-

IBM
MACHINE
LEARNING

IBM EXPLORATORY DATA ANALYSIS FOR EMPLOYEE ATTRITION MODEL

Fatima, Sayeda

8/17/2022

Table of Contents

1) Project Overview

A fundamental issue facing organisations is attraction and retention of best talent. Given the cost of retraining new employees, it is important for a business to prevent loss of good talent. Hence, identification of key factors driving employee churning or turnover is important for the organization's Human Resource (HR) Department.

It is here that machine Learning models can be very useful to gain deeper insight into underlying factors and their relationship in driving employee turnover.

Hence, the main aim of the following machine learning modelling and analysis is to enable the business to:

* To identify different factors predict employee churn

* To gain insight into factors contributing to employee churning

* To enable the business maximize employee attrition

2) About the Dataset

2a) Brief description of chosen data set:

This project uses a hypothetical dataset 'IBM HR Analytics Employee Attrition & Performance' which was downloaded from the following link:

https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset?resource=download

2b) Summary of Data Attributes

The dataset exhibits 1,470 data points (rows) and 35 features (columns) reflecting on employees' background and characteristics and can be downloaded from the following link:

The data also comes with 'Attrition' Column to show current employees and leavers which represents the Class we are trying to predict.

2c) Main Objectives of Analysis

Organizational performance is largely dependent on its employees, their quality and experience. Hence, organizations are continuously faced with the challenge to reduce employee attrition and increase retention. Consequently, this analysis is targeted towards answering the following queries

- What are the various factors contributory to employee attrition?
- Which business units face higher employee attrition rate?

As a consequence, implementation of the model will enable the organization to:

- devise suitable measures to increase employee retention
- to save valuable resources in retraining new employees hired in place of leavers

3) Initial Plan for Data Exploration

3a) Data Exploration

- Data was first loaded into pandas dataframe

Load & Read Dataset

```
# Load the dataset
url = ("C:/Users/fatima.s/Documents/PythonScripts/DATA SCIENCE/IBM Machine Learning Intermediate/MODULE 3 SUPERVISED MACHINE
df = pd.read_csv(url, index_col=False) # keep_default_na = False # na_filter=False,
df.head(100)
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	Relationship
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	

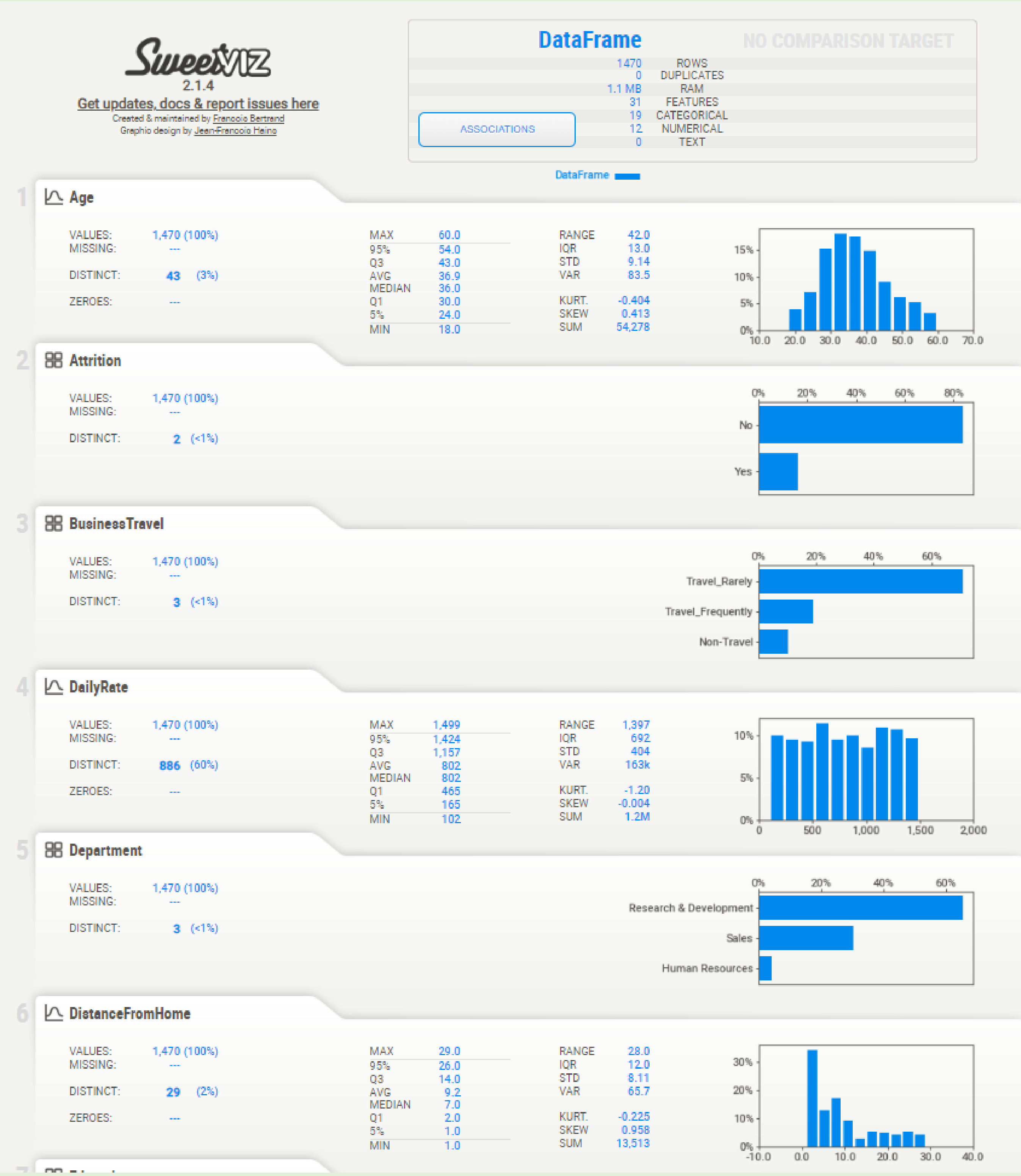
- Column types were explored

Check data set column 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
```

- Automated Exploratory Data Analysis was performed using Sweetviz to check


SweetVIZ
2.1.4
Get updates, docs & report issues here
Created & maintained by Francois Bertrand
Graphic design by Jean-Francois Hains
DataFrame
NO COMPARISON TARGET
1470 ROWS
0 DUPLICATES
1.1 MB RAM
31 FEATURES
19 CATEGORICAL
12 NUMERICAL
0 TEXT
ASSOCIATIONS
DataFrame
1 Age
VALUES: 1,470 (100%)
MISSING: ---
DISTINCT: 43 (3%)
ZEROES: ---
MAX 60.0
95% 54.0
Q3 43.0
AVG 36.9
MEDIAN 36.0
Q1 30.0
5% 24.0
MIN 18.0
RANGE 42.0
IQR 13.0
STD 9.14
VAR 83.5
KURT. -0.404
SKEW 0.413
SUM 54,278
2 Attrition
VALUES: 1,470 (100%)
MISSING: ---
DISTINCT: 2 (<1%)
No
Yes
3 BusinessTravel
VALUES: 1,470 (100%)
MISSING: ---
DISTINCT: 3 (<1%)
Travel_Rarely
Travel_Frequently
Non-Travel
4 DailyRate
VALUES: 1,470 (100%)
MISSING: ---
DISTINCT: 886 (60%)
ZEROES: ---
MAX 1,499
95% 1,424
Q3 1,157
AVG 802
MEDIAN 802
Q1 465
5% 165
MIN 102
RANGE 1,397
IQR 692
STD 404
VAR 163k
KURT. -1.20
SKEW -0.004
SUM 1.2M
5 Department
VALUES: 1,470 (100%)
MISSING: ---
DISTINCT: 3 (<1%)
Research & Development
Sales
Human Resources
6 DistanceFromHome
VALUES: 1,470 (100%)
MISSING: ---
DISTINCT: 29 (2%)
ZEROES: ---
MAX 29.0
95% 26.0
Q3 14.0
AVG 9.2
MEDIAN 7.0
Q1 2.0
5% 1.0
MIN 1.0
RANGE 28.0
IQR 12.0
STD 8.11
VAR 65.7
KURT. -0.225
SKEW 0.958
SUM 13,513

- Descriptive statistics were computed to summarize shape of a dataset's distribution, its dispersion and central tendency

Compute Descriptive Statistics: To summarize shape of a dataset's distribution, its dispersion and central tendency.

```
#To get description of all columns
df.describe(include = 'all')
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber
count	1470.000000	1470	1470	1470.000000	1470	1470.000000	1470.000000	1470	1470.0	1470.000000
unique	NaN	2	3	NaN	3	NaN	NaN	6	NaN	NaN
top	NaN	No	Travel_Rarely	NaN	Research & Development	NaN	NaN	Life Sciences	NaN	NaN
freq	NaN	1233	1043	NaN	961	NaN	NaN	606	NaN	NaN
mean	36.923810	NaN	NaN	802.485714	NaN	9.192517	2.912925	NaN	1.0	1024.865306
std	9.135373	NaN	NaN	403.509100	NaN	8.106864	1.024165	NaN	0.0	602.024335
min	18.000000	NaN	NaN	102.000000	NaN	1.000000	1.000000	NaN	1.0	1.000000
25%	30.000000	NaN	NaN	465.000000	NaN	2.000000	2.000000	NaN	1.0	491.250000
50%	36.000000	NaN	NaN	802.000000	NaN	7.000000	3.000000	NaN	1.0	1020.500000
75%	43.000000	NaN	NaN	1157.000000	NaN	14.000000	4.000000	NaN	1.0	1555.750000
max	60.000000	NaN	NaN	1499.000000	NaN	29.000000	5.000000	NaN	1.0	2068.000000

4) Actions taken for Data Cleansing and Features Engineering

Since the quality of any machine learning model highly depends on quality of data, hence, this stage is not only most important but is also time consuming. Hence, it was conducted in a step-by-step process.

4a) Data Cleansing Actions

- Empty or nearly empty columns were removed using "drop_thresh" to drop columns if 90% of data was empty

Drop Columns if 90% data is empty

```
drop_thresh = df.shape[0]*.10
df = df.loc[:, df.isin([' ','NULL','Nan',0]).mean() < drop_thresh]
df = df.dropna(thresh=drop_thresh, how='all', axis='columns').copy()
df.info()
```

```
print(df.isin([' ','NULL','NaN', 0]).mean())
drop_thresh = .90
df = df.loc[:, df.isin([' ','NULL', 'NaN',0]).mean() < drop_thresh]
print(df.isin([' ','NULL','NaN',0]).mean())
```

```
Age                         0.000000
Attrition                   0.000000
BusinessTravel              0.000000
DailyRate                   0.000000
Department                  0.000000
DistanceFromHome            0.000000
Education                   0.000000
EducationField              0.000000
EmployeeCount               0.000000
EmployeeNumber              0.000000
EnvironmentSatisfaction     0.000000
Gender                      0.000000
HourlyRate                  0.000000
JobInvolvement              0.000000
JobLevel                    0.000000
JobRole                     0.000000
JobSatisfaction             0.000000
MaritalStatus               0.000000
MonthlyIncome               0.000000
MonthlyRate                 0.000000
NumCompaniesWorked          0.134014
Over18                      0.000000
OverTime                    0.000000
```

- Duplicates were dropped using pandas "df.drop_duplicates()" method

Handle Missing Values: Replace remaining ["None","nan", "NaN", ""] values with Zero

```
df = df.replace(["None","nan", "NaN", ""], "0") # Replace all Nan Values with Zero
null = (df.isin(["None","nan", "NaN", ""]).sum()) # Sum as series
null_df=pd.DataFrame({'cols':null.index, 'sum':null.values}).sort_values(by=['sum'],ascending=False)

print(colored("Data has ", 'green', attrs=['bold'])
      +colored((null_df.at[0,'sum']), 'red', attrs=['bold'])
      +colored(" null values.\n ", 'green', attrs=['bold'])
      +colored(null_df.tail(35), 'red', attrs=['bold'])) # print first two rows
```

```
Data has 0 null values.
                          cols  sum
0                          Age    0
26               StandardHours    0
20          NumCompaniesWorked    0
21                      Over18    0
22                    OverTime    0
23           PercentSalaryHike    0
24           PerformanceRating    0
25    RelationshipSatisfaction    0
27            StockOptionLevel    0
18               MonthlyIncome    0
28           TotalWorkingYears    0
29       TrainingTimesLastYear    0
30             WorkLifeBalance    0
31              YearsAtCompany    0
32          YearsInCurrentRole    0
33     YearsSinceLastPromotion    0
19                 MonthlyRate    0
17               MaritalStatus    0
1                    Attrition    0
8                EmployeeCount    0
2               BusinessTravel    0
3                    DailyRate    0
4                   Department    0
5             DistanceFromHome    0
6                    Education    0
7               EducationField    0
9               EmployeeNumber    0
16             JobSatisfaction    0
10     EnvironmentSatisfaction    0
11                      Gender    0
12                  HourlyRate    0
13              JobInvolvement    0
14                    JobLevel    0
15                     JobRole    0
34        YearsWithCurrManager    0
```

- Null values were summed and Data was found to exhibit zero null values. Thus, no filling of null values was required

4b) Features Engineering

In machine learning, feature selection is the method to reduce the number of input variables during developing predictive modelling. This reduction in input variables is necessary not only to minimize computational cost of modeling but also to achieve performance improvement of the model.

Among widely practices feature selection approaches include statistical-based feature selection methods which use statistical measures to evaluate relationship between each input variable and the target variable and then select those exhibiting strongest relationship with the latter. While these methods can be both speedy and effective, however, the ultimate choice of statistical measure is largely dependant on data types of both of these variables.

Irrespective of the statistical measure being employed, two dominant feature selection techniques, that is supervised and unsupervised, exist where the former can be further categorized into wrapper, filter and intrinsic techniques. Filter-based feature selection methods employs statistical measures to evaluate correlation between input and output variables so that those exhibiting highest correlations are selected. Statistical measures employed in filter-based feature selection are normally univariate in nature since they evaluate relationship of single input variables one by one with target variable, disregarding their interaction with each other.

Consequently, adopting filter-based feature selection methods, the employee attrition model approached filter engineering in three steps. Firstly, unique values for all columns were computed after which columns with unique values less than 2 were dropped.

1) Assessing Columns for Feature Selection:

Get unique counts to determine threshold for dropping columns
Drop Columns from dataframe if uniqueness is less than threshold (eg. 2)

```
unique_counts = pd.DataFrame.from_records([(col, df[col].nunique()) for col in df.columns], # get unique counts
                            columns=['Column_Name', 'Unique']).sort_values(by=['Unique'])
print(colored("\nThis can help us determine threshold for which columns to exclude from Features.\n\n ", 'blue', attrs=['bol
      + colored(type(unique_counts), 'green', attrs=['bold'])
      + colored("\n\n", 'green', attrs=['bold'])
      + colored(unique_counts, 'red', attrs=['bold'])
     )

unique = unique_counts[(unique_counts['Unique'] < 2)] #If threshold is lesss than 2 then
drop_unique = (unique['Column_Name'].tolist()) # list of columns to drop

cols_to_exclude = ['EmployeeNumber']
cols_to_exclude = ['EmployeeNumber'] + drop_unique

print(colored("\n\n ", 'blue', attrs=['bold'])
      + colored(type(unique), 'green', attrs=['bold'])
      + colored("\n", 'green', attrs=['bold'])
      + colored(unique, 'red', attrs=['bold'])

      + colored("\n\nList of columns to drop\n", 'blue', attrs=['bold'])
      + colored(cols_to_exclude , 'red', attrs=['bold'])
     )

#Function to Drop Columns & Convert to Categories
for col in df.columns:
    if col in cols_to_exclude:
        df = df.loc[:, ~df.columns.isin(cols_to_exclude)]
df.info()
```

```
This can help us determine threshold for which columns to exclude from Features.

 <class 'pandas.core.frame.DataFrame'>

             Column_Name  Unique
21                Over18       1
26         StandardHours       1
8          EmployeeCount       1
11                Gender       2
1              Attrition       2
24     PerformanceRating       2
22              OverTime       2
17         MaritalStatus       3
```

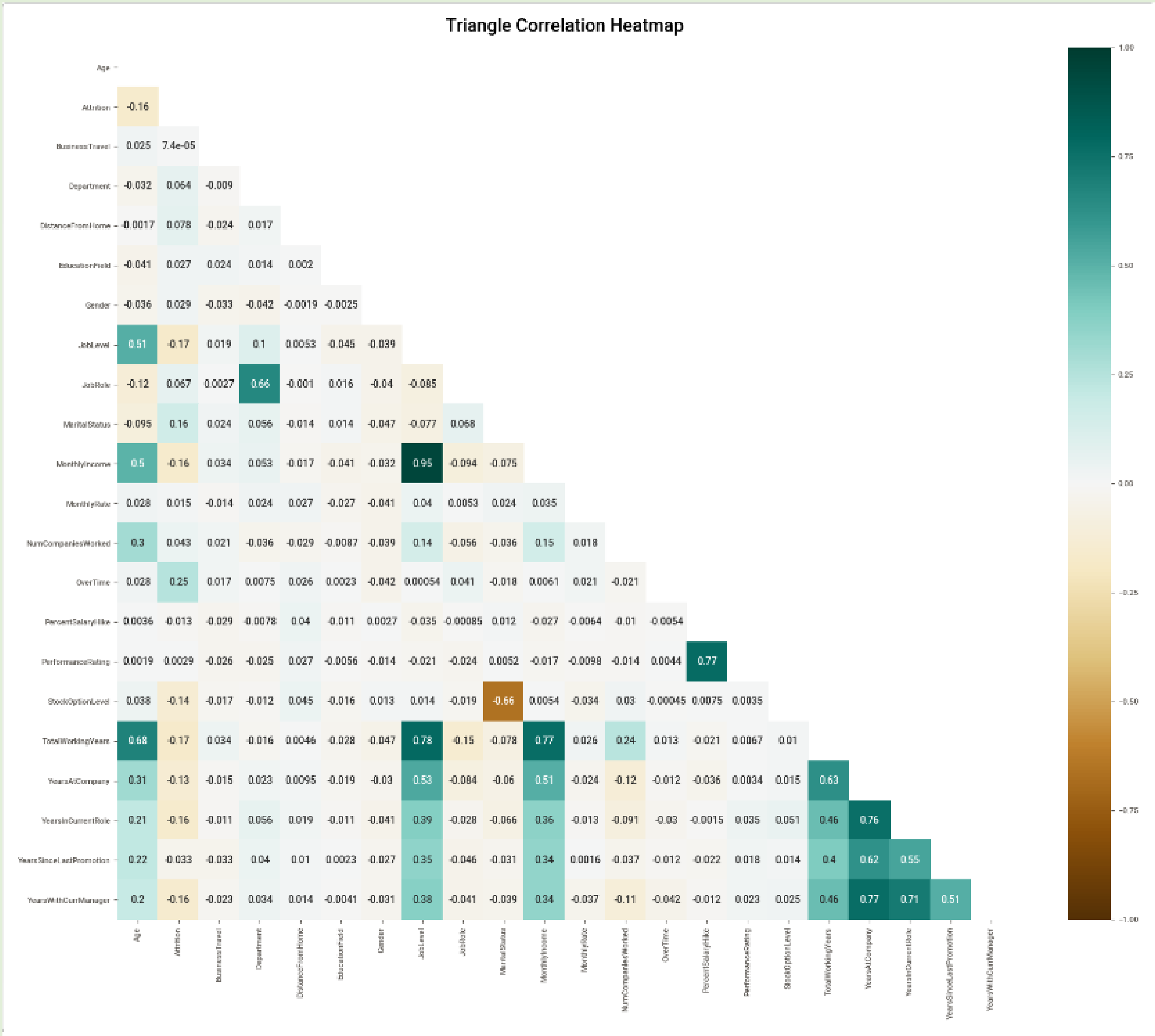
Prior to final features selection Data Encoding of Object or String Columns was carried out to facilitate any statistical computation during features selection process. Hence, after deep copying of original dataset, a function was created and employed to encode object data using Scikit-learn label encoder.

2) Data Encoding of Object/String Columns:

* List all Object/String Columns
* Deep copy the original data
* Create Function to employ Scikit-learn label encoding to encode object data
* Create a new dataframe with encoded data description to attach to model outcomes

```
#Function to encode object/string columns

#List all Object/String Columns
from sklearn import preprocessing
cat_columns = df.select_dtypes(include=[object]) # Get Object Type Columns to Convert to Encoded Categories
#cat_columns.info()

categorical_column = list(cat_columns.columns)# list of columns to for label encoding

print(colored("\n\nColumns Requiring Encoding: \n", 'blue', attrs=['bold'])
              + colored(categorical_column, 'green', attrs=['bold']))

#Deep copy the original data
df_encoded = df.copy(deep=True)

# Make Empty Dataframe to decode encoded data later
decode_features = pd.DataFrame()

##### Employ Scikit-Learn label encoding to encode object data #####
lab_enc = preprocessing.LabelEncoder()
for col in categorical_column:
        df_encoded[col] = lab_enc.fit_transform(df[col])
        le_name_mapping = dict(zip(lab_enc.classes_, lab_enc.transform(lab_enc.classes_)))

        ##### Decode Encoded Data #####
        feature_df = pd.DataFrame([le_name_mapping])
        feature_df = feature_df.astype(str)
        print(feature_df)
        feature_df= (col + "_" + feature_df.iloc[0:])
        feature_df["Feature"] = col
        print(feature_df)
        decode_features = decode_features.append(feature_df)# Append Dictionaries to Empty Dataframe for Later Decoding

        ##### Print Encoded Data #####
        print(colored("Feature: \n", 'blue', attrs=['bold'])
              + colored(col, 'red', attrs=['bold'])
              + colored("\nMapping: \n", 'blue', attrs=['bold'])
              + colored(le_name_mapping, 'green', attrs=['bold'])
              + colored("\n\n", 'blue', attrs=['bold'])
             )
df_encoded.head(3)

##### Make Decoded Factor Dataframe with Description #####
#print(decode_features)
factor_list = decode_features.T # Transpose Dataframe and place in new dataframe
factor_list = factor_list.replace(np.nan, "/") # nan values with forward slash
factor_list["Factors"] = factor_list.astype(str).agg("".join,axis=1).replace(r'[^\w\s]|/', '', regex=True) # Aggregate All (
factor_list.reset_index() # Reset index before copying/assigning it to a new column
factor_list['Description'] = factor_list.index  # Assign index to column
```

Statistical measures were then employed with supervised filter-based feature selection technique. Using Pearson's Correlation, the first set of features are selected based on the strength of positive correlation with taget variable 'Attrition'. Additionaly, Pearson's Correlation Matrix was also computed to select feature pairs exhibiting positive correlations with each other.



All feature lists were then combined to filter out dataframe columns not included in the 'final_features' list.

```
df_encoded = df_encoded.filter(final_features)
df_encoded.info()
```

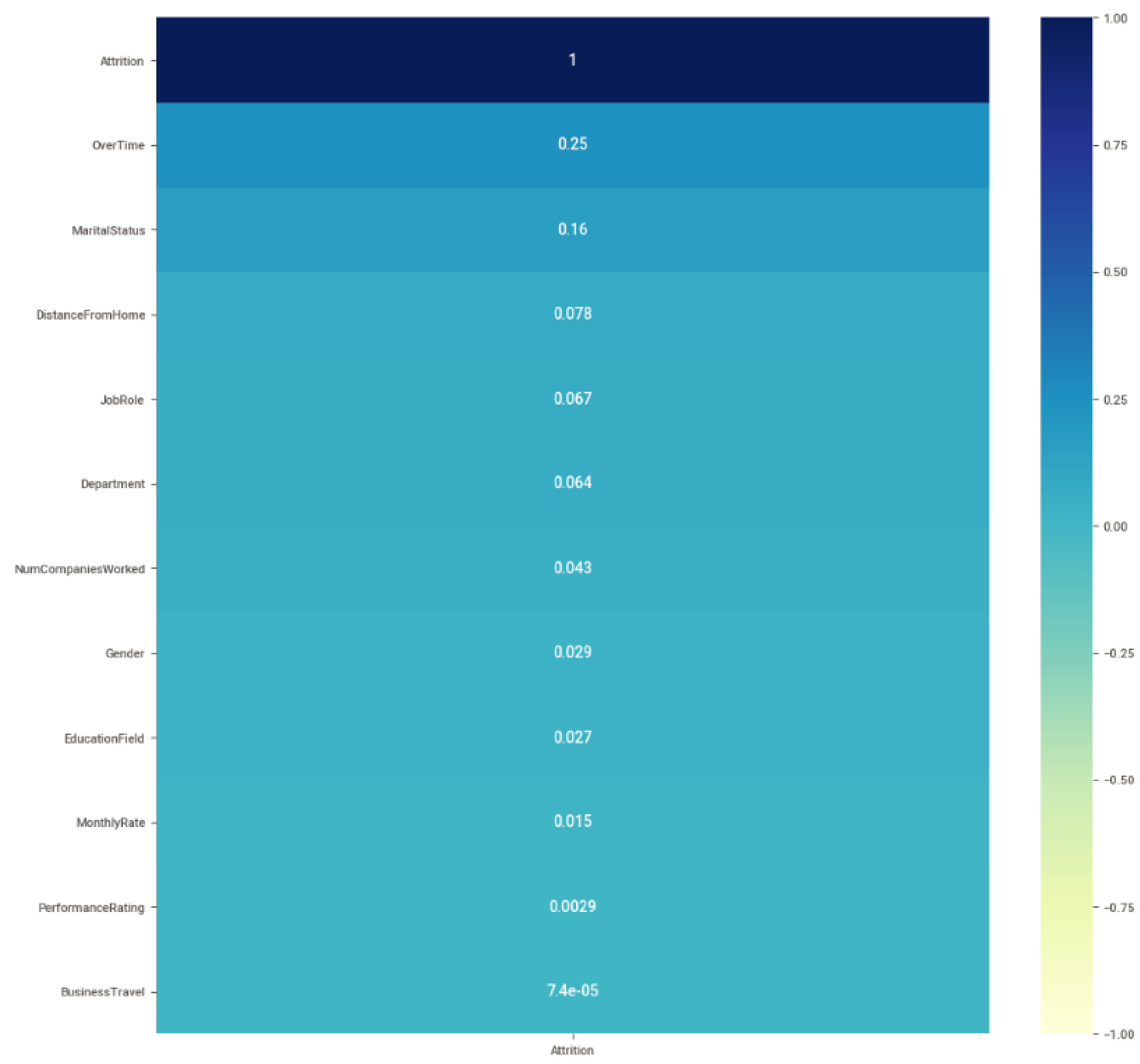
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1470 entries, 0 to 1469
Data columns (total 22 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Age                      1470 non-null   int64
 1   Attrition                1470 non-null   int32
 2   BusinessTravel           1470 non-null   int32
 3   Department               1470 non-null   int32
 4   DistanceFromHome         1470 non-null   int64
 5   EducationField           1470 non-null   int32
 6   Gender                   1470 non-null   int32
 7   JobLevel                 1470 non-null   int64
 8   JobRole                  1470 non-null   int32
 9   MaritalStatus            1470 non-null   int32
 10  MonthlyIncome            1470 non-null   int64
 11  MonthlyRate              1470 non-null   int64
 12  NumCompaniesWorked       1470 non-null   int64
 13  OverTime                 1470 non-null   int32
 14  PercentSalaryHike        1470 non-null   int64
 15  PerformanceRating        1470 non-null   int64
 16  StockOptionLevel         1470 non-null   int64
 17  TotalWorkingYears        1470 non-null   int64
 18  YearsAtCompany           1470 non-null   int64
 19  YearsInCurrentRole       1470 non-null   int64
 20  YearsSinceLastPromotion  1470 non-null   int64
 21  YearsWithCurrManager     1470 non-null   int64
dtypes: int32(8), int64(14)
memory usage: 218.2 KB
```

5) Key Findings and Insights, which synthesizes the results of Exploratory Data Analysis in an insightful and actionable manner

5a) Summary of Features Exhibiting Positive Correlation with Target

The following features exhibited positive correlations with target variable:

```
Positive Correlations between Individual Features and Target Variable
             Features1      Corr
18            OverTime  0.246118
14       MaritalStatus  0.162070
4     DistanceFromHome  0.077924
12             JobRole  0.067151
3           Department  0.063991
17  NumCompaniesWorked  0.043494
8               Gender  0.029453
6       EducationField  0.026846
16         MonthlyRate  0.015170
20   PerformanceRating  0.002889
1       BusinessTravel  0.000074
```



5b) Summary of Positive Correlations for Feature Pairs

In addition, the following feature pairs displayed high correlation

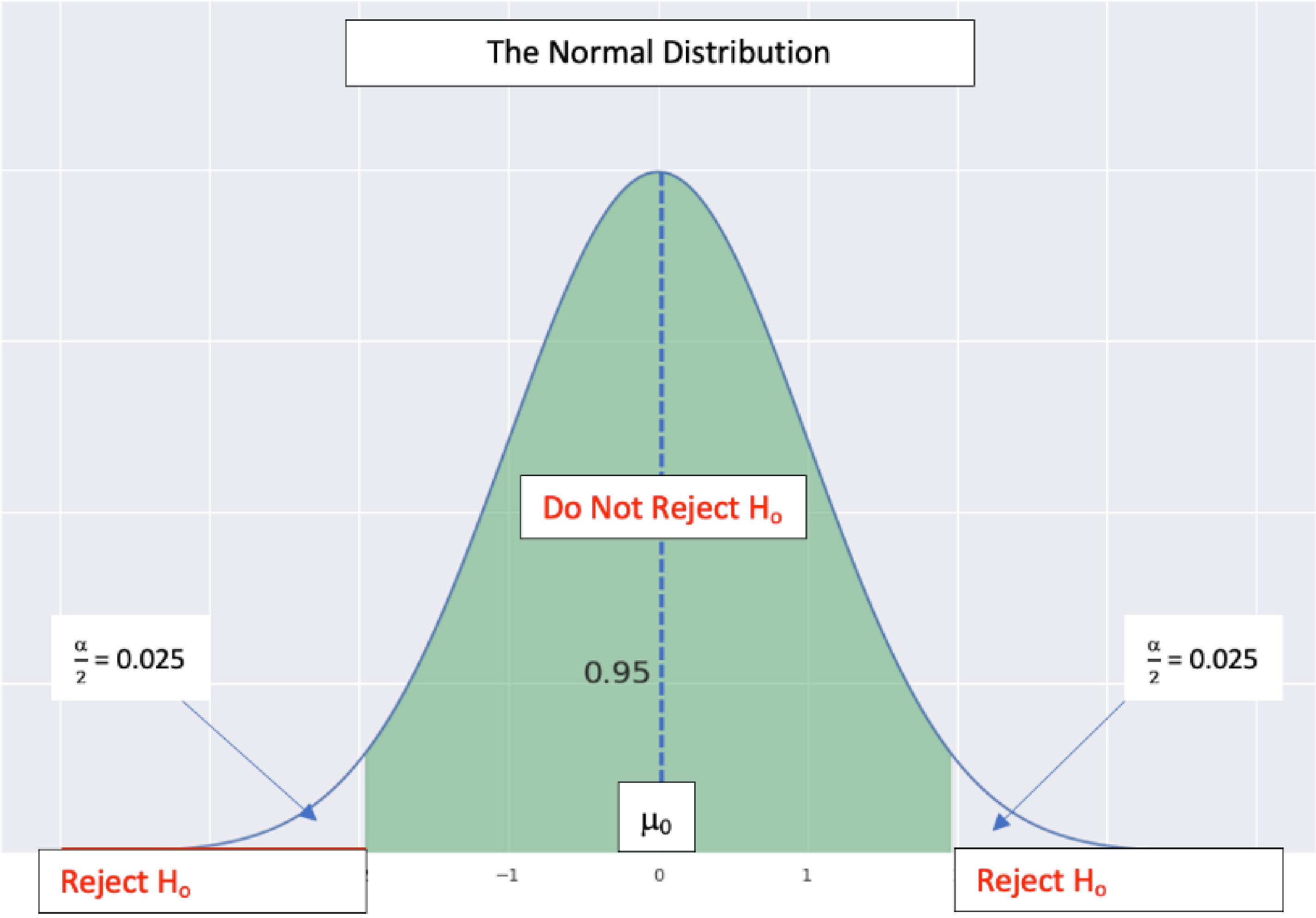
```
Correlation Matrix Results for Feature Pairs
                Features2                  Features3  Correlation_abs
31          MonthlyIncome                   JobLevel         0.950300
33      TotalWorkingYears                   JobLevel         0.782208
35      PerformanceRating          PercentSalaryHike         0.773550
37      TotalWorkingYears              MonthlyIncome         0.772893
39   YearsWithCurrManager             YearsAtCompany         0.769212
41         YearsAtCompany         YearsInCurrentRole         0.758754
43   YearsWithCurrManager         YearsInCurrentRole         0.714365
45                    Age          TotalWorkingYears         0.680381
47       StockOptionLevel              MaritalStatus         0.662577
49                JobRole                 Department         0.662431
51      TotalWorkingYears             YearsAtCompany         0.628133
53         YearsAtCompany    YearsSinceLastPromotion         0.618409
```

Based on the above two correlation findings, the following final features were selected while others were dropped as they were considered to bear no or less impacts towards employee attrition.

```
List of Final Features
['Age' 'Attrition' 'BusinessTravel' 'Department' 'DistanceFromHome'
 'EducationField' 'Gender' 'JobLevel' 'JobRole' 'MaritalStatus'
 'MonthlyIncome' 'MonthlyRate' 'NumCompaniesWorked' 'OverTime'
 'PercentSalaryHike' 'PerformanceRating' 'StockOptionLevel'
 'TotalWorkingYears' 'YearsAtCompany' 'YearsInCurrentRole'
 'YearsSinceLastPromotion' 'YearsWithCurrManager']
```

6) Three Major Hypothesis

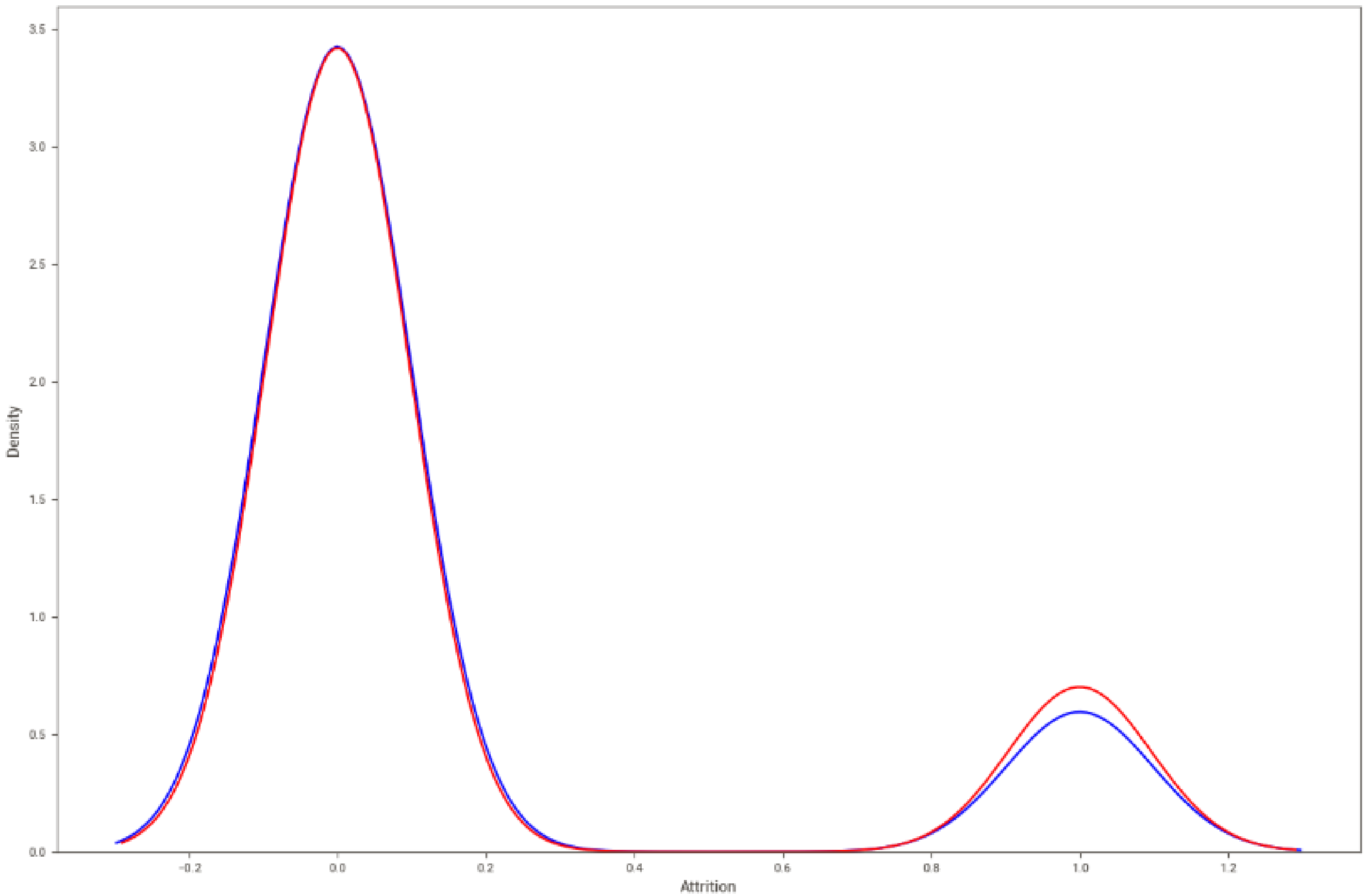
Null hypothesis (H_0) is a statistical hypothesis which postulates random factors causing difference in observations.
Alternative hypothesis (H_A) is a statistical hypothesis which postulates real impacts causing difference in observations.



Based on the above diagram, the significance level is the decision point for null hypothesis acceptance or vice versa.
This significance level is normally set at 5%, 1% or 0.5%, depending on business requirements.
Hence, given a 5% significance level, alpha$(\alpha) = 0.05$.
Consequently, for 2-tailed test, alpha should be divided by 2, which will yield 0.025 for alpha set at 0.05.
Hence, Null Hypothesis will be Rejected if computed p-value is less than alpha and vice versa.

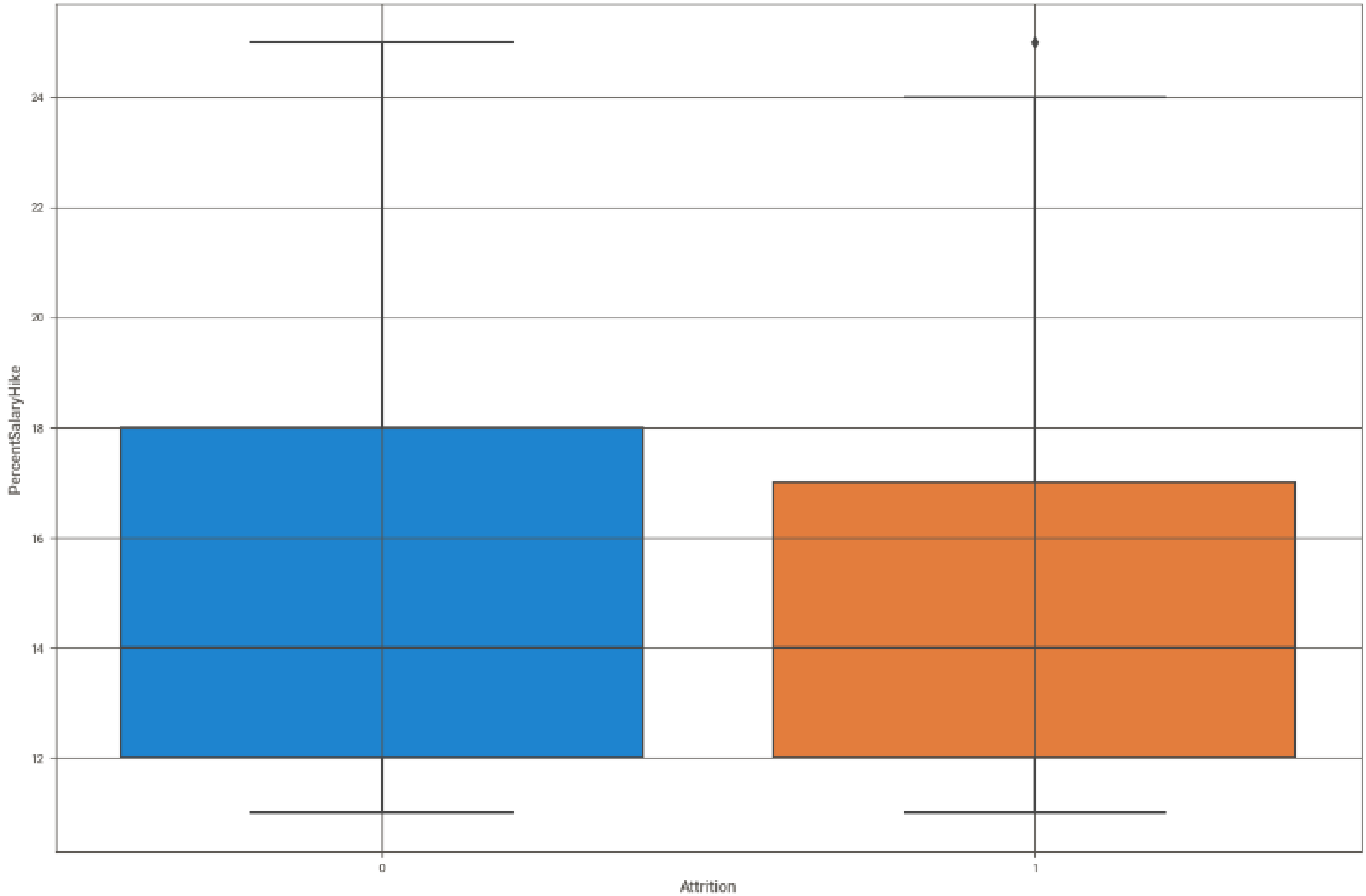
6a) Hypothesizing Gender Differences in Attrition Rates

```
H_01:μ_1-μ_2=0  There is No Significant Gender Difference between Employee Attrition of Female and Male Workers.
H_A1:μ_1-μ_2!=0   There is Significant Gender Difference between Employee Attrition of Female and Male Workers.

%age Distribution of Gender

      %  Gender
1  60.0       1
0  40.0       0

Value Counts show an imbalanced Class Distribution with 60.0 %  in class 1 and 40.0 %  in class 0

Mean Attrition for Females is: 0.14795918367346939
Mean Attrition for Males is: 0.17006802721088435

t_value1 = -1.1289761152328313
p_value1 = 0.25909236414147996


Conclusion:
Since p_value 0.25909236414147996 is greater than alpha 0.05
We Accept the Null Hypothesis that there is No Significant Gender Difference between Attrition Rates of Female and Male Worker
s.

<AxesSubplot:xlabel='Attrition', ylabel='Density'>
```



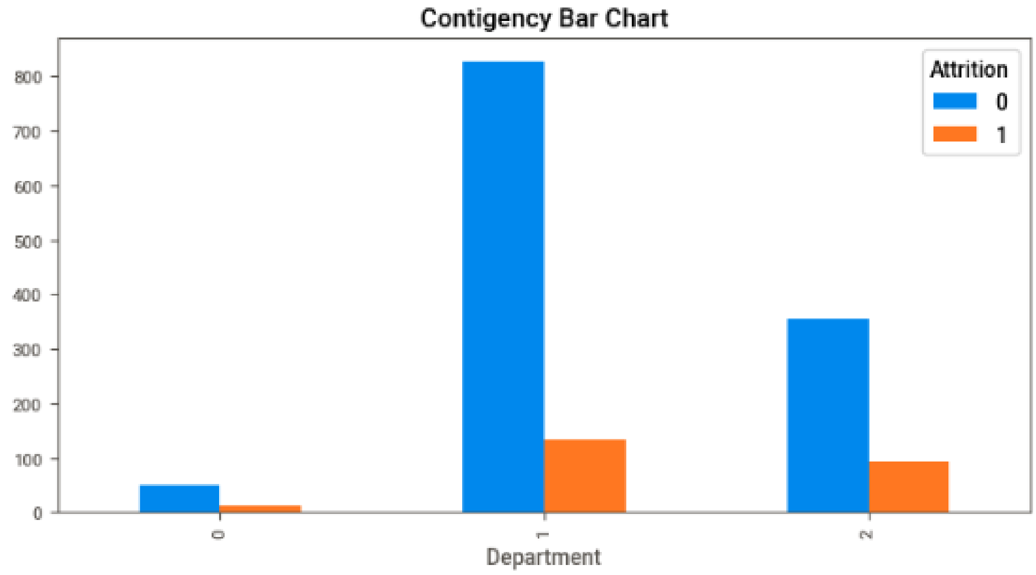
6b) Hypothesizing Differences in Salary Hike between Leavers and Stayers

```
H_02:μ_1-μ_2=0  Average Salary Hike of Leavers is Less than or Equal to Stayers.
H_A2:μ_1-μ_2!=0  Average Salary Hike of Leavers is Greater than or Equal to Stayers.

Summary Statistics

AOV Table Results
                         df       sum_sq   mean_sq         F  PR(>F)
C(PercentSalaryHike)    14.0    1.805708  0.128979  0.952689  0.5006
Residual              1455.0  196.984088  0.135384       NaN     NaN

p_val_sal = 0.5005995155055496


Conclusion:
Since p_value 0.5005995155055496 is greater than alpha 0.05
We Accept Null Hypothesis that Average Salary Hike of Leavers is Less than or Equal to Stayers.
```



6c) Hypothesizing Differences in Salary Hike between Leavers and Stayers

```
H_03:μ_1-μ_2=0  There is No Significant Difference in Attrition Rate Across Different Departments.
H_A3:μ_1-μ_2!=0   There is Significant Difference in Attrition Rate Across Different Departments.

Contigency Table
Attrition     0    1
Department
0            51   12
1           828  133
2           354   92

Summary Statistics

Mean Attrition for Department 0 is: 0.19047619047619047
Mean Attrition for Department 1 is: 0.1383975026014568
Mean Attrition for Department 2 is: 0.2062780269058296

Chi-Square Statistic: 10.79600732241067
p Value: 0.004525606574479633
Degree of Freedom: 2
Expected Frequencies: [[ 52.84285714  10.15714286]
 [806.06326531 154.93673469]
 [374.09387755  71.90612245]]


Conclusion:
Since p_value 0.004525606574479633 is less than alpha 0.05
We Reject the Null Hypothesis that there is No Significant Difference in Attrition Rate Across Different Departments.

Text(0.5, 1.0, 'Contigency Bar Chart')
```



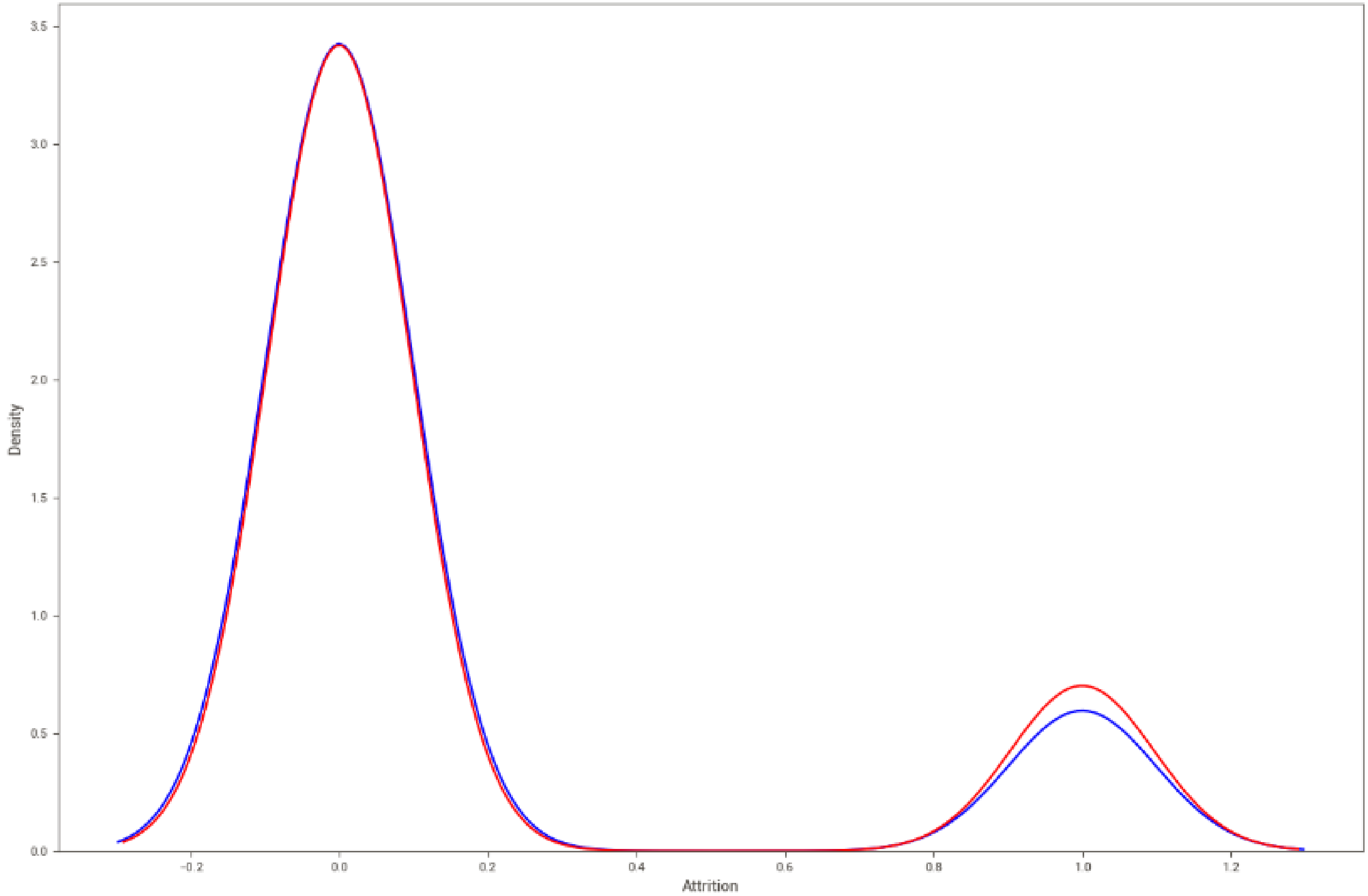
7) Conducting a formal significance test for one of the hypotheses and discuss the results

7a) Significance Test for Hypothesis 1

Hypothesis 1: Hypothesizing Gender Differences in Attrition Rates.

Due to unknown standard deviation, a one-tailed t-test has been used for testing population means between female and male Genders. Since, it is a one-tailed test, at 5% significance level, 0.05 alpha (α) has been used.

```
H_01:μ_1-μ_2=0  There is No Significant Gender Difference between Employee Attrition of Female and Male Workers.
H_A1:μ_1-μ_2!=0   There is Significant Gender Difference between Employee Attrition of Female and Male Workers.

%age Distribution of Gender

      %  Gender
1  60.0       1
0  40.0       0

Value Counts show an imbalanced Class Distribution with 60.0 %  in class 1 and 40.0 %  in class 0

Mean Attrition for Females is: 0.14795918367346939
Mean Attrition for Males is: 0.17006802721088435

t_value1 = -1.1289761152328313
p_value1 = 0.25909236414147996

<AxesSubplot:xlabel='Attrition', ylabel='Density'>
```



7b) Results and Discussion

```
Results:
Since p_value 0.25909236414147996 is greater than alpha 0.05
We Accept the Null Hypothesis that there is No Significant Gender Difference between Attrition Rates
of Female and Male Workers.
```

8) Suggestions for next steps in analysing this data

- EDA now should proceed towards analysing problematic data outliners for all final features to increase statistical significance of the model.
- Using Gower Distancing, further cluster analysis can be done to gain in-depth understanding of employee clusters at risk of turnover rather than utilizing one size fit all approach (see, Section 10(e).

9) A paragraph that summarizes the quality of this data set and request for additional data if needed

Overall data quality was quite good since EDA exhibited zero null values and contained extensive variables to work on. However, while the data did contain extensive parameters, nevertheless, including other pull factors like community fit, workload, etc. may prove to be useful. For example, based on social relationship theory, pull factors like community fit, industry, etc has been found to exhibit a negative correlation with intentions to leave, especially with increased perceptions of needs fulfilment and social networking (see, Ramesh and Gelfand, 2010). Hence, including this information can shed further light on employee turnover. Then, including other mooring factors, such as, personal life involvement may also improve data quality as well as model's predictive power since these may also serve as underlying factors in employee turnover. For instance, numerous studies found higher turnover rates among employees exhibiting high family centrality and work interference with family life (see, Bagger et al., 2008; Haldorai, et al., 2019). Hence, provision of these additional parameters may render enhanced insight into employee attrition and may even change predictive outcomes.

10) Link to Other Useful Models

a) https://github.com/IBM/employee-attrition-aif360/blob/master/notebooks/employee-attrition.ipynb
b) https://github.com/JNYH/employee_attrition/blob/master/employee_attrition.ipynb
c) https://github.com/elastic/examples/tree/master/Machine%20Learning/Analytics%20Jupyter%20Notebooks
d) https://github.com/ganesh10-india/HR_Analytics-Employee_Attrition-Classification-Models/blob/main/HR_Analytics_Employee_Attrition_Classification_Models.ipynb
e) https://www.adam-d-mckinnon.com/posts/2020-08-04-clusteranalysis/

11) Github Link to Assignment Notebook

https://github.com/FATIMASP/IBM-MACHINE-LEARNING-CERTIFICATION/blob/main/Exploratory%20Data%20Analysis%20for%20Machine%20Learning/EDA%20SUPERVISED%20CLASSIFICATION%20EMPLOYEE%20ATTRITION.ipynb