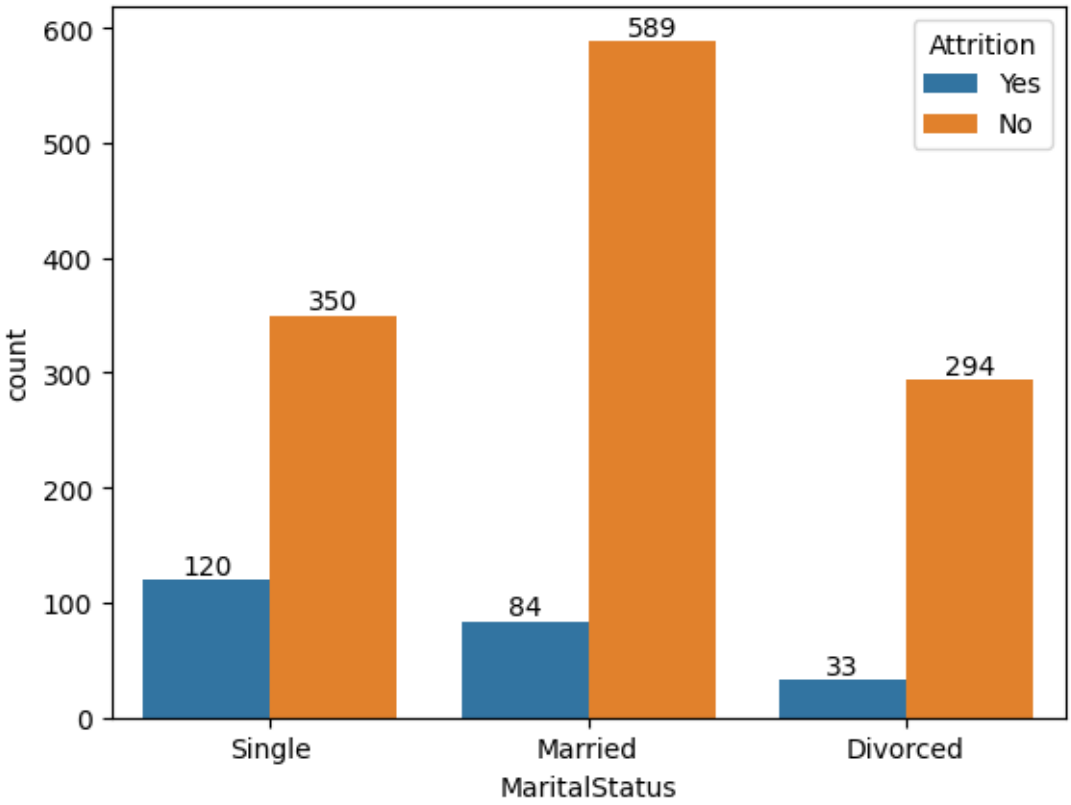


[13]:
```
# Attrition By Gender
ax=sns.countplot(df, x="Gender", hue="Attrition")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.show()
```



Attrition for women is 14.8%. Attrition for men is 17%.

[14]:
```
# Attrition By Marital Status
ax=sns.countplot(df, x="MaritalStatus", hue="Attrition")
ax.bar_label(ax.containers[0])
ax.bar_label(ax.containers[1])
plt.show()
```



Attrition for single employees is 25.5%. Attrition for married employees is 12.5%. Attrition for divorced employees is 10.1%.

[15]:
```
# Aggregate on Department and Attrition
agg_attrition = df.groupby(['Department','Attrition']).agg(num_attrition =␣
 ↪('Attrition', 'count')).reset_index()
# agg_attrition.Attrition = agg_attrition.Attrition.map({"No": "No Attrition",␣
 ↪"Yes": "Attrition",})
agg_attrition
```

[15]:

	Department	Attrition	num_attrition
0	Human Resources	No	51
1	Human Resources	Yes	12
2	Research & Development	No	828
3	Research & Development	Yes	133
4	Sales	No	354
5	Sales	Yes	92

[16]:
```
# Modify Attrition values
agg_attrition.Attrition = agg_attrition.Attrition.map({"No": "No Attrition",␣
 ↪"Yes": "Attrition",})
```

```
agg_attrition
```

[16]:
```
               Department     Attrition  num_attrition
0         Human Resources  No Attrition             51
1         Human Resources     Attrition             12
2  Research & Development  No Attrition            828
3  Research & Development     Attrition            133
4                   Sales  No Attrition            354
5                   Sales     Attrition             92
```

[17]:
```
# Load Plotly library
import plotly.express as px

# Pie Chart to show Sales Department Attrition
fig = px.pie(agg_attrition[agg_attrition['Department']=='Sales'],␣
 ↪values='num_attrition', names='Attrition',
             title='Sales Attrition',color_discrete_sequence=px.colors.
 ↪sequential.Blugrn,width=350,height=300)
fig.update_layout(paper_bgcolor="tan",
font_color = "black",
title_font_color="black",
legend_title_font_color="black")
fig.show()
```

[18]:
```
# Pie Chart to show R & D Attrition
fig = px.pie(agg_attrition[agg_attrition['Department']=='Research &␣
 ↪Development'], values='num_attrition', names='Attrition',
             title='Research & Development␣
 ↪Attrition',color_discrete_sequence=px.colors.sequential.
 ↪Blugrn,width=350,height=300)
fig.update_layout(paper_bgcolor="tan",
font_color = "black",
title_font_color="black",
legend_title_font_color="black")
fig.show()
```

[19]:
```
# Pie Chart to show HR Attrition
fig = px.pie(agg_attrition[agg_attrition['Department']=='Human Resources'],␣
 ↪values='num_attrition', names='Attrition',
             title='Human Resources Attrition',color_discrete_sequence=px.
 ↪colors.sequential.Blugrn,width=350,height=300)
fig.update_layout(paper_bgcolor="tan",
font_color = "black",
title_font_color="black",
legend_title_font_color="black")
fig.show()
```

1.4 Feature Engineering

[20]:
```
# columns before changing values to 1 and 0.
df[['Attrition', 'Gender', 'OverTime', 'Over18']]
```

[20]:
```
     Attrition  Gender OverTime Over18
0          Yes  Female      Yes      Y
1           No    Male       No      Y
2          Yes    Male      Yes      Y
3           No  Female      Yes      Y
4           No    Male       No      Y
…          …      …       …     …
1465        No    Male       No      Y
1466        No    Male       No      Y
1467        No    Male      Yes      Y
1468        No    Male       No      Y
1469        No    Male       No      Y

[1470 rows x 4 columns]
```

[21]:
```
# Binary: Attrition, Gender, Overtime, Over18
# Change values to 1 and 0

df['Attrition'] = df['Attrition'].apply(lambda x: 1 if x== 'Yes' else 0)
df['Gender']    = df['Gender'].apply(lambda x: 1 if x== 'Male' else 0)
df['OverTime']  = df['OverTime'].apply(lambda x: 1 if x== 'Yes' else 0)
df['Over18']    = df['Over18'].apply(lambda x: 1 if x== 'Y' else 0)
```

[22]:
```
# display columns post-replacement
df[['Attrition', 'Gender', 'OverTime', 'Over18']]
```

[22]:
```
      Attrition  Gender  OverTime  Over18
0             1       0         1       1
1             0       1         0       1
2             1       1         1       1
3             0       0         1       1
4             0       1         0       1
…          …     …        …     …
1465          0       1         0       1
1466          0       1         0       1
1467          0       1         1       1
1468          0       1         0       1
1469          0       1         0       1

[1470 rows x 4 columns]
```

[23]:
```
df.BusinessTravel.value_counts()
```

[23]:
```
BusinessTravel
Travel_Rarely        1043
Travel_Frequently     277
Non-Travel            150
Name: count, dtype: int64
```

[24]:
```
df.Department.value_counts()
```

[24]:
```
Department
Research & Development    961
Sales                     446
Human Resources            63
Name: count, dtype: int64
```

[25]:
```
df.JobRole.value_counts()
```

[25]:
```
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: count, dtype: int64
```

[26]:
```
df.MaritalStatus.value_counts()
```

[26]:
```
MaritalStatus
Married     673
Single      470
Divorced    327
Name: count, dtype: int64
```

[27]:
```
# One-hot Encoding: BusinessTravel
df = pd.
 ↪get_dummies(df,columns=['BusinessTravel','Department','JobRole','MaritalStatus',␣
 ↪'EducationField'], prefix=["BT",'Dept','JR','MS','EF'], dtype=int)
```

[28]:
```
df.head(5)
```

[28]:

	Age	Attrition	DailyRate	DistanceFromHome	Education	EmployeeCount	\
0	41	1	1102	1	2	1	
1	49	0	279	8	1	1	
2	37	1	1373	2	2	1	

```
3   33          0       1392                  3          4              1
4   27          0        591                  2          1              1

   EmployeeNumber  EnvironmentSatisfaction  Gender  HourlyRate  \
0               1                        2       0          94
1               2                        3       1          61
2               4                        4       1          92
3               5                        4       0          56
4               7                        1       1          40

   JobInvolvement  JobLevel  JobSatisfaction  MonthlyIncome  MonthlyRate  \
0               3         2                4           5993        19479
1               2         2                2           5130        24907
2               2         1                3           2090         2396
3               3         1                3           2909        23159
4               3         1                2           3468        16632

   NumCompaniesWorked  Over18  OverTime  PercentSalaryHike  PerformanceRating  \
0                   8       1         1                 11                  3
1                   1       1         0                 23                  4
2                   6       1         1                 15                  3
3                   1       1         1                 11                  3
4                   9       1         0                 12                  3

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

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

   YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager  \
0                   4                        0                     5
1                   7                        1                     7
2                   0                        0                     0
3                   7                        3                     0
4                   2                        2                     2

   BT_Non-Travel  BT_Travel_Frequently  BT_Travel_Rarely  \
0              0                     0                 1
```

```
1              0                     1                 0
2              0                     0                 1
3              0                     1                 0
4              0                     0                 1

   Dept_Human Resources  Dept_Research & Development  Dept_Sales  \
0                     0                            0           1
1                     0                            1           0
2                     0                            1           0
3                     0                            1           0
4                     0                            1           0

   JR_Healthcare Representative  JR_Human Resources  JR_Laboratory Technician  \
0                             0                   0                         0
1                             0                   0                         0
2                             0                   0                         1
3                             0                   0                         0
4                             0                   0                         1

   JR_Manager  JR_Manufacturing Director  JR_Research Director  \
0           0                         0                    0
1           0                         0                    0
2           0                         0                    0
3           0                         0                    0
4           0                         0                    0

   JR_Research Scientist  JR_Sales Executive  JR_Sales Representative  \
0                      0                   1                        0
1                      1                   0                        0
2                      0                   0                        0
3                      1                   0                        0
4                      0                   0                        0

   MS_Divorced  MS_Married  MS_Single  EF_Human Resources  EF_Life Sciences  \
0            0           0          1                   0                 1
1            0           1          0                   0                 1
2            0           0          1                   0                 0
3            0           1          0                   0                 1
4            0           1          0                   0                 0

   EF_Marketing  EF_Medical  EF_Other  EF_Technical Degree
0             0           0         0                    0
1             0           0         0                    0
2             0           0         1                    0
3             0           0         0                    0
4             0           1         0                    0
```

[29]:
```
# Rename BT_Non-Travel
df=df.rename(columns={'BT_Non-Travel': 'BT_NonTravel'})
```

[30]:
```
# drop EmployeeCount, Over18, StandardHours, EmployeeNumber
df=df.drop(['EmployeeCount','Over18', 'StandardHours', 'EmployeeNumber'],␣
 ↪axis=1)
```

[31]:
```
# Display list of columns post encoding
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 50 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Age                             1470 non-null   int64
 1   Attrition                       1470 non-null   int64
 2   DailyRate                       1470 non-null   int64
 3   DistanceFromHome                1470 non-null   int64
 4   Education                       1470 non-null   int64
 5   EnvironmentSatisfaction         1470 non-null   int64
 6   Gender                          1470 non-null   int64
 7   HourlyRate                      1470 non-null   int64
 8   JobInvolvement                  1470 non-null   int64
 9   JobLevel                        1470 non-null   int64
 10  JobSatisfaction                 1470 non-null   int64
 11  MonthlyIncome                   1470 non-null   int64
 12  MonthlyRate                     1470 non-null   int64
 13  NumCompaniesWorked              1470 non-null   int64
 14  OverTime                        1470 non-null   int64
 15  PercentSalaryHike               1470 non-null   int64
 16  PerformanceRating               1470 non-null   int64
 17  RelationshipSatisfaction        1470 non-null   int64
 18  StockOptionLevel                1470 non-null   int64
 19  TotalWorkingYears               1470 non-null   int64
 20  TrainingTimesLastYear           1470 non-null   int64
 21  WorkLifeBalance                 1470 non-null   int64
 22  YearsAtCompany                  1470 non-null   int64
 23  YearsInCurrentRole              1470 non-null   int64
 24  YearsSinceLastPromotion         1470 non-null   int64
 25  YearsWithCurrManager            1470 non-null   int64
 26  BT_NonTravel                    1470 non-null   int64
 27  BT_Travel_Frequently            1470 non-null   int64
 28  BT_Travel_Rarely                1470 non-null   int64
 29  Dept_Human Resources            1470 non-null   int64
 30  Dept_Research & Development     1470 non-null   int64
 31  Dept_Sales                      1470 non-null   int64
```

```
 32  JR_Healthcare Representative  1470 non-null   int64
 33  JR_Human Resources            1470 non-null   int64
 34  JR_Laboratory Technician      1470 non-null   int64
 35  JR_Manager                    1470 non-null   int64
 36  JR_Manufacturing Director     1470 non-null   int64
 37  JR_Research Director          1470 non-null   int64
 38  JR_Research Scientist         1470 non-null   int64
 39  JR_Sales Executive            1470 non-null   int64
 40  JR_Sales Representative       1470 non-null   int64
 41  MS_Divorced                   1470 non-null   int64
 42  MS_Married                    1470 non-null   int64
 43  MS_Single                     1470 non-null   int64
 44  EF_Human Resources            1470 non-null   int64
 45  EF_Life Sciences              1470 non-null   int64
 46  EF_Marketing                  1470 non-null   int64
 47  EF_Medical                    1470 non-null   int64
 48  EF_Other                      1470 non-null   int64
 49  EF_Technical Degree           1470 non-null   int64
dtypes: int64(50)
memory usage: 574.3 KB
```

1.5 Split dataset into features and target

[97]:
```
df=df.rename(columns={'JR_Laboratory Technician': 'JR_LaboratoryTechnician',
                      'JR_Sales Representative': 'JR_SalesRepresentative'})
```

[98]:
```
# Split into Features (X) and Target (Y)

X = df.drop('Attrition', axis=1)
y = df['Attrition']
```

1.6 SelectKBest Features

[106]:
```
from sklearn.feature_selection import SelectKBest, chi2, mutual_info_classif,␣
 ↪f_classif

# Add random state to mutual_info_classif to get the same results each run
import functools
mutual_info_classif_rand_state = functools.partial(mutual_info_classif,␣
 ↪random_state=28)

# select top 2 features using mutual_info_classif
reg = SelectKBest(mutual_info_classif_rand_state, k=17).fit(X,y)
#X_new = selector.fit_transform(X, y)
X_transformed = reg.transform(X)
New_X = X[[val for i,val in enumerate(X.columns) if reg.get_support()[i]]]
print(New_X.columns)
```

```
Index(['Age', 'JobLevel', 'JobSatisfaction', 'MonthlyIncome', 'OverTime',
       'StockOptionLevel', 'TotalWorkingYears', 'YearsAtCompany',
       'YearsInCurrentRole', 'YearsWithCurrManager', 'BT_NonTravel',
       'BT_Travel_Frequently', 'JR_LaboratoryTechnician', 'JR_Manager',
       'JR_SalesRepresentative', 'MS_Married', 'MS_Single'],
      dtype='object')
```

Requested SelectKBest to select the 17 best features, reducing the number of features from 49.

1.7 Hyperparameter Tuning

[107]:
```
from sklearn.ensemble import RandomForestClassifier
from scipy.stats import loguniform
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import RandomizedSearchCV

from sklearn.datasets import make_blobs
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import GridSearchCV

# define models and parameters
model = RandomForestClassifier()
n_estimators = [10, 100, 1000]
max_features = ['sqrt', 'log2']

# define grid search
grid = dict(n_estimators=n_estimators,max_features=max_features)
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
grid_search = GridSearchCV(estimator=model, param_grid=grid, n_jobs=-1, cv=cv,␣
 ↪scoring='accuracy',error_score=0)
grid_result = grid_search.fit(New_X, y)

#summarize result
print('Best Score: %s' % grid_result.best_score_)
print('Best Hyperparameters: %s' % grid_result.best_params_)
```

```
Best Score: 0.8562358276643992
Best Hyperparameters: {'max_features': 'sqrt', 'n_estimators': 1000}
```

1.8 Imbalance

[108]:
```
# Address the imbalance between Attrition (237) and No Attrition (1233)
from imblearn.over_sampling import RandomOverSampler

ros = RandomOverSampler()
X_bal,y_bal = ros.fit_resample(New_X,y)
```

[109]:
```
# Print rebalanced X and y dimensions
print(X_bal.shape)
print(y_bal.shape)
```

```
(2466, 17)
(2466,)
```

1.9 Model Training

[110]:
```
from sklearn.model_selection import train_test_split

# split into Train and Test

#X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2,␣
 ↪random_state=42)

X_train, X_test, y_train, y_test = train_test_split(X_bal,y_bal, test_size=0.3,␣
 ↪random_state=42)
```

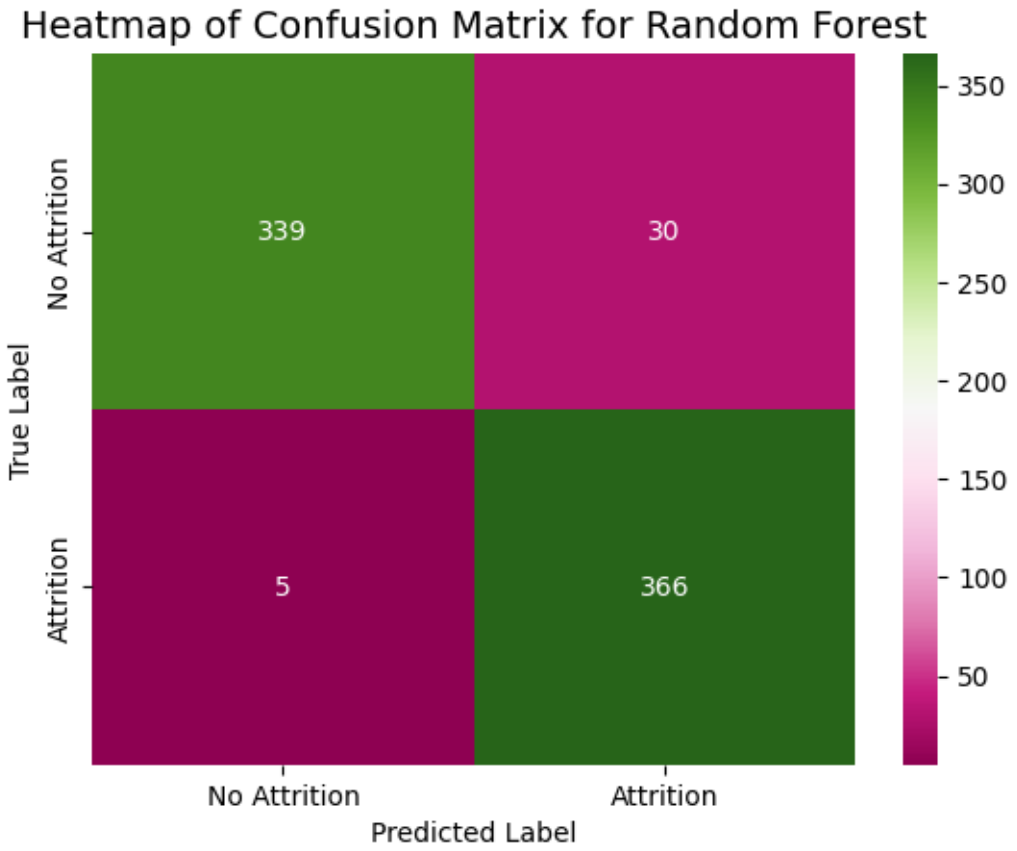
[111]:
```
# model

model = RandomForestClassifier(n_jobs=-1, random_state=42,max_features= 'sqrt',␣
 ↪n_estimators= 1000)

# fit

model.fit(X_train, y_train)

# predict

y_pred = model.predict(X_test)

model.score(X_test, y_test)
```

[111]:
```
0.9527027027027027
```

[112]:
```
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns

print("Classification Report for Random Forest")
print(classification_report(y_test, y_pred))
classes = ['No Attrition', 'Attrition']

sns.heatmap(confusion_matrix(y_test,y_pred), annot=True,␣
 ↪fmt="d",cmap="PiYG",xticklabels=classes, yticklabels=classes)

plt.title('Heatmap of Confusion Matrix for Random Forest', fontsize = 14)
```

```
plt.xlabel('Predicted Label', fontsize = 10) # x-axis label with fontsize 15
plt.ylabel('True Label', fontsize = 10) # y-axis label with fontsize 15
plt.show()
```

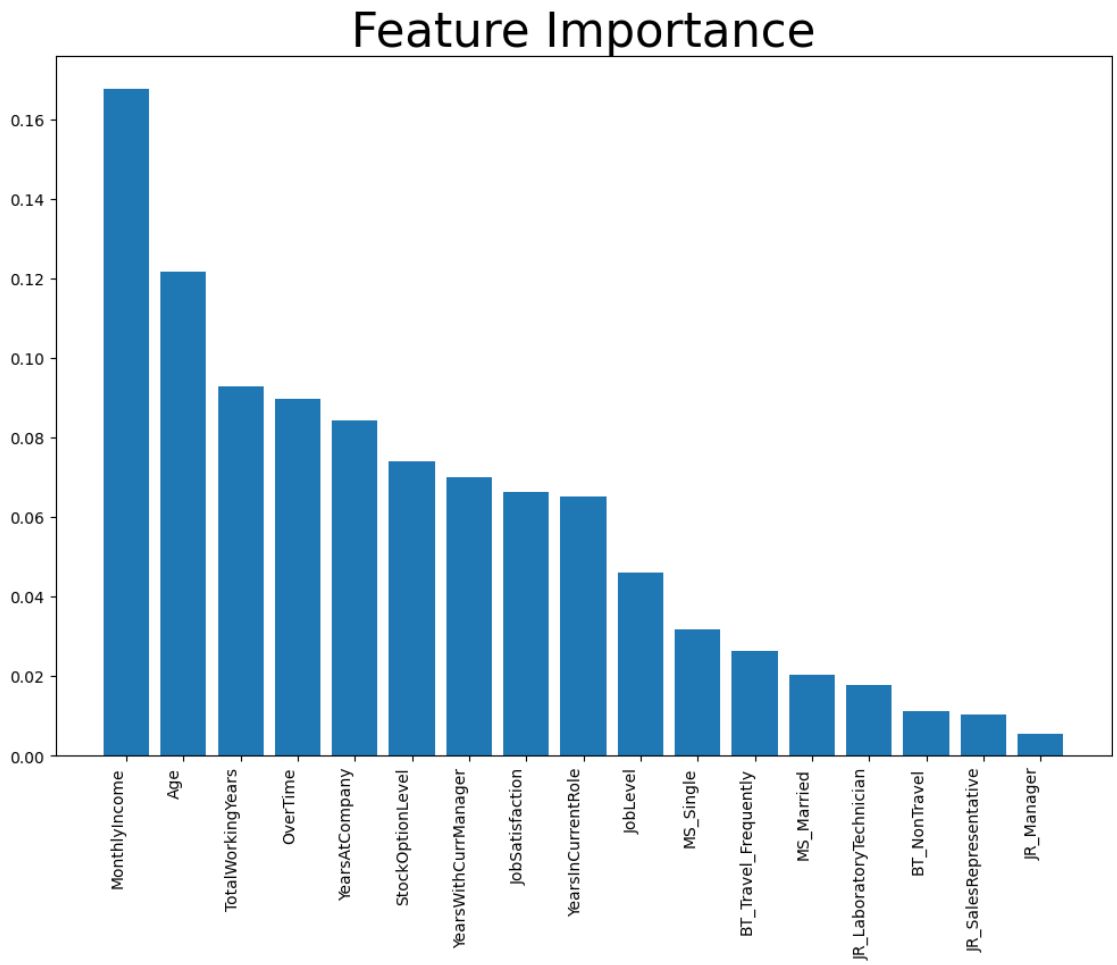
```
Classification Report for Random Forest
              precision    recall  f1-score   support

           0       0.99      0.92      0.95       369
           1       0.92      0.99      0.95       371

    accuracy                           0.95       740
   macro avg       0.95      0.95      0.95       740
weighted avg       0.95      0.95      0.95       740
```



The model achieved 87% accuracy using 24 features and the parameters: max_features='sqrt', n_estimators=1000. Addressed the imbalance between Attrition and No Attrition by using RandomOverSampler and reduced features to 17, increasing the model's accuracy to 95%.

[113]:
```
sorted_importances = dict(sorted(zip(model.feature_names_in_, model.
  ↪feature_importances_), key=lambda x: x[1], reverse=True)
)
```

[114]:
```
plt.figure(figsize=(12,8))
plt.title('Feature Importance', fontsize=30)
plt.bar(sorted_importances.keys(), sorted_importances.values())
plt.xticks(rotation=90, ha='right')
plt.show()
```



1.10 Save Model

[116]:
```
import joblib
# # Use the dump() function to save the model

joblib.dump(model,'HRattrition_model_jl.sav.bz2',compress=('bz2',2))
```

[116]: ['HRattrition_model_jl.sav.bz2']

1.11 Conclusion

- Trained RandomForestClassifier model achieving 95% accuracy by using 17 of 49 features, addressing the imbalance between Attrition and No Attrition, and hypertuning model parameters. The model's accuracy without addressing the imbalance is 87%.
- Attrition for single employees (25.5%) is 2 times married employees (12.5%) and 2.5 times divorced employees (10.1%).
- Attrition for male employees (17%)is slightly higher than female employees (14.8%).
- The Sales department has the highest attrition rate at 20.6%. HR's rate is 19%. R & D's rate is 13.8%.
- The features with the highest importance rating were MonthlyIncome, Age, TotalWorkingYears, YearsAtCompany, and OverTime.
- The 17 features used to train the model are:
  'Age', 'JobLevel', 'JobSatisfaction', 'MonthlyIncome', 'OverTime', 'StockOptionLevel', 'TotalWorkingYears', 'YearsAtCompany', 'YearsInCurrentRole', 'YearsWithCurrManager', 'BT_NonTravel', 'BT_Travel_Frequently', 'JR_LaboratoryTechnician', 'JR_Manager', 'JR_SalesRepresentative', 'MS_Married', 'MS_Single'