## 

## PROJECT : HR ANALYTIC – UNDERSTANDING ATTRITION IN HR

**In association with Data Trained Academy Batch: 1842**

**PREPARED BY :- NIDHI SINGH**

Example of an end-to-end machine learning project in Data Science for beginners.

I am going to write about a complete end-to-end project for HR Analytic Project which should serve as a guiding path for many Data Science aspirants.

I have written down all the techniques in the form of sub-topics that I will be explaining one by one. And those sub-topics are as follows:  
  
1.      Problem Definition.  
2.      Data Analysis.  
3.      EDA Concluding Remark.  
4.      Pre-Processing Pipeline.  
5.      Building Machine Learning Models.  
6.      Concluding Remarks.

Let’s start with the problem definition or a short introduction on the project ‘ Attrition in HR’ that I have chosen to elaborate.

**Introduction:**

**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 is **the departure of employees from the organization for any reason (voluntary or involuntary), including resignation, termination, death or retirement**.

**Attrition in HR**

Attrition in human resources refers to the gradual loss of employees overtime. 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?**

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.

Thanks to Data Science and Machine Learning, which has been very useful in many industries that have managed to bring accuracy or detect negative incidents. Here in this blog, I have created a Machine Learning model to detect how does HR Analytics help in analyzing attrition. In total the dataset has 35 features and 1470 entries rows of data. Using all these previously acquired information and analysis done with the data I have achieved a good model that has 93% accuracy. Let’s see what are the steps involved to attain this accuracy.

Various visualization techniques have also been used to understand the Correlation, Multicollinearity and importance of the features with respect to the algorithms.

Note: Various special words are used in the article assuming the fact that the reader is aware of the language used in data science.

**Hardware & Software Requirements & Tools Used:**

### Hardware used:

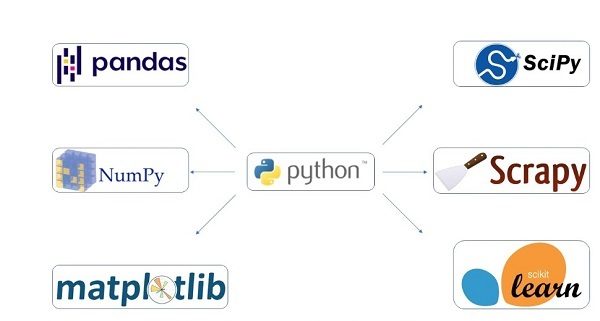
* Processor: Core i5 -10300H CPU @ 2.50GHz
* RAM: 8 GB
* Operating System: 64-bit
* ROM/SSD: 1 TB SSD
* Graphics: NVIDIA GeForce GTX 1650 Ti

**Software requirement**:

* Anaconda Navigator - Jupyter Notebook

**Libraries Used**:

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Scipy
* Date Time
* Scikit Learn



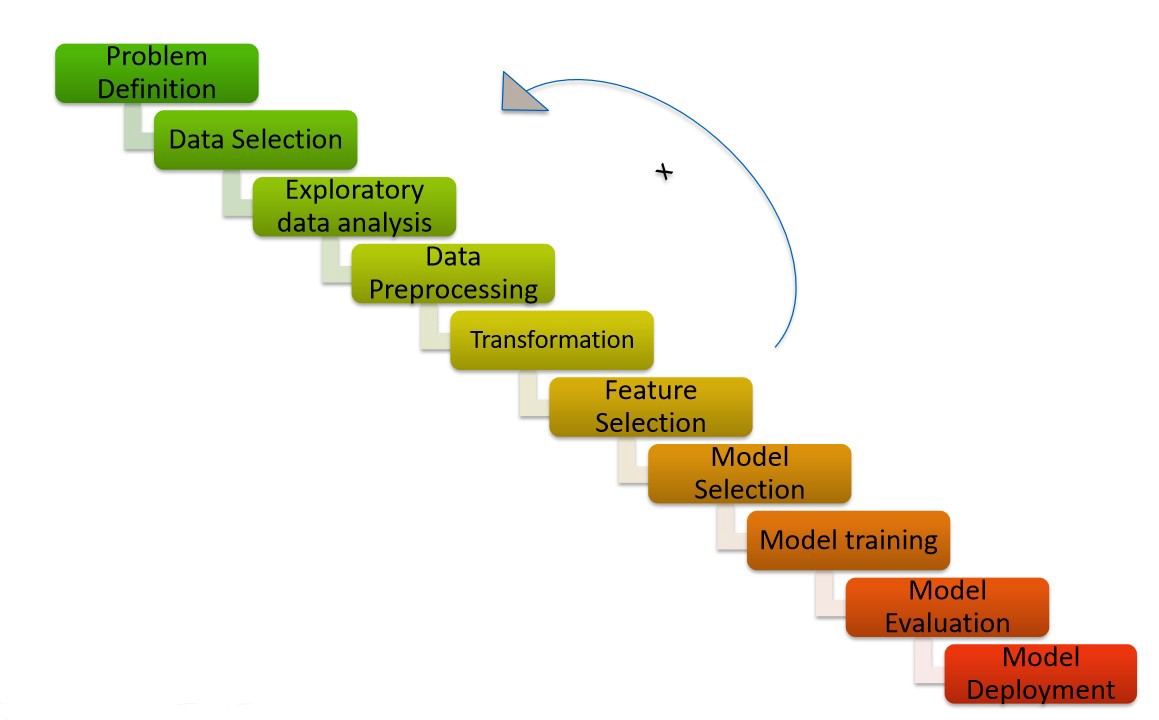
Heading forward we will try to understand the problem statement and the dataset.

1. **Problem Definition**

Every year a lot of companies hire a number of 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. The aim of these programs is to increase the effectiveness of their employees. But where HR Analytics fit in this? and is it just about improving the performance of employees?

1. **Data Analysis**

In order to build a Machine Learning Model, we have a Machine Learning Life Cycle that every Machine Learning Project has to touch upon in the life of the model. Let’s take a look at the model life cycle and then we will look into the actual machine learning model and understand it better along with the lifecycle as we move forward.

****

Now that we understand the lifecycle of a Machine Learning Model, lets import the necessary libraries and proceed further.

**Importing the necessary Libraries:**

To analyze the dataset or even to import the dataset, we have imported all the necessary libraries as shows below.

Pandas has been used to import the dataset and also in creating data frames.

Numpy has been used for numerical tasks.

Seaborn and Matplotlib have been used for Data Visualization.

Date Time has been used to extract day/month/date separately.

Scipy has been used in the Zscore method for removing outliers.

Sklearn has been used in the model building.

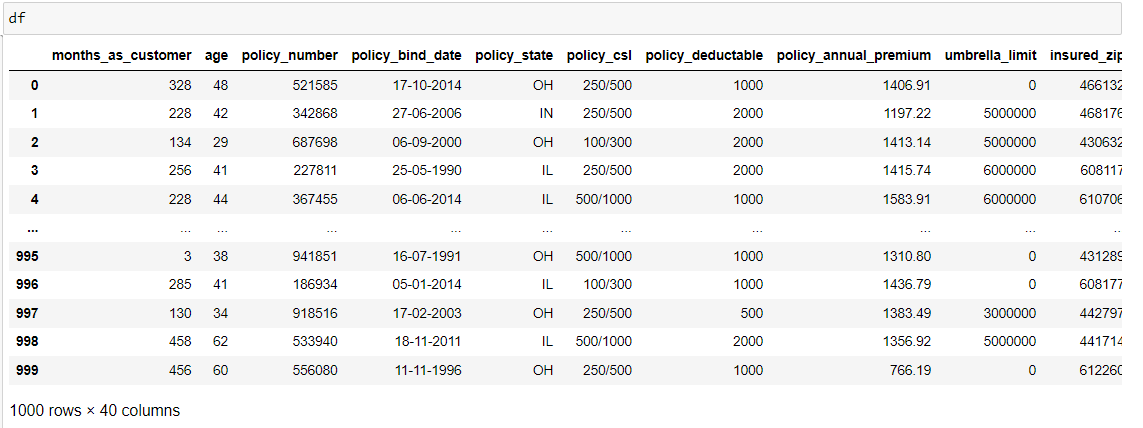
**Importing the Dataset**

Let’s import the dataset first.



Copied the raw data and saved it as a csv file on my local computer after which I imported the entire dataset on this Jupyter Notebook with the help of pandas.

I have imported the dataset which was in “csv” format as “df”. Below is how the dataset looks.



By observing the dataset, we could make out that the dataset contains both categorical and numerical columns. Here "Attrition" is our target column, sine it has two categories so it is termed to be a "Classification Problem" where we need predict whether the features is involved in attrition orr not. As it is a classification problem hence, we will be using all the classification algorithms while building the model that we will see as the data analysis proceeds.

1. **Exploratory Data Analysis (EDA)**

As per the lifecycle of the machine learning model we have already completed points 1 and 2. Now let’s move on to the point 3, 4, 5 and 6 which is the most crucial part of any machine learning model. If we prepare the dataset, analyze it and clean it in the best way possible the better model accuracy we will get, or the model can get over fitted or under fitted. We will discuss further all the steps that are used.

**Data Preparation:**

In this part we will firstly be exploring the data with some basic steps and then further proceed with some crucial analysis, like feature extraction, imputing and encoding.

Let’s start with checking shape, unique values, value counts, info etc.

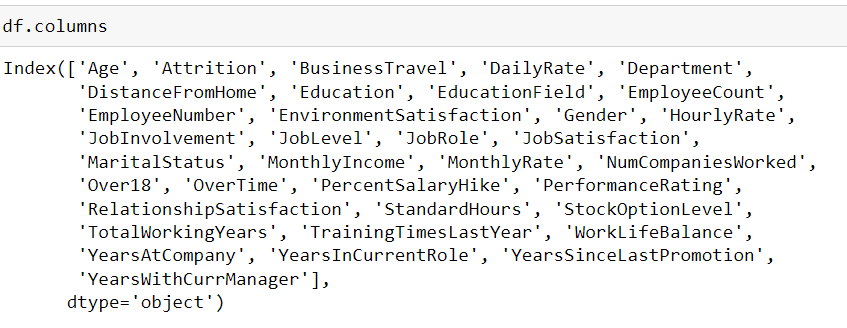
After doing the analysis if we find any unnecessary columns in the dataset, we can drop those columns.

By using ‘df.shape’ we also figured out how many rows and columns we have. We have got the result that we have 1470 rows and 35 columns. PCA can be done, however I decided not to lose any data at this time as the dataset is comparatively small and the first lesson of a data scientist is to preserve as much data as possible hence proceeded will all the data.

Graphical user interface, text, application, email

Description automatically generated

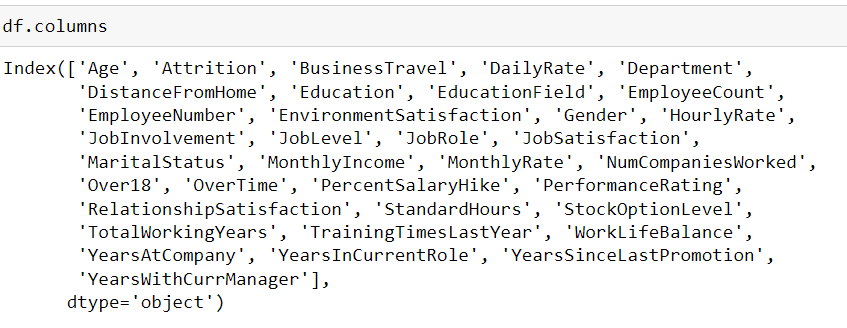
Out of 35 columns 34 are independent columns and remaining one is our target variable “Attrition".



By using ‘df.columns’ all the columns in the dataset are listed as shown above.

**Target label:**

Attrition: Y-YES / N-NO



This gives the information about the dataset which includes indexing type, column Name, non-null values, dtypes and memory usage of the dataset.

Checking the data types using the “df.dtypes” below.

Text

Description automatically generated with medium confidence

Checking NULL VALUES

Graphical user interface, text

Description automatically generated

**Observations:**

* We can see that there are NO null values in the dataset.
* The dataset contains 3 different types of data namely integer data type, float data type and object data types
* We can observe the columns “EmployeeNumber” has unique id values. So, it not required for the prediction so we can drop it.
* Coluumn ‘Over18’ has only one value throughout the dataset so it will not help us in any way so we drop the same column too

Text

Description automatically generated with medium confidence

Text

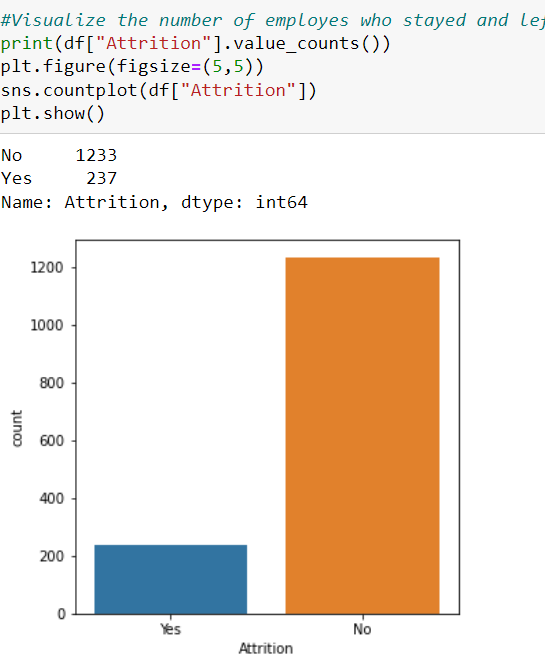
Description automatically generated

**Data Visualization:**

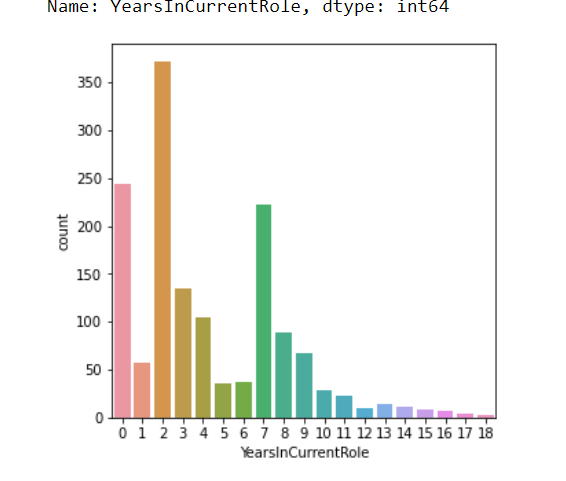
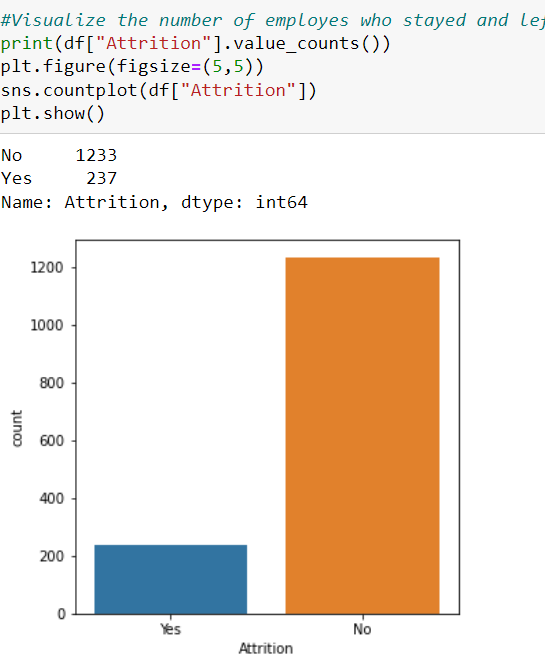
**Preparing for Visualization**

We will look into the categorical and numerical columns so that we can create two different lists and visualize the features accordingly.

**Univararite Analysis (Categorical Columns)**



Attrition count "no" is higher then "yes", which also means that the data is imbalanced, hence we will be required to balance the data

Chart, histogram

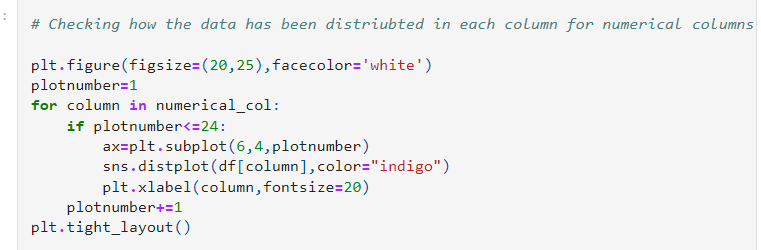
Description automatically generatedChart, bar chart

Description automatically generatedChart, bar chart

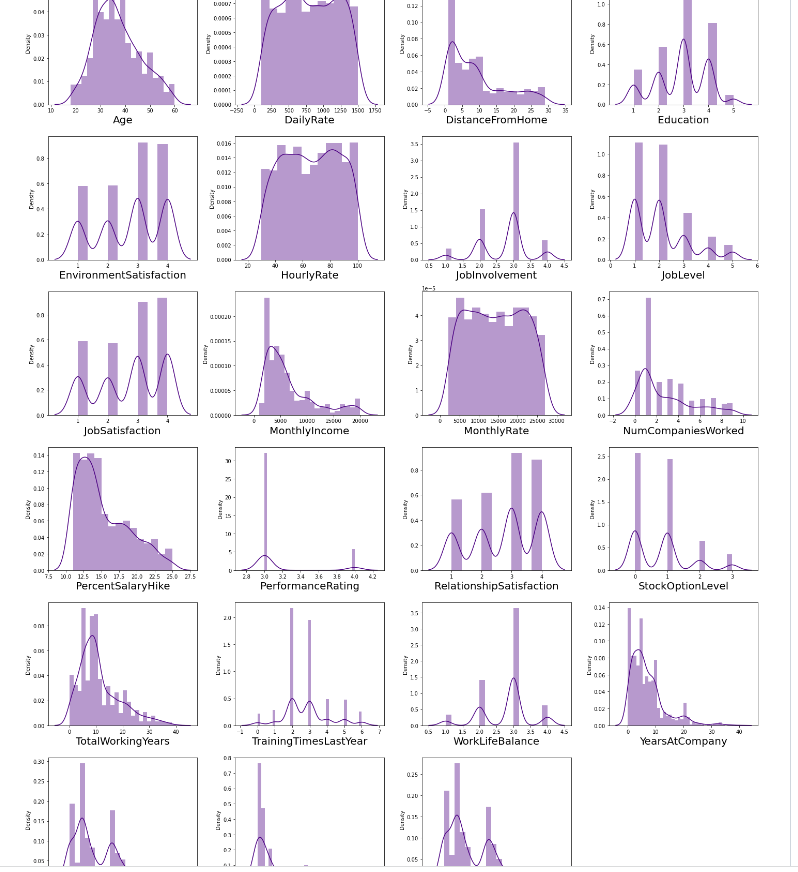
Description automatically generated

Above are the countplot charts of few categorial features.

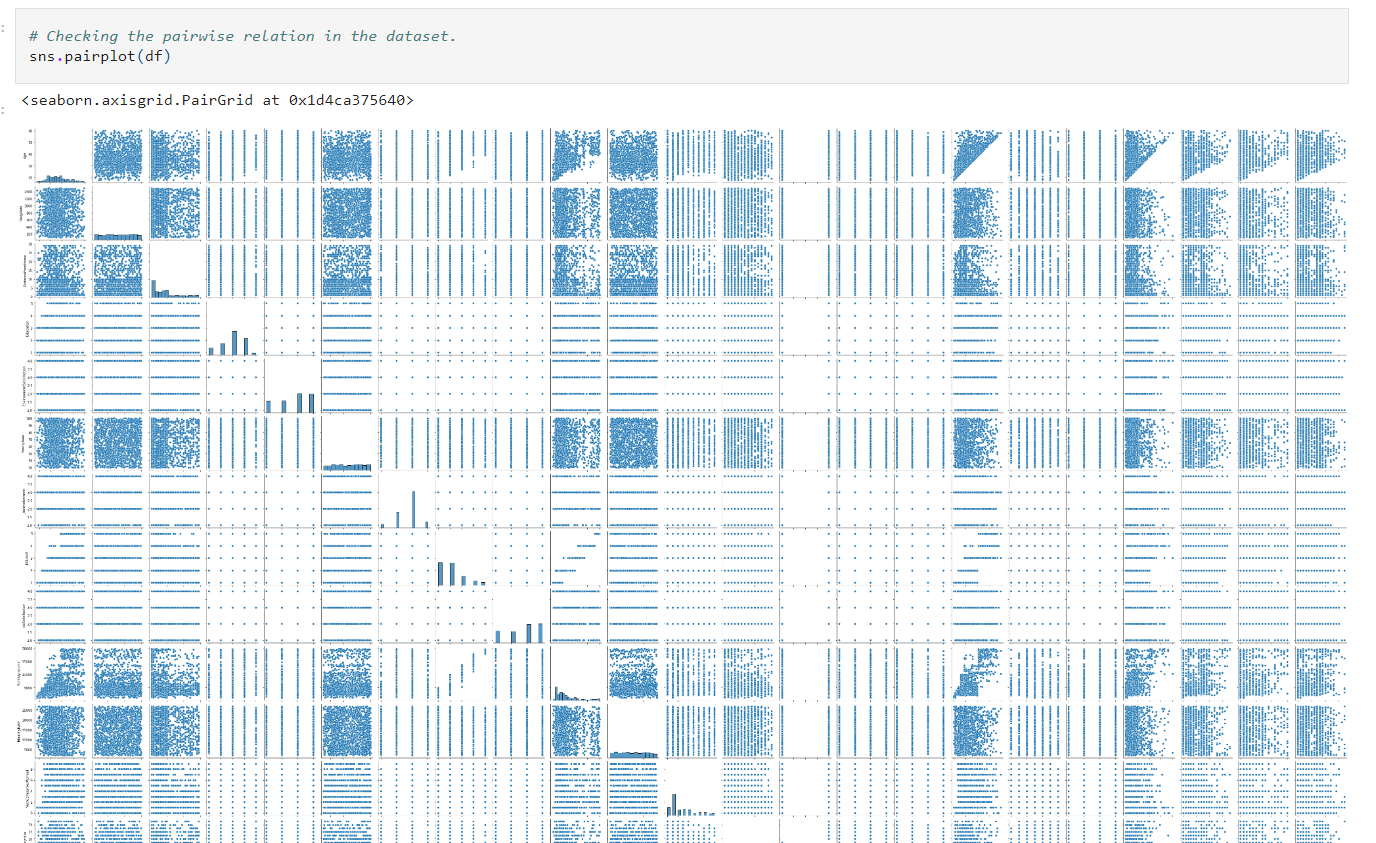
**Checking the Distribution of the dataset (numerical columns)**



From the above distribution plot we can infer that the Age column seems to be normal and there is no skewness in this column. The columns DailyRate, HourlyRate and MonthlyRate are almost normally distributed and these columns have no skewness. Apart from the above mentioned columns, none of the columns are normally distributed and all of them are skewed. We will remove these skewness later.



**Multivariate Analysis:**

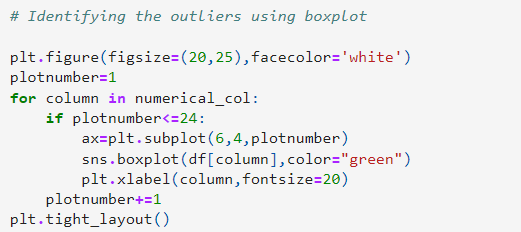


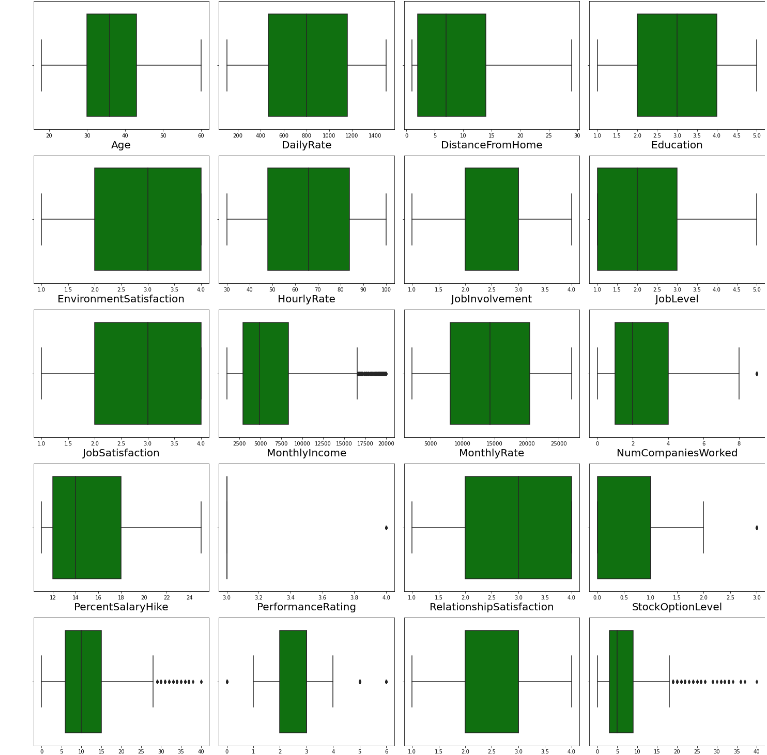
Now we have done with the visualization in order to analyze and understand the data. So, in this EDA part, we have looked into various aspect of the dataset, like looking for the null values and imputing, extracting date time, observing the value counts and doing the feature extraction etc.

Now we will be performing Data Pre-processing by identifying the outliers and removing them. Along with it we will also look for the skewness of the dataset and remove the skewness.

1. **Pre-Processing Pipeline**

**Identifying the outliers**

****



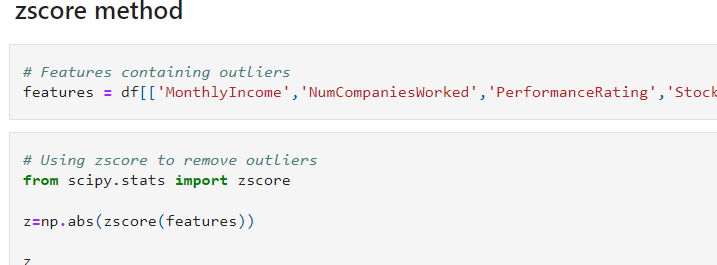
We have used box plot to identify the outliers and we can find the outliers in the following columns:

We have created a box plot visual for all our integer datatype columns to check for outliers. We do see some of the columns where there are presence of outliers and we will need to treat it accordingly.

* MonthlyIncome
* NumCompaniesWorked
* PerformanceRating
* StockOptionLevel
* TotalWorkingYears
* TrainingTimesLastYear
* YearsAtCompany
* YearsInCurrentRole
* YearsSinceLastPromotion
* YearsWithCurrManager

All the above columns show visible outlier details.

These are the numerical columns which contains outliers. Removing the outliers in these columns using Zscore method.



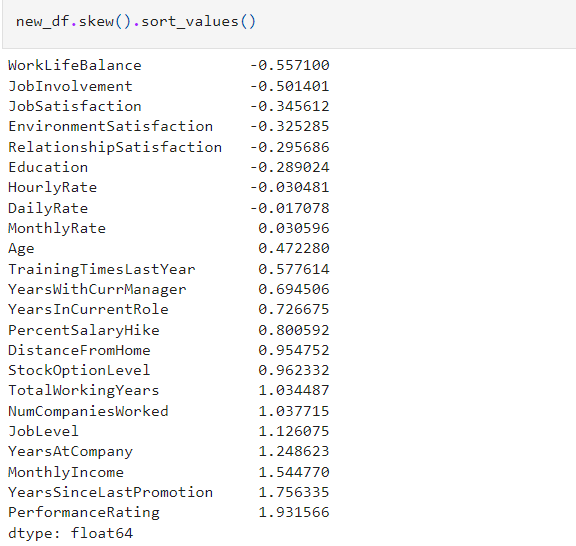
**Percentage data loss:**

Graphical user interface, text, application

Description automatically generated

After removing the outliers, we are checking the data loss percentage by comparing the rows in our original data set and the new data set and 5.6% data loss is in the acceptable range.

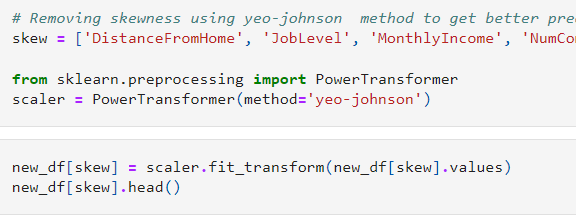
**Checking skewness in the data:**

****

The columns containing skewness more than +0.5 and -0.5 are

DistanceFromHome JobInvolvement JobLevel MonthlyIncome NumCompaniesWorked PercentSalaryHike PerformanceRating StockOptionLevel TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany YearsInCurrentRole YearsWithCurrManager YearsSinceLastPromotion Here PerformanceRating and WorkLifeBalane are categorical columns, so no need to remove skewness in these columns.

Here, I am using the yeo-johnson method to remove the skewness.

****

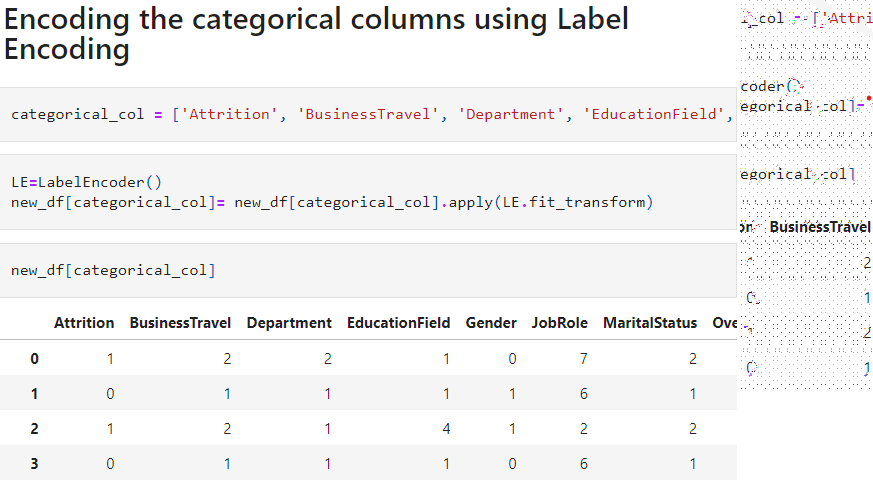
**THEN WE CHECKED THE SKEWNESS AND FOUND THE SKEWNESS WERE REDUCED TO ACCEPTED VALUES.**

After removing skewness, The data distribution looks better after removing the skewness compared to the previous data.

Now we have completed our analysis of the dataset and also cleaned the dataset so that we can build a good model.

However, we have seen that the dataset has both numerical and categorical data. The model only understands numerical data; hence we will encode the categorical data. Also, we have seen that there can be some multicollinearity, that we will see through a heatmap and also further remove it. Again, we have also seen that the target variable is imbalanced, hence we will fix it by oversampling. And finally, we will scale the data so that it is ready to be trained and tested.

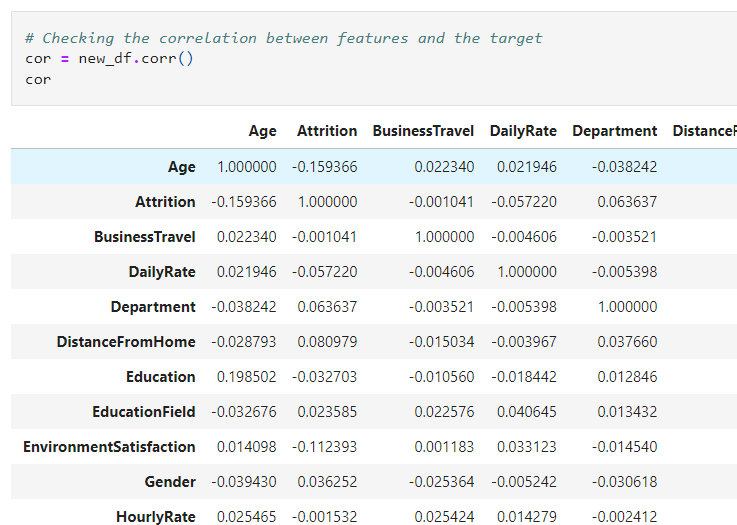
**Encoding the categorical columns using Label Encoding**



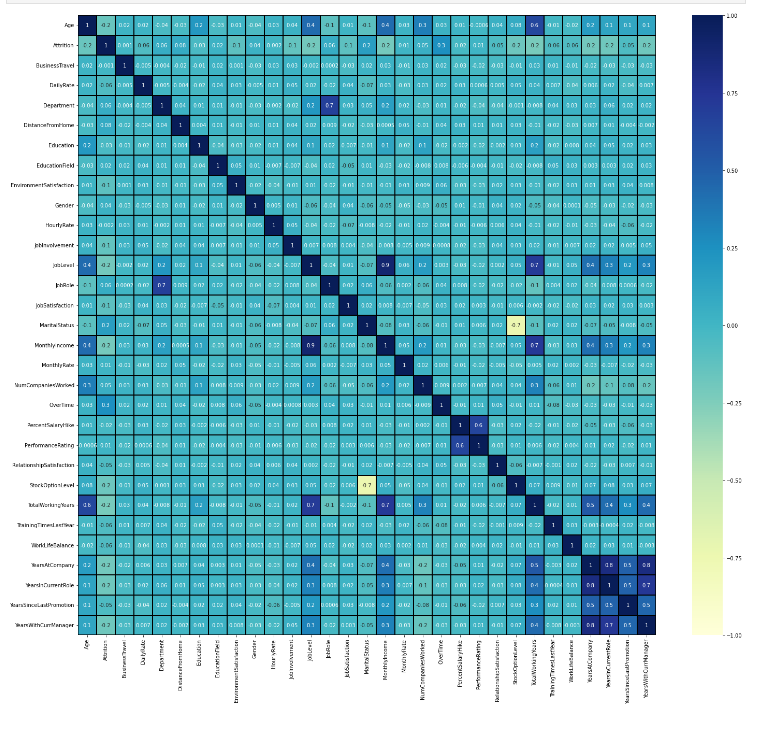
Now we have encoded the dataset using label encoder and the dataset looks like above.

Moving forward, to check the correlation between the feature and target and also the relation between the features using the heatmap.

**CORRELATION**

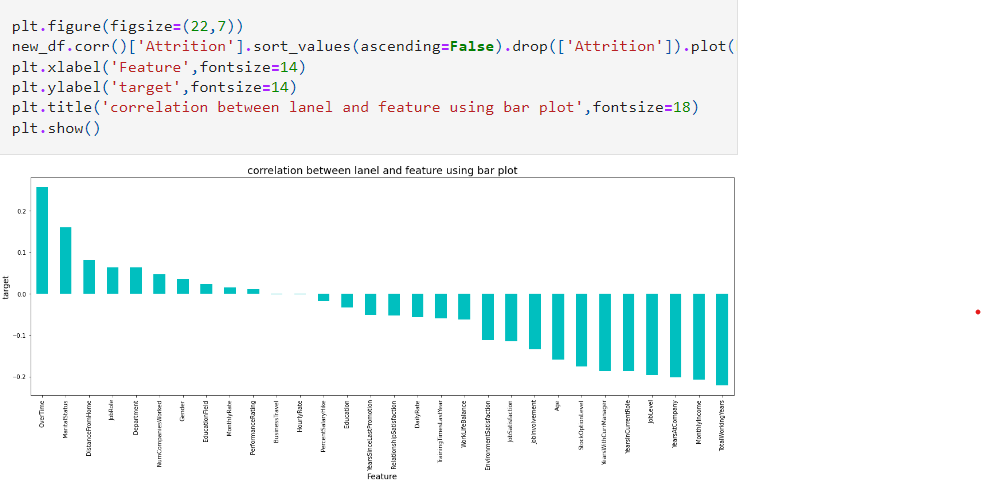
****

**Heatmap of correlation**

****

This heatmap shows the correlation matrix by visualizing the data. we can observe the relation between one feature to other.

This heat mapcontains both positive and negative correlation. We can notice that the target variable "Attrition" has very less correlation with the feature columns. The columns BussinessTravel and HourlyRate have no relation with the target so we can drop these columns. The only columns StockOptionLevel and MaritalStatus have correlation with each other. Apart from this here is no multicollinearity problem, so no need to worry much.

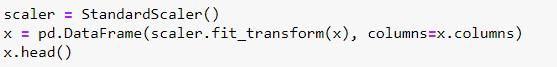
**CORRELATION USING BARPLOT**

Here we can notice the columns BusinessTravel and HourlyRate have very less corrrelation with the target so we drop those columns.

**Splitting the dataset into Features and Target**



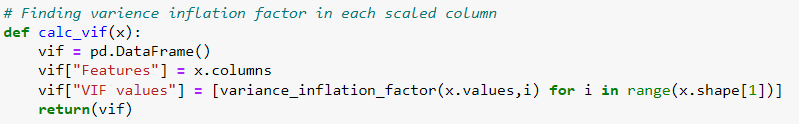
## Feature Scaling using Standard Scaler

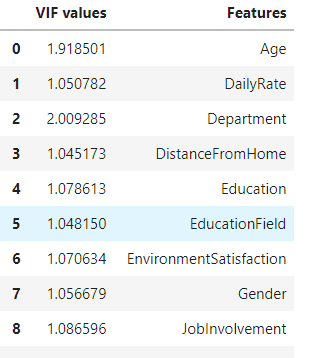


The data has now been scaled.

In the heat map we have found some features having high correlation between each other which means multicollinearity exists. So, let's check the VIF value to solve multicollinearity problem.

**Checking Multicollinearity**

****

****

None of the columns have the VIF value above 10 which means there is no multicollinearity issue.

**SMOTE (BALANCING THE TARGET)**

**Text

Description automatically generated**

**Chart, bar chart

Description automatically generated**

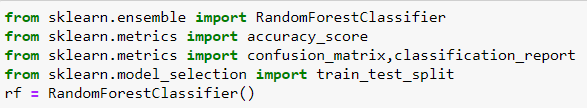
As we have treated the oversampling issue using SMOTE, now we can proceed with modelling.

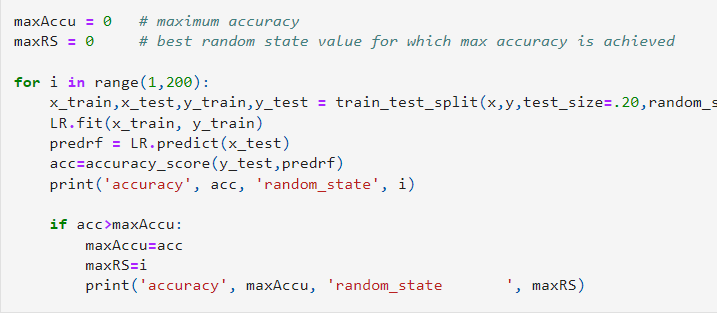
1. **Machine Learning**

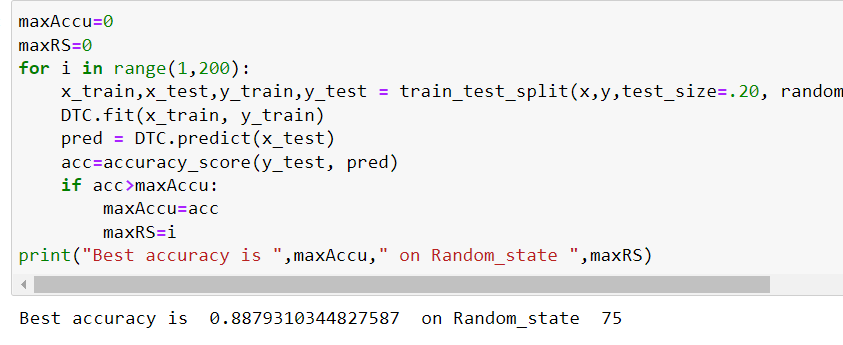
**Finding best random state**

let’s find the best random state in which we can build the model*.*

(Random state ensures that the splits that you generate are reproducible. Scikit-learn use random permutations to generate the splits. The random state that you provide is used as a seed to the random number generator. This ensures that the random numbers are generated in the same order.)

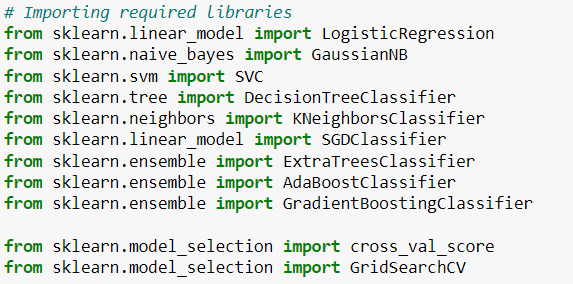




****

.

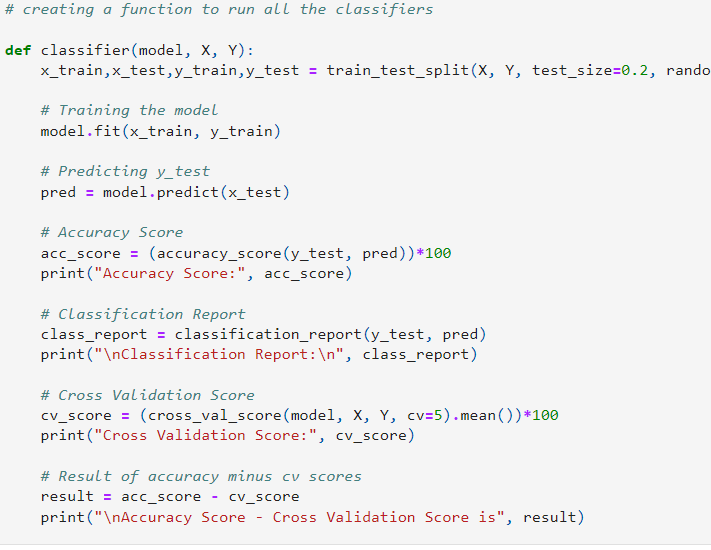
We import all the required libraries as shown below.

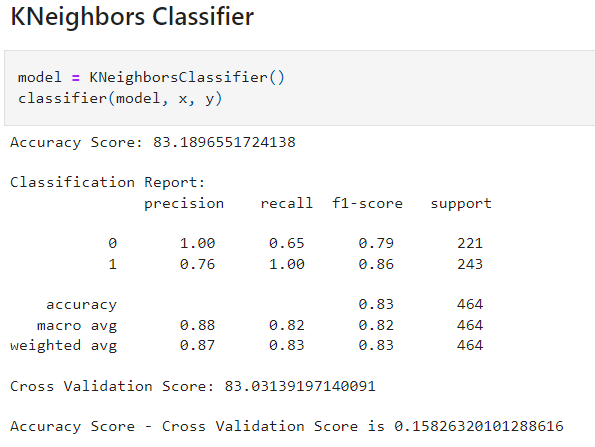
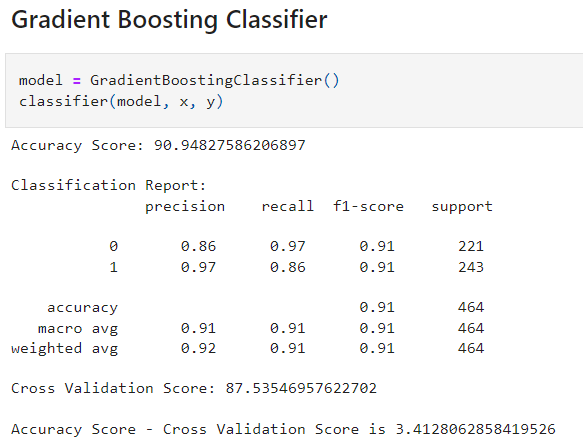
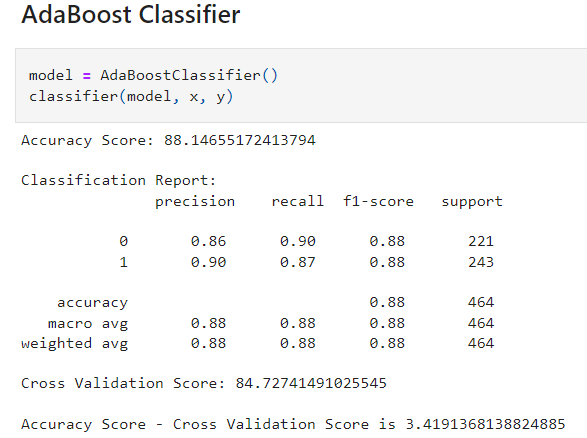
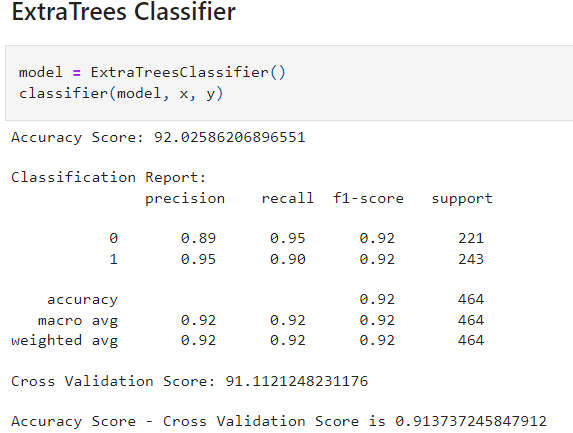
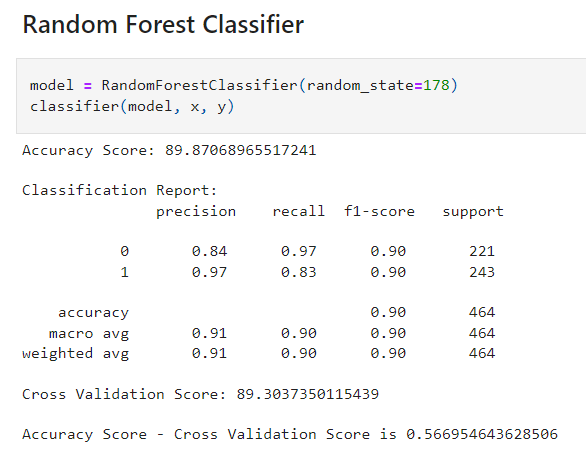
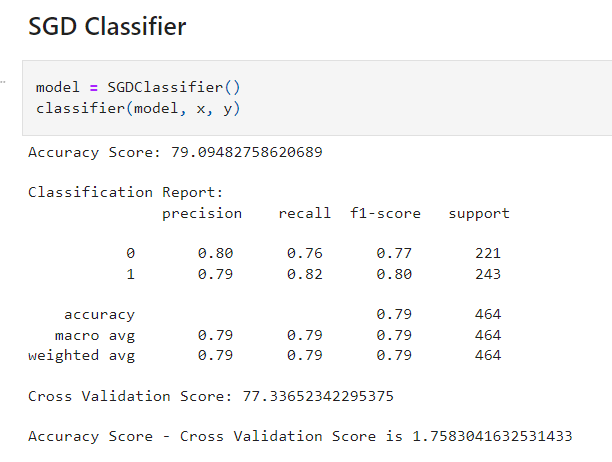
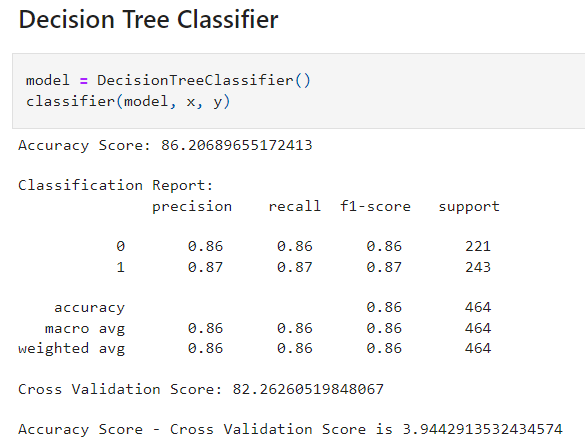
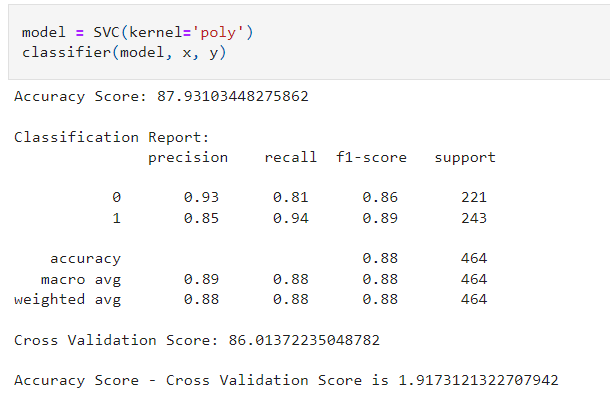
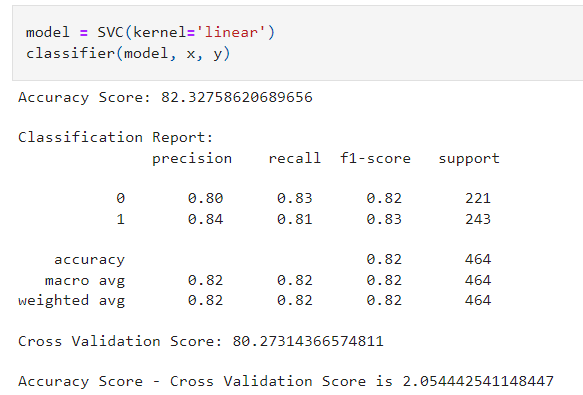
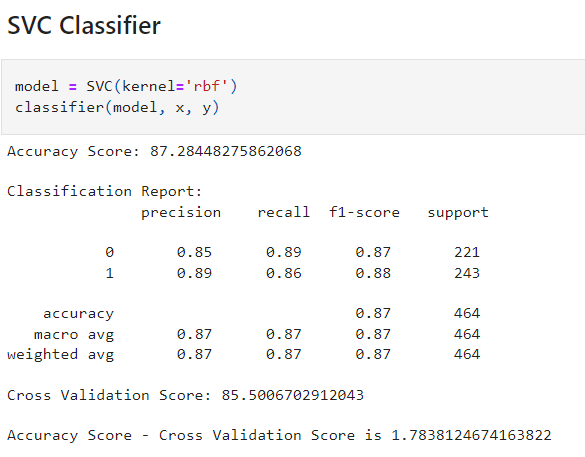
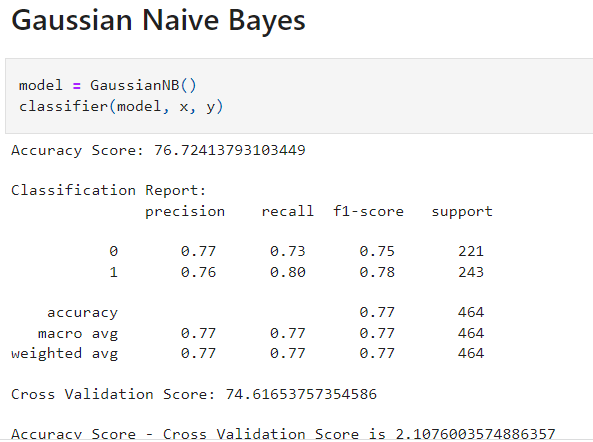
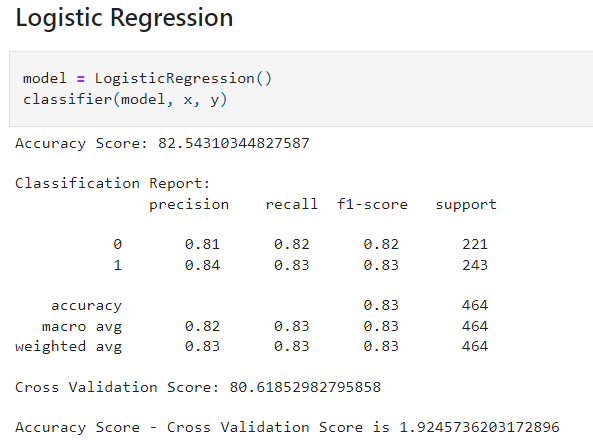


I have created a function which can be used for all the classifier machine learning algorithms and get the Accuracy score, Classification Report, Cross Validation Score and Accuracy Score - Cross Validation Score for comparison between all the algorithms for finding out the best suited algorithm for Hyper Parameter Tuning.

**Machine Learning Algorithms**

We will run various algorithms and check which has the highest Accuracy score, Cross Validation Score and (Accuracy Score - Cross Validation Score) for comparison between all the algorithms using the below definition for all classifiers.

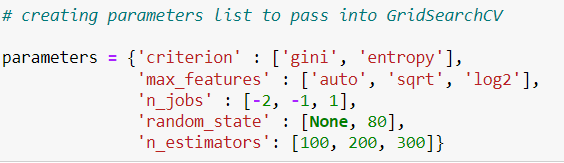




Referring all the above algorithms we can see that ExtraTreesClassifier gives the best results since the (Accuracy Score - Cross Validation Score) is the least comparing others while having higher Cross Validation Score and the highest Accuracy Score comparing all the other models.

Now, that we have found the best fit model, lets perform Hyper Parameter Tuning to improve the performance of the model.

**Hyper parameter tuning**

Creating a list of parameters to pass into the Grid Search CV.

Running Grid Search CV for ExtraTreesClassifier at cv = 5

Text

Description automatically generated

Getting the list of the best parameters from Grid Search CV.

Text

Description automatically generated

Here we have got the best parameters, and we will build our final model using these parameters.

Graphical user interface, text, application

Description automatically generated

The Final Accuracy Score of the final Model.

A picture containing text

Description automatically generated

We successfully performed the Hyper Parameter Tuning on the Final Model.

The Final Cross Validation Score of the final Model.

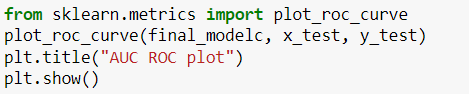
Graphical user interface, application, Word

Description automatically generated

We got final accuracy score of 93.10% and Cross Validation Score of 91.75% which is really good.

**AUC ROC curve**

Plotting and AUC ROC curve for the final model.



Graphical user interface

Description automatically generated with medium confidence

So here we can see that the area under curve is really good for this model.

We got final accuracy score of 93.10% and Cross Validation Score of 91.7% and also AUC score is 0.98 which is really good.

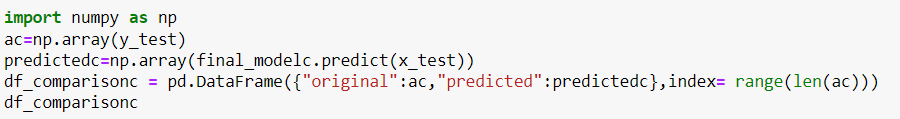
**Saving the model in pickle Format**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**Prediction Conclusion**

We will predict the "Attrition" target column using the final model sending the “x\_test” set for predicting the “y\_test” and then compare the original “y\_test” and the predicted “y\_test” in a dataframe.



**A picture containing diagram

Description automatically generated**

Hence predicted the "Attrition” using the final Model and presented it as a data frame to compare the predictions.

Saving the comparison file as a csv file.

Graphical user interface

Description automatically generated with low confidence

Saved the file as a csv for future reference purposes.

1. **Concluding Remarks**

In the beginning of the blog, we have discussed about the lifecycle of a Machine Learning Model, you can see how we have touched based on each point and finally reached up to the model building and made the model ready for deployment.

Let’s take a quick recap on all the steps that we went through starting from understanding the Problem Definition then going through the Data Analysis and EDA processes. We went through the necessary Pre-processing Data steps before the final Building Machine Learning Models step came into picture.

The Insurance industry area needs a good vision on data, and in every model building problem Data Analysis and Feature Engineering is the most crucial part.

You can see how we have handled numerical and categorical data and also how we build different machine learning models on the same dataset. And using hyper parameter tuning we can improve our model accuracy. Hence, we ended up with a prediction accuracy of 91% after all the above steps are completed.

Using this machine Learning Model, we can predict how these features are responsible for Attrition of Employees.

**Thank You**