**HR Analytics Project- Understanding the Attrition in HR**

*By Kunal Chand*

**1. Problem Statement:**

Every year a lot of companies hire many employees. The companies invest time and money in training those employees, not just this but there are training programs within the companies for their existing employees as well. These programs aim to increase the effectiveness of their employees. But where does HR Analytics fit in this? and is it just about improving the performance of employees?

**HR Analytics**

Human resource analytics (HR analytics) is an area in the field of analytics that refers to applying analytic processes to the human resource department of an organization in the hope of improving employee performance and therefore getting a better return on investment. HR analytics does not just deal with gathering data on employee efficiency. Instead, **it aims to provide insight into each process by gathering data and then using it to make relevant decisions about how to improve these processes.**

**Attrition in HR**

Attrition in human resources refers to the gradual loss of employees over time. In general, relatively high attrition is problematic for companies. HR professionals often assume a leadership role in designing company compensation programs, work culture, and motivation systems that help the organization retain top employees.

How does Attrition affect companies? and how does HR Analytics help in analyzing attrition? We will discuss the first question here and for the second question, we will write the code and try to understand the process step by step.

**Attrition affecting Companies**

A major problem in high employee attrition is its cost to an organization. Job postings, hiring processes, paperwork, and new hire training are some of the common expenses of losing employees and replacing them. Additionally, regular employee turnover prohibits your organization from increasing its collective knowledge base and experience over time. This is especially concerning if your business is customer-facing, as customers often prefer to interact with familiar people. Errors and issues are more likely if you constantly have new workers.

**2. Data Analysis**.

Data exploration is the first step in data analysis and typically involves summarizing the main characteristics of a data set, including its size, accuracy, initial patterns in the data, and other attributes. It is commonly conducted by data analysts using visual analytics tools, but it can also be done in more advanced statistical software, Python.

Before it can analyze data collected by multiple data sources and stored in data warehouses, an organization must know how many cases are in a data set, what variables are included, how many missing values there are, and what general hypotheses the data is likely to support. An initial exploration of the data set can help answer these questions by familiarizing analysts with the data with

Which they are working. We divided the data 8:2 for Training and Testing purposes respectively.

# 3. EDA Concluding Remark

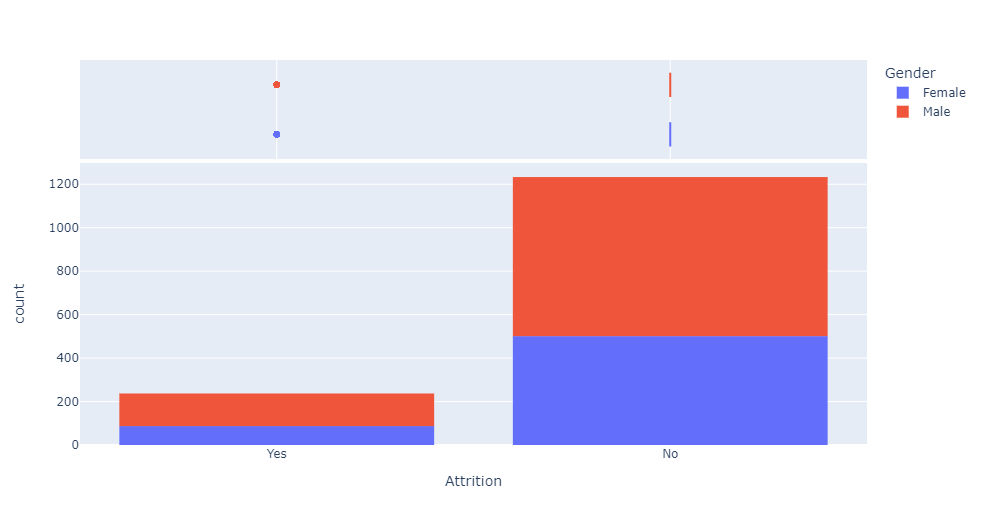
As for any basic model building, we have to understand the type of target variable; the data of the target variable is continued or classified.

Data Analysis is always the difficult part, for better understanding different kinds of bar plots, distribution plots are created with the target Column for finding the insights of the dataset we have.

Analytical Modeling always starts with the target variable we have, and in that case, our target variable is Attrition, for that, we create some box plots with the target variable to understand which feature columns help to learn the model best and which feature columns reduce the accuracy of the model.

And after finding the relation and correlation with the target variable we choose either Regression Model or Classification Model. Here in this problem, our target feature column is categorized data so we build our Machine Learning model on Regression.

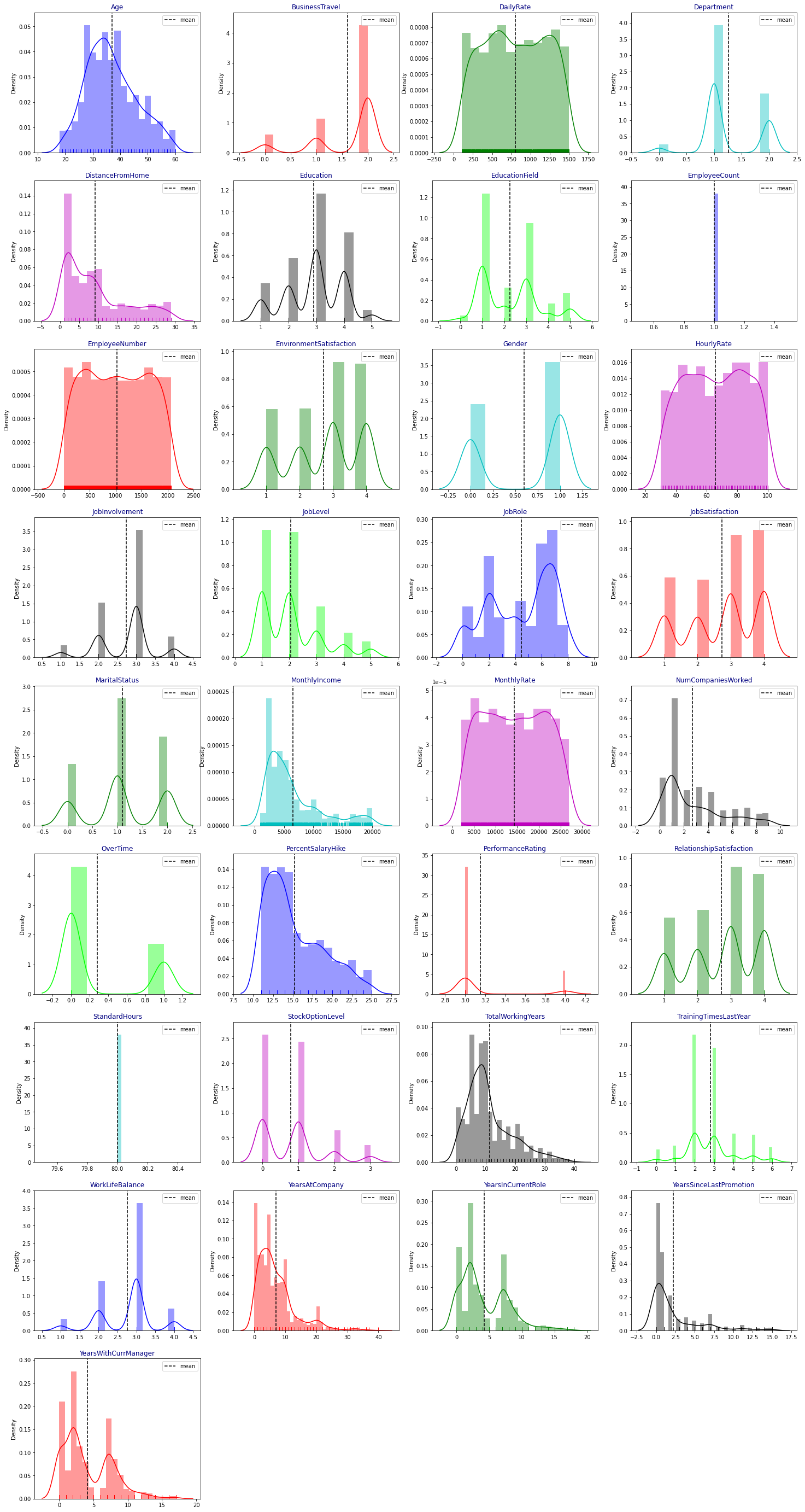
Visualization of the Attrition with gender column.



observations:

1. So, on average, Male Employees left the company more as compared to female employees...

# Use subplot and distribution plot to check data are normalized or not.



observations:

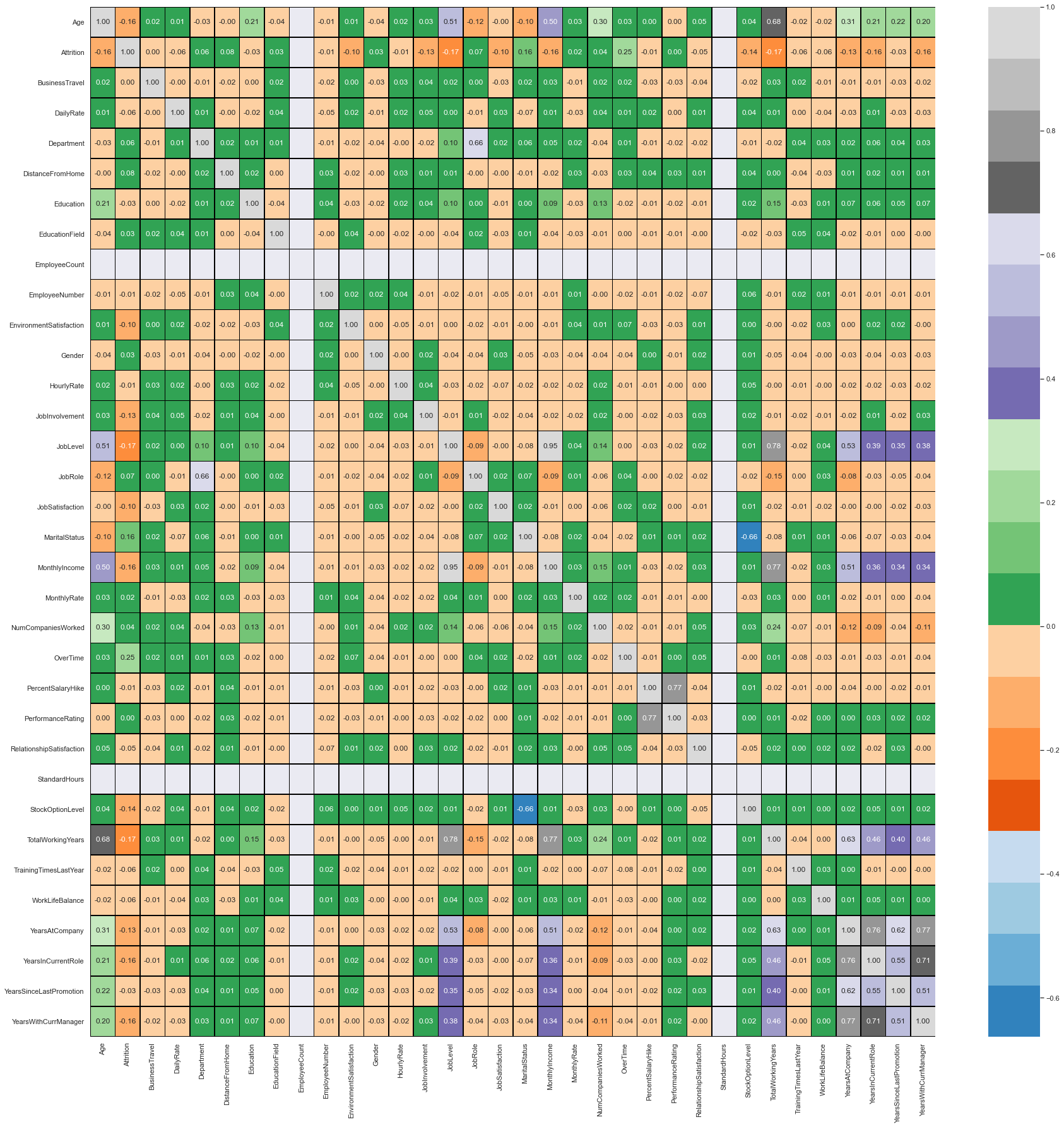
1. From the above plotting of distribution plot we see that some features columns are not normally distributed.

2. some columns are skewed towards the right.

3. Building blocks are out of the normal curve hence

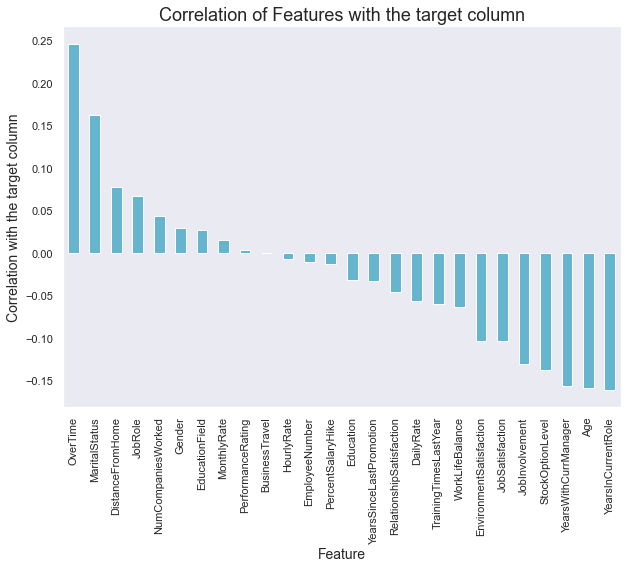
outliers are present.

# CORRELATION BETWEEN THE COLUMNS:



Correlation: From the above result it is clear that some columns make positive correlations and some make the negative correlation. The positively correlated columns have a great impact on the target column while the negatively correlated have less or zero impact on the target column.

**Most Impacting feature column**

****

**Observations:**

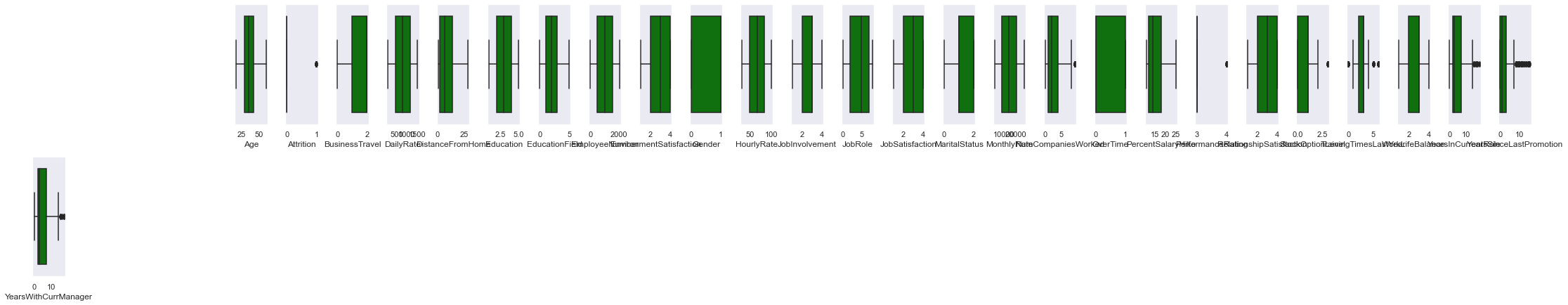
1. overtime, Martial status, and Distance from home make a high impact on attrition.

2. Year in current role Age years with current manager reduces the attrition.

3. those employees are near from their home town and single are less like to change the job.

# Detecting outliers:

# download (43).png

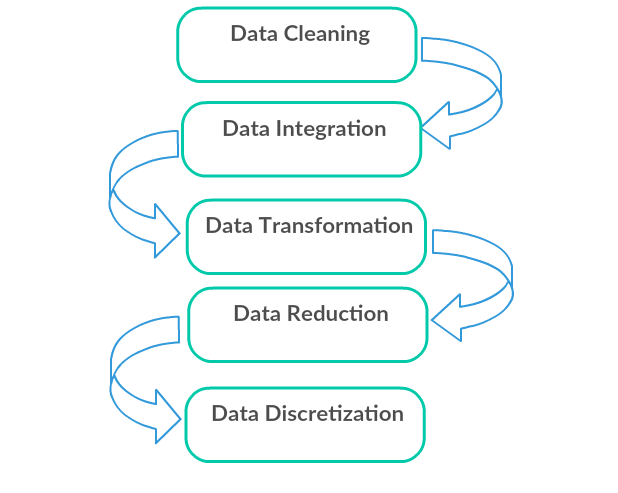


Observations:

So, many outliers are present

**4. Pre-Processing Pipeline.**

Data pre-processing can refer to the manipulation or dropping of data before it is used to ensure or enhance performance, and is an important step in the data mining process.



1. Data Cleaning: First we clean the data which have no use in prediction like the Over18 column, and then we drop the data which has a high no of missing percentages.
2. Data Integration: then we do some EDA process for finding out the meaning full insights of the data.
3. Data transformation is the process of changing the format, structure, or values of data; we use a labeled encoder for coding the object data into integer data.
4. Data Reduction: it is the process of finding the most correlated columns, and combining them because the machine does not understand which feature columns impact the most on accuracy.
5. Data discretization converts a large number of data values into smaller once, so that data evaluation and data management becomes very easy, using box plots is makes a clear understanding of the data.

**5. Building Machine Learning Models**.

After analyzing the dataset, I observe that many of the feature columns are object type so first, we have to convert them in integer or float type so that machine interpret the data and for that we do label encode all the feature column.

After label encoding, we find that many feature columns have Nan values so we use mean and median for filling that missing data,

Then find the correlation between the columns with target columns and delete the non-related feature columns.

We observe that the target column is skewed so we remove the skewness of the target column because normal data gives better results when we make the M.L model.

The target column is object type so we start work on Classification Modelling building.

* Testing of Identified Approaches (Algorithms)

1. Logistic Regression
2. Regurgitation:

Ridge Classification

1. Ensemble techniques

DecisionTreeClassifier

# Random Forest Classifier

# 4. Support Vector Classifier

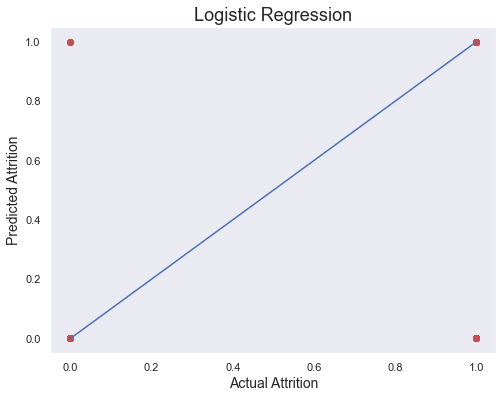
5. K-nearest Neighbour Classifier

**Logistic Regression Model**

• Logistic Regression is a machine learning algorithm based on supervised learning.

• It performs a regression task. Regression models a target prediction value based on independent variables.

• It is mostly used for finding out the relationship between variables and forecasting.



**print(confusion\_matrix(y\_test,pred\_test))**

**print(accuracy\_score(y\_test,pred\_test))**

**print(classification\_report(y\_test,pred\_test))**

**[[243 3]**

**[ 34 14]]**

**0.8741496598639455**

**precision recall f1-score support**

**0 0.88 0.99 0.93 246**

**1 0.82 0.29 0.43 48**

**accuracy 0.87 294**

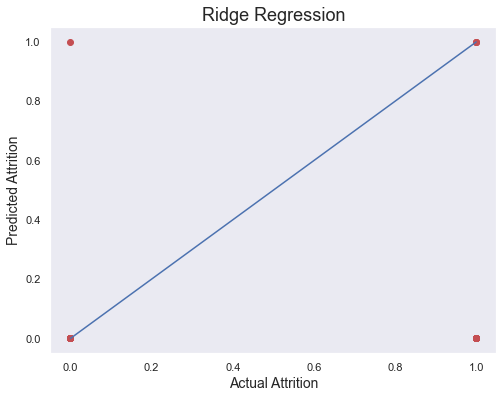
**macro avg 0.85 0.64 0.68 294**

**weighted avg 0.87 0.87 0.85 294**

# 1. Ridge

The [**Ridge**](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html#sklearn.linear_model.Ridge) regressor has a classifier variant: **[RidgeClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html" \l "sklearn.linear_model.RidgeClassifier" \o "sklearn.linear_model.RidgeClassifier)**. This classifier first converts binary targets to {-1, 1} and then treats the problem as a regression task, optimizing the same objective as above. The predicted class corresponds to the sign of the regressor’s prediction. For multiclass classification, the problem is treated as multi-output regression, and the predicted class corresponds to the output with the highest value.

It might seem questionable to use a (penalized) Least Squares loss to fit a classification model instead of the more traditional logistic or hinge losses. However, in practice, all those models can lead to similar cross-validation scores in terms of accuracy or precision/recall, while the penalized least squares loss used by the **[RidgeClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html" \l "sklearn.linear_model.RidgeClassifier" \o "sklearn.linear_model.RidgeClassifier)** allows for a very different choice of the numerical solvers with distinct computational performance profiles.



print(accuracy\_score(y\_test,pred\_rd))

print(confusion\_matrix(y\_test,pred\_rd))

print(classification\_report(y\_test,pred\_rd))

**0.8435374149659864**

**[[245 1]**

**[ 45 3]]**

**precision recall f1-score support**

**0 0.84 1.00 0.91 246**

**1 0.75 0.06 0.12 48**

**accuracy 0.84 294**

**macro avg 0.80 0.53 0.51 294**

**weighted avg 0.83 0.84 0.78 294**

**DecisionTreeClassifier**

**Decision Trees (DTs)** are a non-parametric supervised learning method used for classification. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.



**print(accuracy\_score(y\_test,pred\_decision))**

**print(confusion\_matrix(y\_test,pred\_decision))**

**print(classification\_report(y\_test,pred\_decision))**

**0.7551020408163265**

**[[209 37]**

**[ 35 13]]**

**precision recall f1-score support**

**0 0.86 0.85 0.85 246**

**1 0.26 0.27 0.27 48**

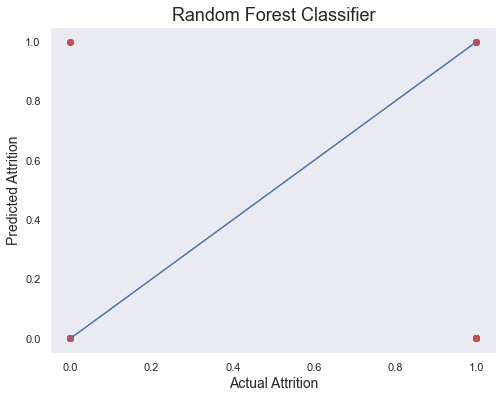
**accuracy 0.76 294**

**macro avg 0.56 0.56 0.56 294**

**weighted avg 0.76 0.76 0.76 294**

### 3 Random Forest Classifiers

The core unit of random forest classifiers is the decision tree. The decision tree is a hierarchical structure that is built using the features (or the independent variables) of a data set. Each node of the decision tree is split according to a measure associated with a subset of the features. The random forest is a collection of decision trees that are associated with a set of bootstrap samples that are generated from the original data set. The nodes are split based on the entropy (or Gini index) of a selected subset of the features. The subsets that are created from the original data set, using bootstrapping, are of the same size as the original data set.



print(accuracy\_score(y\_test,pred\_random))

print(confusion\_matrix(y\_test,pred\_random))

print(classification\_report(y\_test,pred\_random))

**0.8435374149659864**

**[[244 2]**

**[ 44 4]]**

**precision recall f1-score support**

**0 0.85 0.99 0.91 246**

**1 0.67 0.08 0.15 48**

**accuracy 0.84 294**

**macro avg 0.76 0.54 0.53 294**

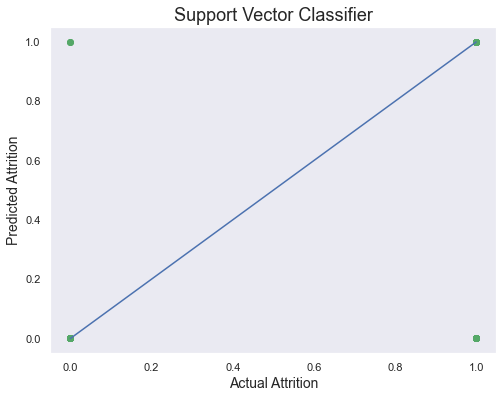
**weighted avg 0.82 0.84 0.79 294**

**Support vector machines (SVMs)**

 are a set of supervised learning methods used for [classification](https://scikit-learn.org/stable/modules/svm.html#svm-classification).

The advantages of support vector machines are:

* Effective in high dimensional spaces.
* Still effective in cases where many dimensions are greater than the number of samples.
* Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
* Versatile: different [Kernel functions](https://scikit-learn.org/stable/modules/svm.html#svm-kernels) can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.



print(accuracy\_score(y\_test,pred\_support))

print(confusion\_matrix(y\_test,pred\_support))

print(classification\_report(y\_test,pred\_support))

**0.8707482993197279**

**[[244 2]**

**[ 36 12]]**

**precision recall f1-score support**

**0 0.87 0.99 0.93 246**

**1 0.86 0.25 0.39 48**

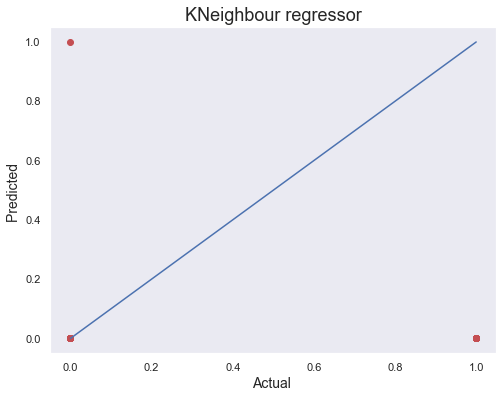
**accuracy 0.87 294**

**macro avg 0.86 0.62 0.66 294**

**weighted avg 0.87 0.87 0.84 294**

K-Nearest Neighbour Classification

1. K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on the Supervised Learning technique.
2. K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.
3. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a good suite category by using K- NN algorithm.
4. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for Classification problems.
5. K-NN is a **non-parametric algorithm**, which means it does not make any assumptions on underlying data.
6. It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
7. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.



print(accuracy\_score(y\_test,pred\_k))

print(confusion\_matrix(y\_test,pred\_k))

print(classification\_report(y\_test,pred\_k))

**0.8333333333333334**

**[[245 1]**

**[ 48 0]]**

**precision recall f1-score support**

**0 0.84 1.00 0.91 246**

**1 0.00 0.00 0.00 48**

**accuracy 0.83 294**

**macro avg 0.42 0.50 0.45 294**

**weighted avg 0.70 0.83 0.76 294**

**6. Concluding Remarks.**

# So, our Aim is achieved as we have successfully ticked all our parameters as mentioned in our Aim Column. It is seen overall Quality is the most effective attribute in predicting the house price and that the Logistic regression is the most effective model for our Dataset.

# We tested 6 models out of which Logistic Regression performing well:

# Plotting Auc-Roc curve..with logistic regression prediction.

# download (28).png

**Precision recall f1-score support**

**0 0.88 0.99 0.93 246**

**1 0.82 0.29 0.43 48**

**accuracy 0.87 294**

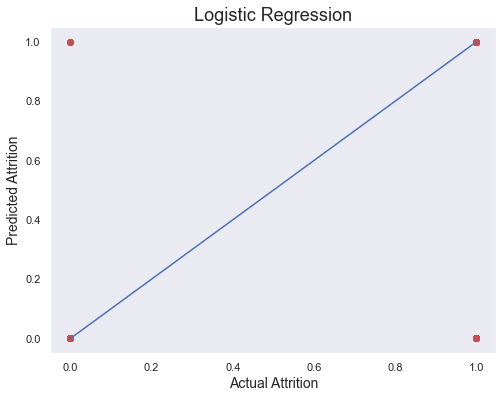
**macro avg 0.85 0.64 0.68 294**

**weighted avg 0.87 0.87 0.85 294**

Cross-validation score is:- 86.60032919347597

accuracy\_score is :- 87.41496598639455

Our Model performs with an Accuracy Score of 87.41%...

****

That's it! We reached the end of our exercise.

# Throughout this kernel, we put into practice many of the strategies

# for predicting the attrition of HR Analytics.

We philosophized about the variables, we analyzed Attrition alone and with the most correlated variables, we dealt with missing data and outliers, we tested some of the fundamental statistical assumptions and we even transformed categorical variables into dummy variables. That's a lot of work that Python helped us make easier**.**

**Thank you**