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



**Introduction:**

This article is on HR analytics, a project on machine learning, which we will be doing with the help of python programming language. 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. The topics that we will be covering are:

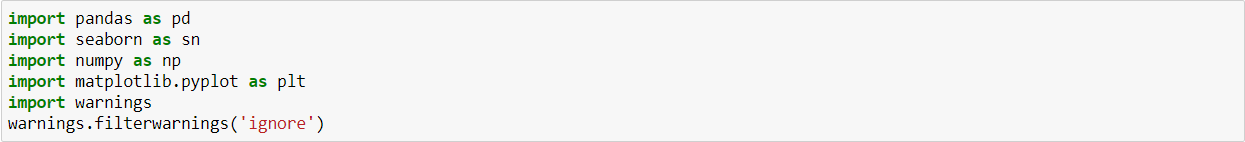
1.      Problem Definition  
2.      Data Analysis  
3.      EDA Concluding Remarks  
4.      Pre-processing Pipeline  
5.      Building Machine Learning Models  
6.     Concluding Remarks

**1. Problem Definition:**

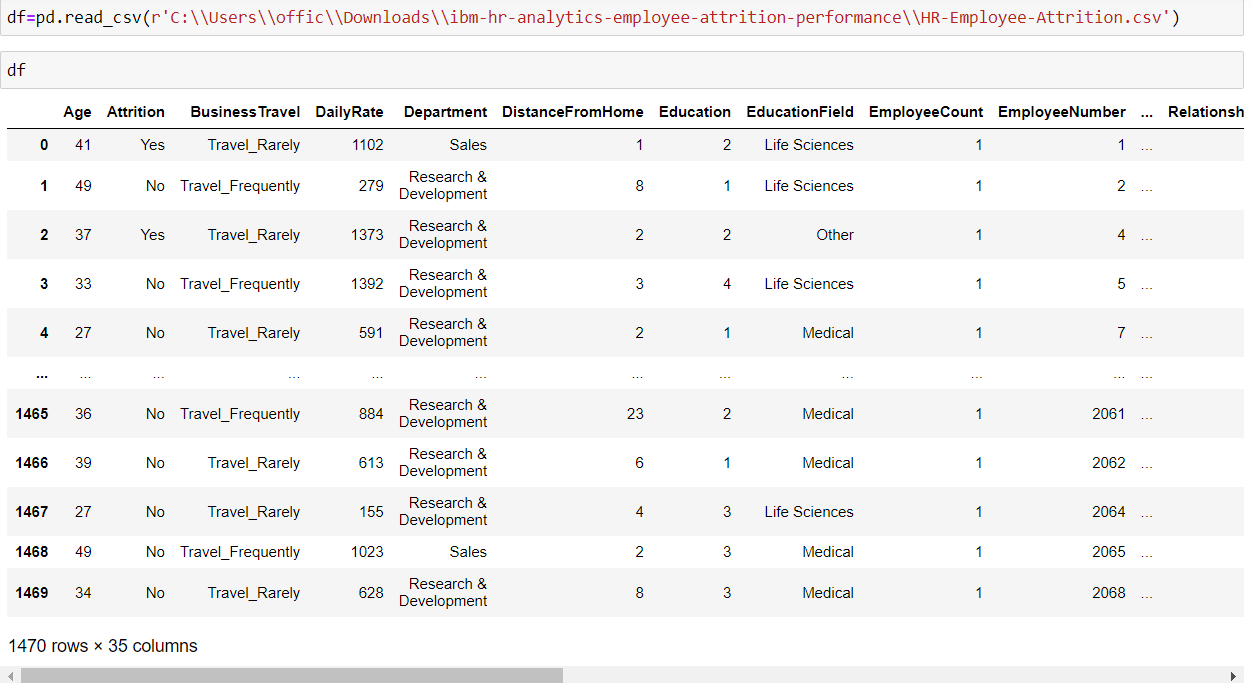
Attrition in human resources refers to the gradual loss of employees’ overtime. In general, relatively high attrition is problematic for companies. HR professionals often assume a leadership role in designing company compensation programs, work culture, and motivation systems that help the organization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

The purpose of this project is to predict the attrition of each employee, to find out which employee is more likely to leave the organization. This type of prediction will help the organizations to prevent attrition and when to hire new people.

**2. Data Analysis:**

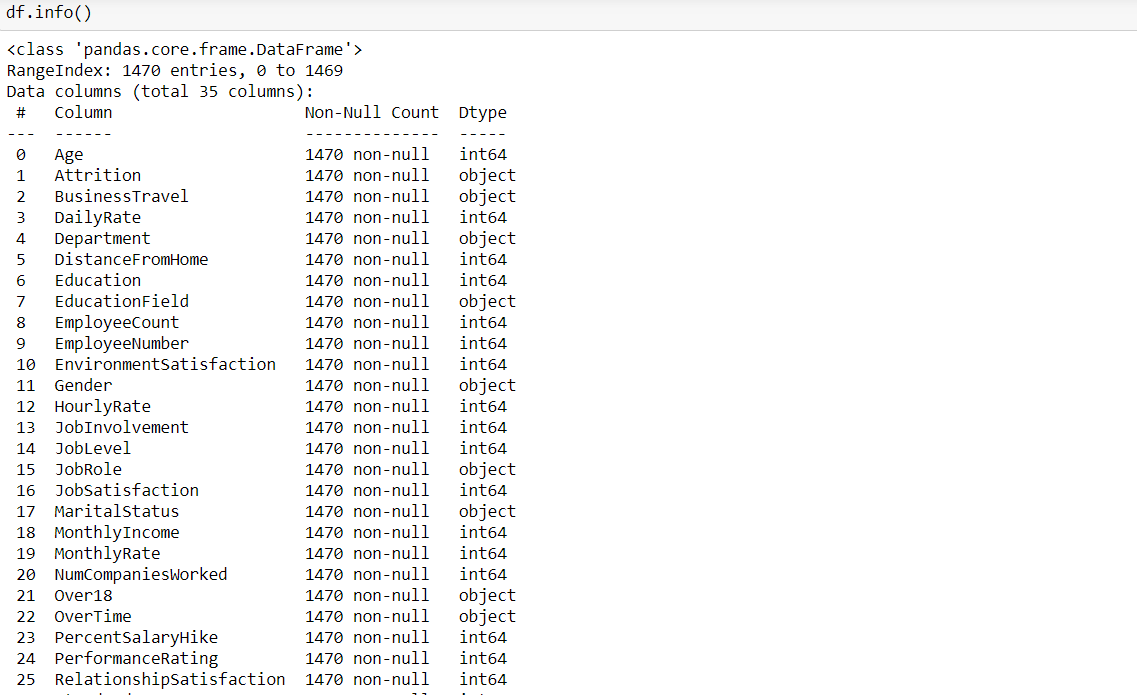
****

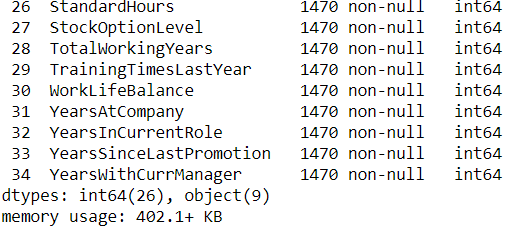
* Firstly, importing all the important libraries that are required for the EDA and feature engineering process. Next step would be to import the dataset and the python notebook and analyse the dataset carefully.



* After importing the dataset, we can observe that there are 1470 rows and 35 columns in this dataset, which is not that huge. The column names are related to the employees personal and professional information like Age, Department, Distance from home, education etc.
* Here our target variable or column is Attrition, and the column consists of either yes or no as the content. This clearly depicts that this will be a classification problem where we have to predict the Attrition of each employee.
* We can see the obvious observation from the dataset, dataset has integer and object datatypes, we need to encode the object datatype columns for the machine learning purpose.

**3. EDA Concluding Remarks:**

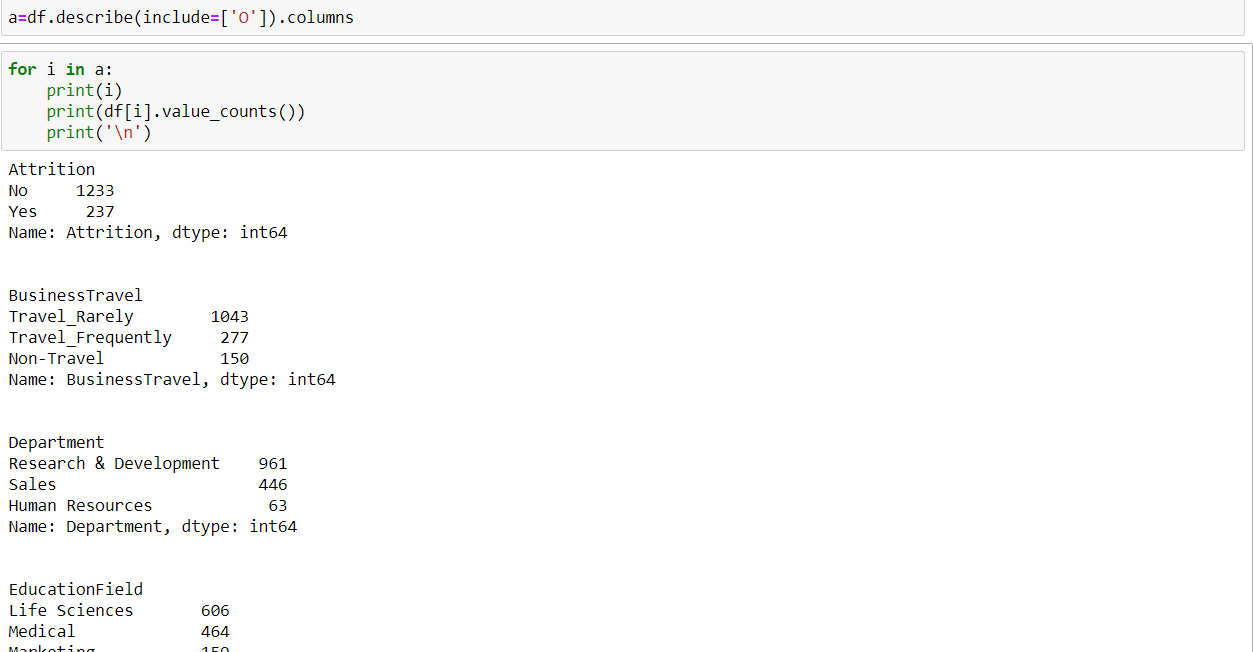
****

****

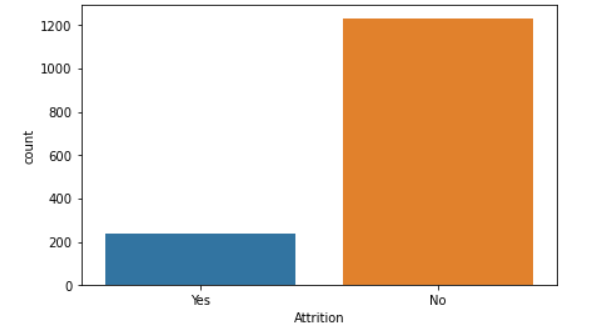
* We can see from the above screenshot, there aren’t any null values present in any of the columns.
* We can see the column names and their respective datatypes.



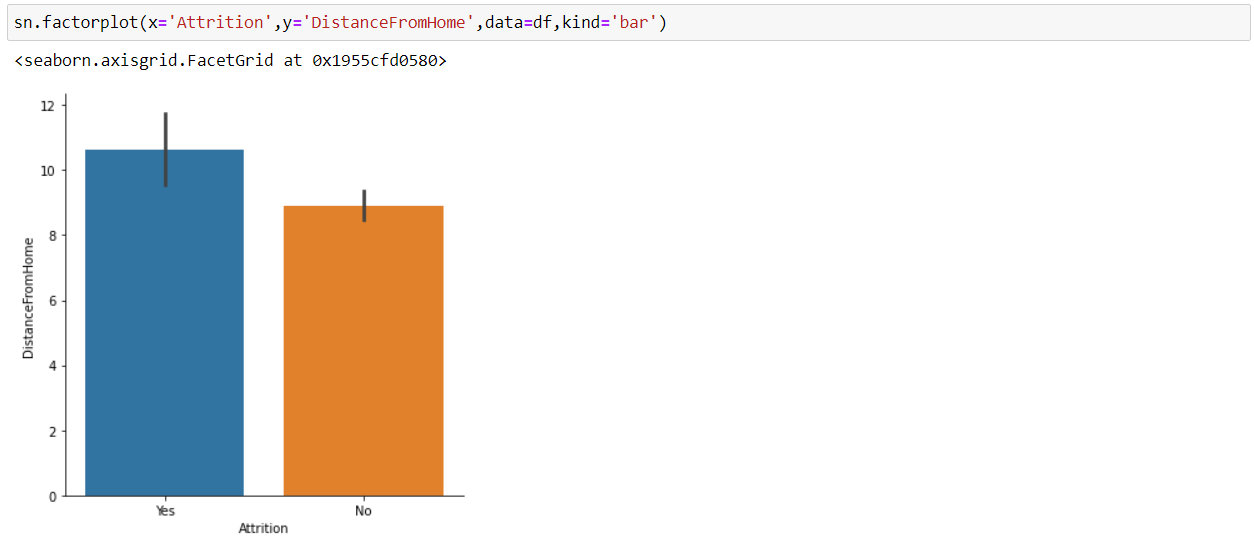
* After seeing the unique values of the above columns, it consists of only one value or all are unique value, it will not help us in machine learning, hence we decided to drop the columns.
* After this we checked the unique values present in each column, we will need to encode them in the upcoming steps.



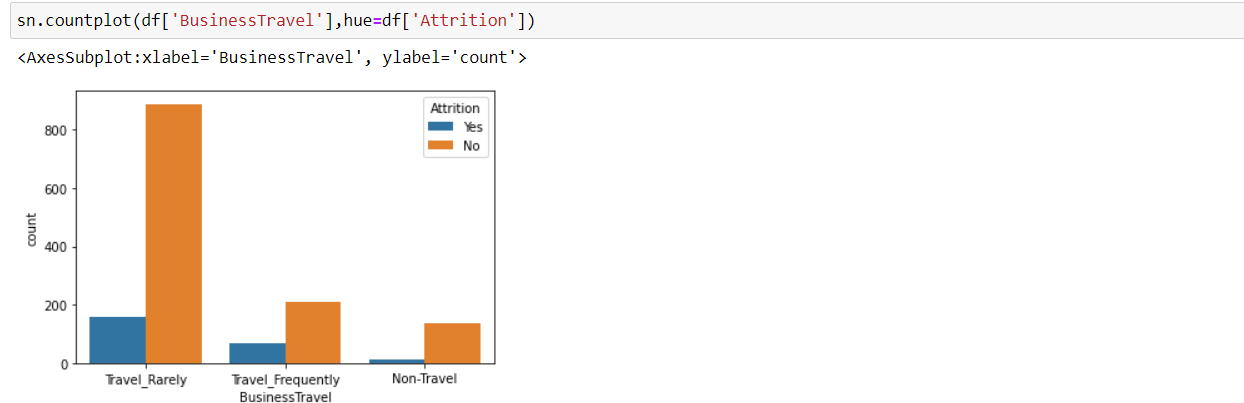
* All the object datatype columns are saved a variable and then with the help of for loop we are observing the unique values of each column.
* We analysed that majority of the object datatype column are in the classification format, we will be encoding them.
* Now we will be analysing the target column Attrition, whether it is imbalanced or not, with the help of countplot.



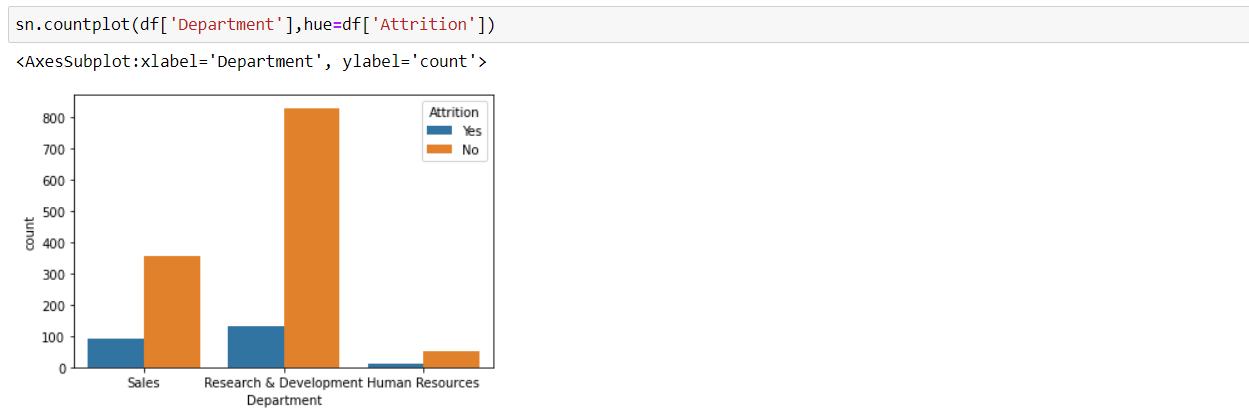
* As we can see the target column is highly imbalanced, we need to balance this later.
* Now we will see the relationship of our target column which is attrition with other columns.



* We can see from the graph, employee who travel far from home, leave their jobs.
* Employees who travel more than 10kms has the probability of leaving their job.

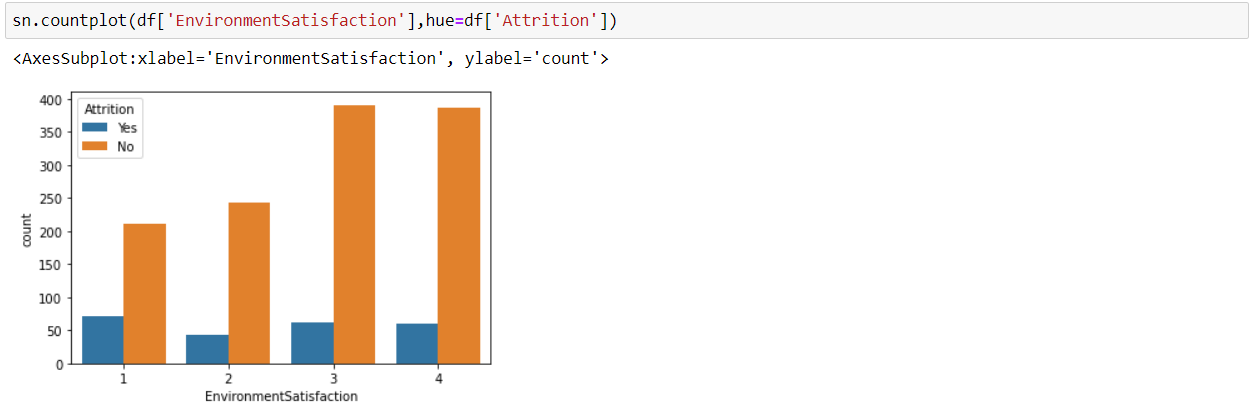


* Employees who travel frequently are more likely to leave their jobs whereas people who travel rarely are more likely to stick with the jobs.

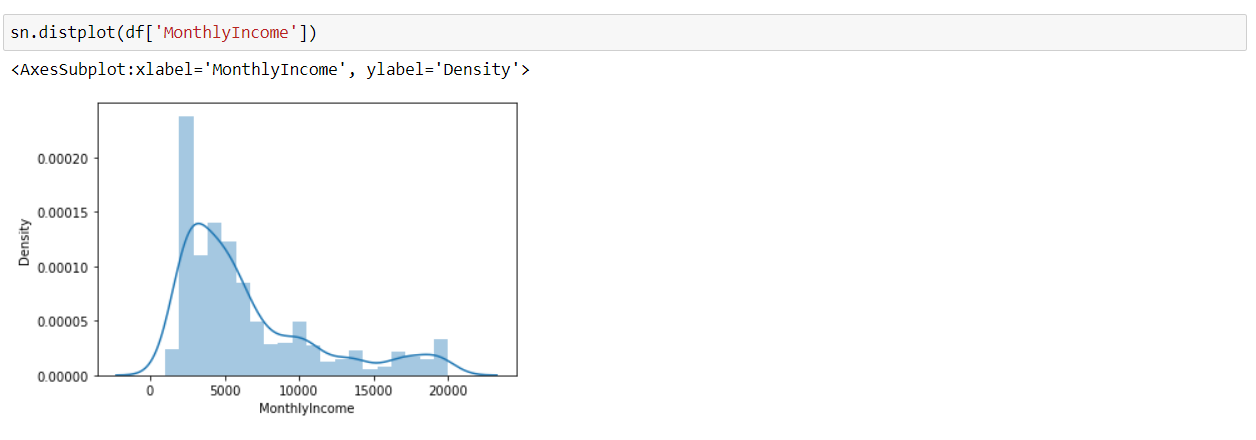


* These are the three departments that people work in the company,

in which Research and Development department shows the highest rate of no attrition whereas Human Resources department shows the lowest rate of no attrition.

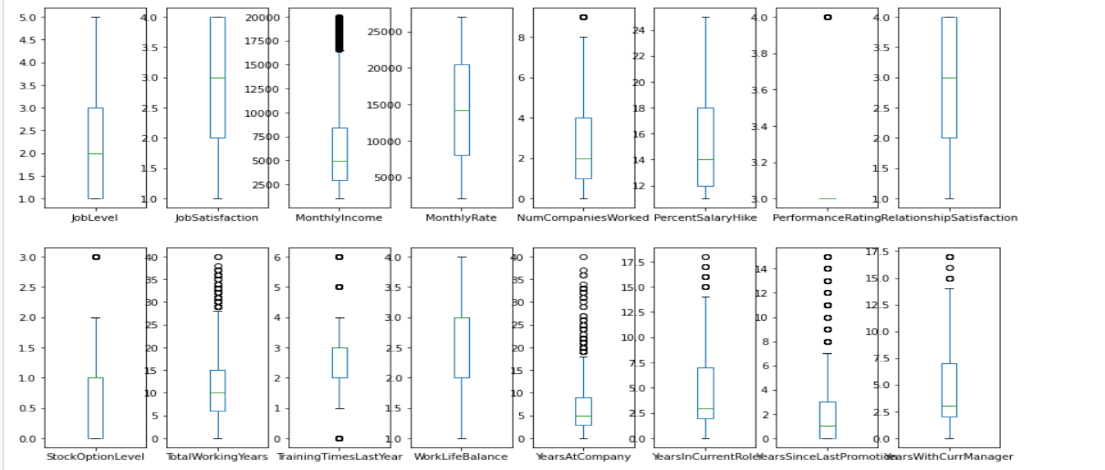


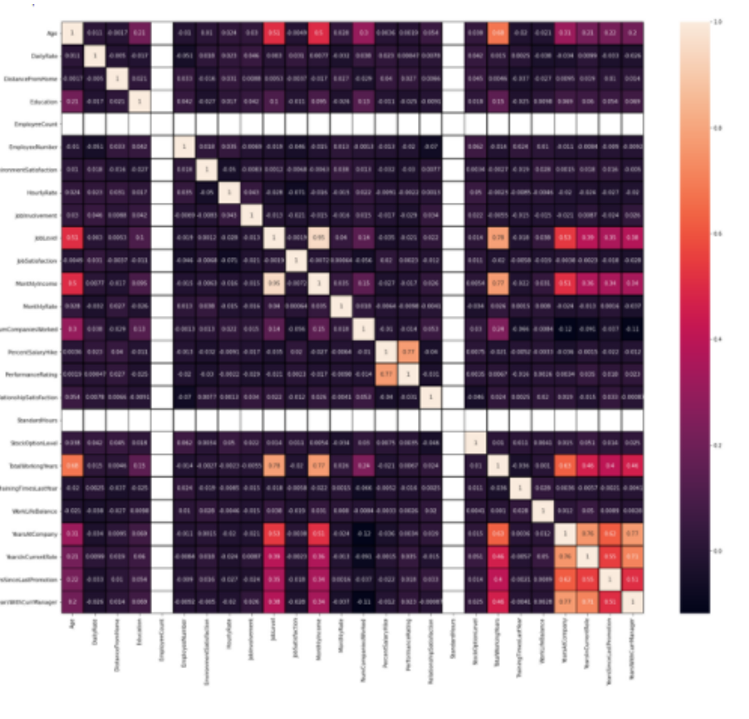
* As we can see from the graph, the plot is between the environment satisfaction and attrition where 4 is considered to be highly satisfied and 1 being not satisfied.
* Employees who are highly satisfied with the environment are not likely to leave their jobs, whereas people who are less satisfied from the environment i.e., 2 and 1 are more likely to leave their job.
* We have a fair bit of idea of which attributes will be helpful, to predict the attrition of an employee.
* Now we need the check the distribution plot of monthly income column, and see whether the graph is skewed or not.



* Clearly it can be seen from the graph, the distribution is skewed, there can be outliers present, we need to check it later. We need to have a look at box plot, so that we have a clear idea about outliers.
* It can be seen clearly from the box plot that there are several columns like MonthlyIncome, TotalWorkingYears, yearsatcompany etc., have outliers present. Outliers are represented as black dots present in the box plot.





* Now we will check for the correlation of the attributes with the target column ‘Attrition’ with the help of heatmap.
* Heatmap is nothing but a two-dimensional graphical representation of data where the individual values that are contained in a matrix is represented as colours.
* 1 represents the high positive correlation whereas 0 represents the weak negative correlation.
* After seeing the heatmap we can deduce that which attributes are strongly correlated and which are not.
* We did all the possible EDA steps, now we need to do all the essential pre processing steps for the machine learning purpose and to get a better accuracy of the model.

**3. Pre-Processing Pipeline:**

* Firstly, I dealt with the skewness of the column, monthly income, we need to get the skewness closer to 0 or in the range of -0.5 to +0.5. We intend to have a normal distribution of the graph, normal distribution helps us achieving a better accuracy and it also depicts that all the unwanted values are removed or reduced.

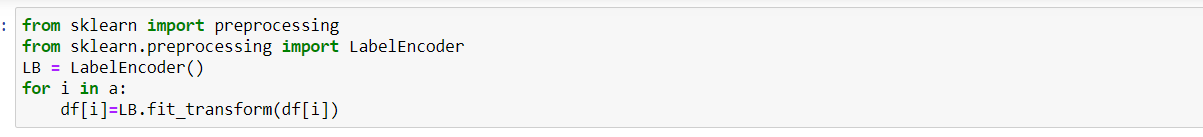


* There are many techniques to remove the skewness, but I chose log transformation technique, it is considered to be a powerful technique in order to remove or reduce the skewness.
* After the removal of skewness, I looked into the performance rating column unique values, I can see that there are two unique values present in it, 3 and 4, I replaced them with 0 and 1 just to avoid any unwanted noise during machine learning.



*Encoding*

* After dealing with skewness and removing the noise, I have to **encode** the object datatype column. We encode the object datatype column because machine understands numerical data, and to perform machine learning we need to encode categorical data into numerical data.



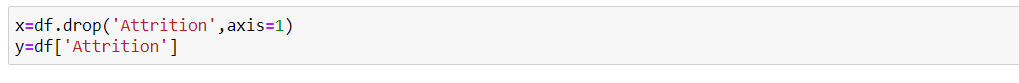
* Here we are using Label Encoder to encode the categorical columns. This method used on ordinal data and also used when we have large number of unique values in single attribute.

*Handling Outliers*

* Now we need to handle our outliers, to remove the outliers we will be using two methods: -

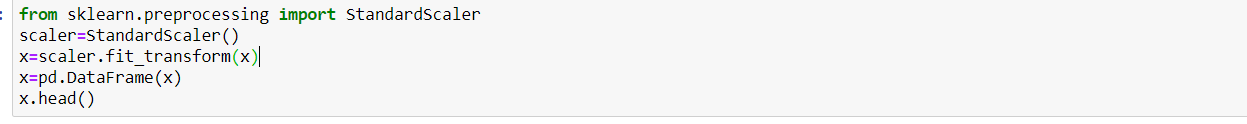
1. Zscore
2. Inter Quartile Range

* We used both the methods, but after seeing it we can see the data loss is more than 40% which is considered as huge. We cannot afford to lose that much of precious data as it can hamper our machine learning model, so we scrape the idea of removing the outliers.
* Now we will split our independent variable x and our target variable y for further machine learning.



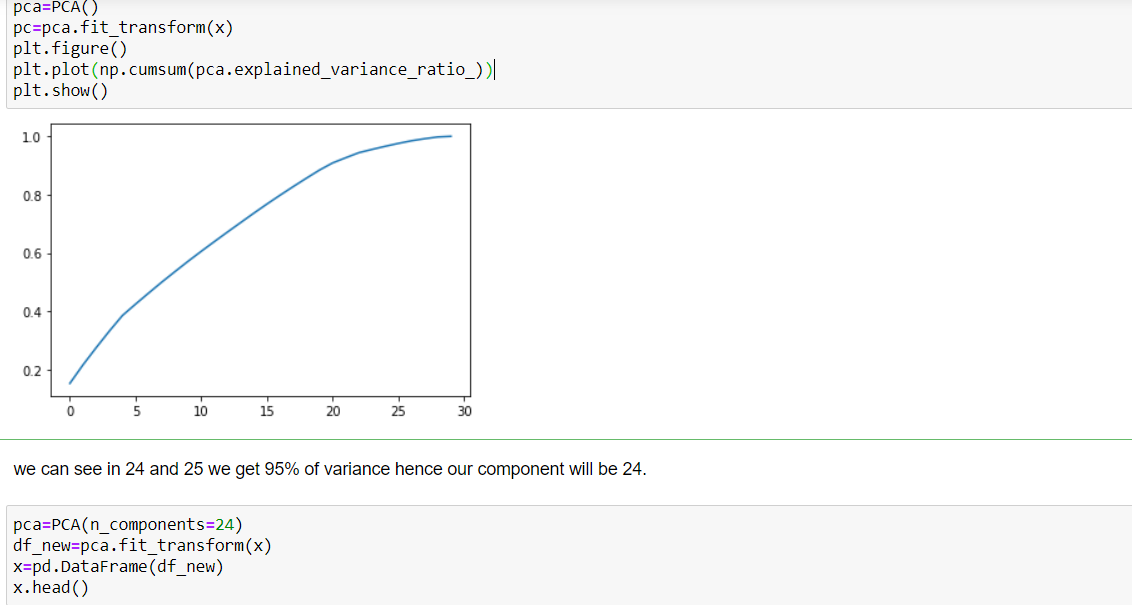
*Scaling*

* After splitting the data, we are scaling our data by using standard scaler method. Scaling is a method to normalize our independent variables. Scaling is essential for machine learning algorithms that calculate distance between data. For instance, most of the classifiers calculate the distance between two points by the distance. If one of the features has large value then distance consider this particular feature. Scaling is necessary, where large and small values present in variables.

**

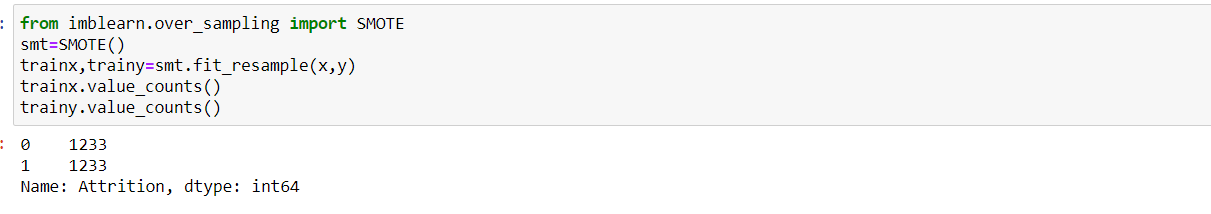
* Standard deviation method converts data with a mean value of 0 and standard deviation of 1.

*PCA:*

* Principle Component Analysis is a technique of reducing the dimensionality of variables. It is done when we are dealing with large dataset, here we can see there are 30 columns involved, we can reduce the number of columns with the help of PCA technique. It makes easier for the computer to do the machine learning.
* We have to select that, to how many columns we need to reduce our dataset, the no. of columns or n\_components are decided on the basis of on which component we are getting the variance more than 95%. From the graph we can see that on 24 components we are getting 95% variance and above. Hence, we reduced our number of columns to 24.

*Sampling:*

* Above in the graph we saw an imbalance in our target variable ‘Attrition’. We have to balance the count of yes and no and we will be doing that with the help of SMOTE technique. We will increase the number of yes values with the help of over sampling.
* As we can see the count of 0 and 1 i.e., yes and no is equal now

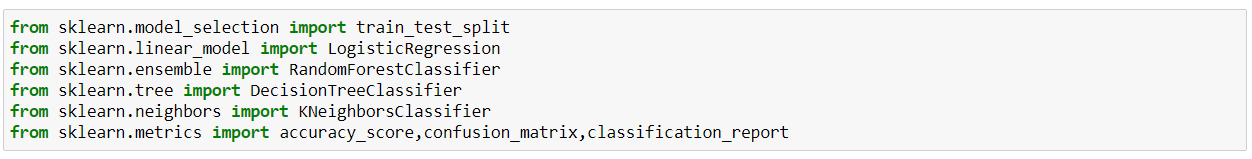


**4. Building Machine Learning Models:**

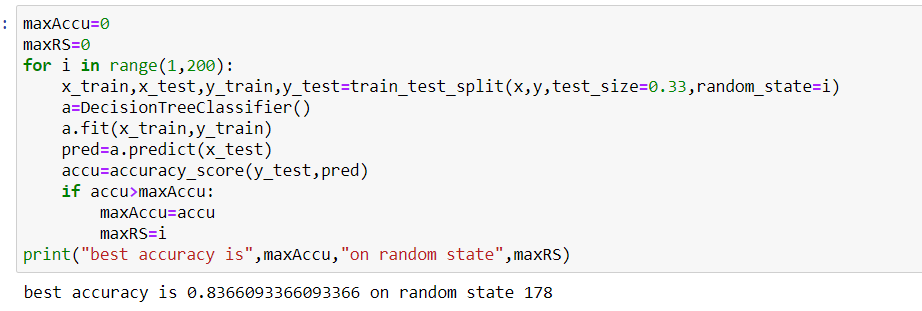
* Now we will be training and testing different models after doing all the data cleaning steps, I will be choosing 4 models for the training and testing purpose: -

1. Logistic Regression
2. Random Forest Classifier
3. KNeighbors Classifier
4. Decision Tree Classifier

* We will be importing all the necessary libraries for building different types of models.



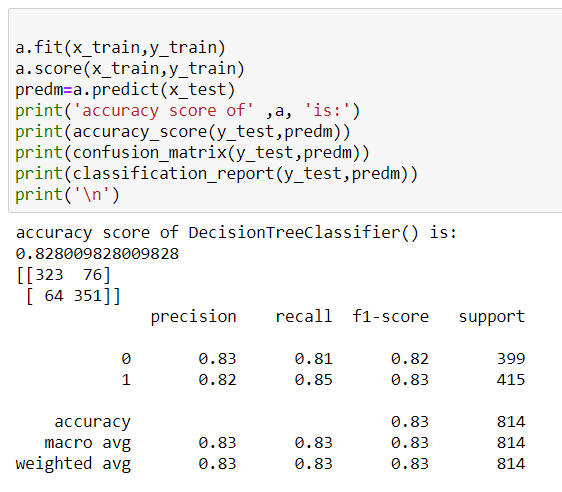
* After importing the libraries, we will be selecting the best random state for every model.



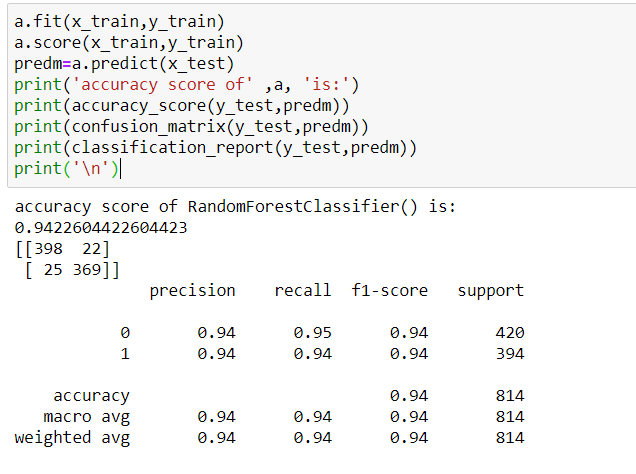
* We first created two variables to store the score and the random state and assigned value 0 to them. After that we took a range of 1 to 100 for the random state and created the train test split and then performed model fitting using each random state. We then kept the highest score in one variable and the random state for that score in another. And finally, we printed the score and the random state. We will be doing this step for every model.



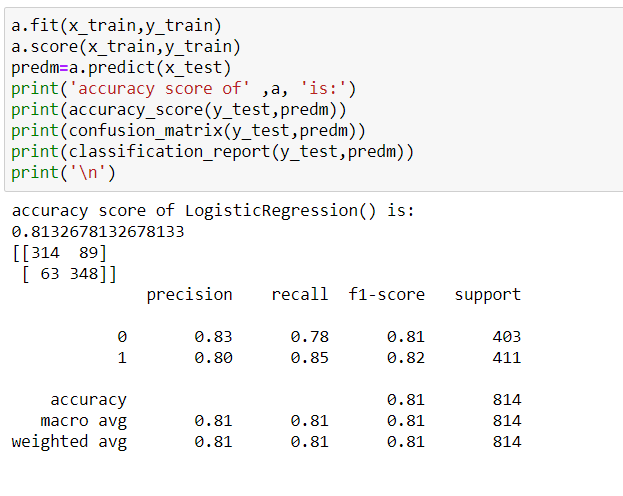
* Now we will use the random state with highest score which was printed above and will do the train and test split.



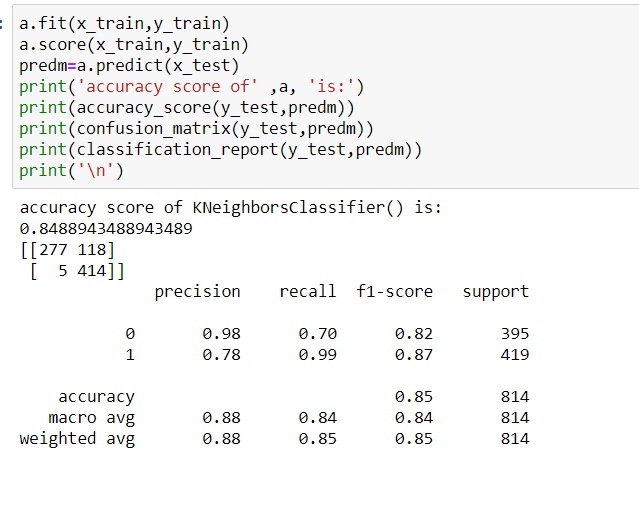
* After the split we import different models from their libraries and assign them to a variable, then using the variable on the train data we perform the model fitting. After that we send the test features to the model for predicting the target and store the prediction in a variable. Finally, we check the accuracy of the predictions by comparing the predicted and the actual target and then print the accuracy score. We also print the confusion matrix along with the classification report to get better understanding of the model performance.
* We will do all these steps for all the models. For decision tree we got the accuracy score of 82.8% and F1 score of 82% for ‘yes’ and 83% for ‘no’.



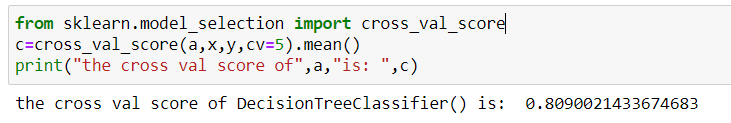
* For Random Forest Classifier we got the accuracy score of 94.2% and F1 score of 94% for ‘yes’ and 94% for ‘no’.

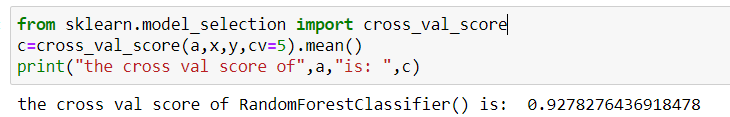


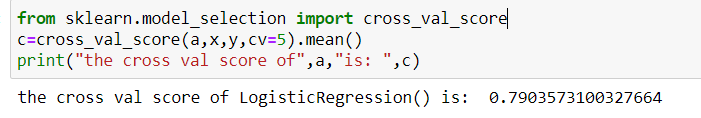
* For Logistic Regression we got the accuracy score of 81.32% and F1 score of 81% for ‘yes’ and 82% for ‘no’.

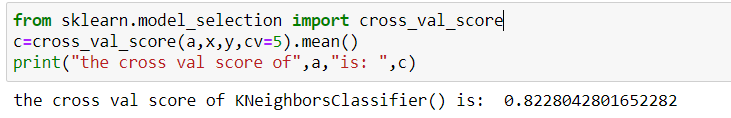


* For KNeighbors Classifier we got the accuracy score of 84.88% and F1 score of 82% for ‘yes’ and 87% for ‘no’.
* Now we need to check the cross-validation score, which basically checks if there is any overfitting or underfitting of the model.





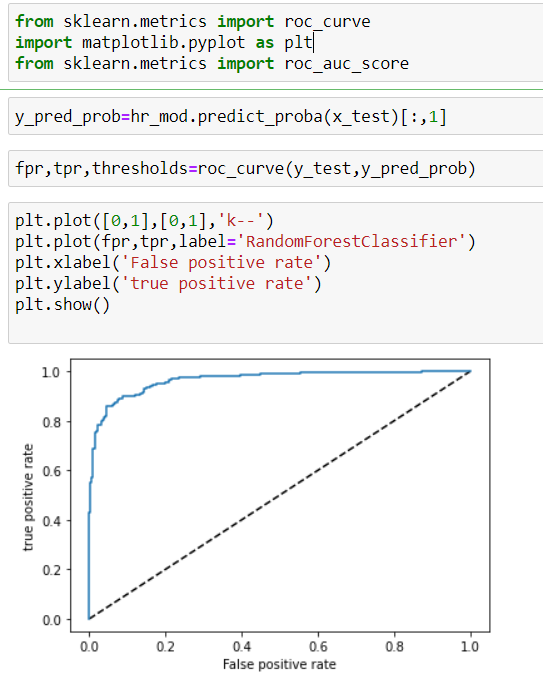


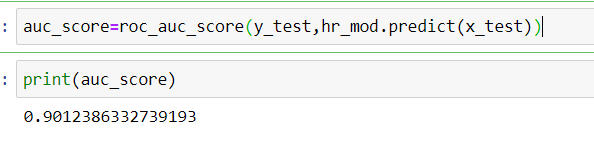


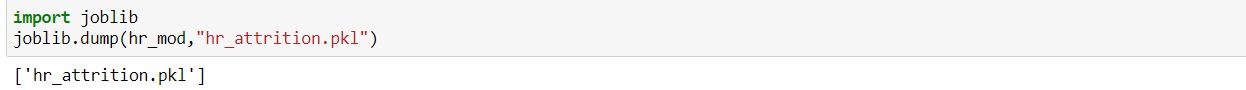
* After seeing the cross-validation score and the model performance, the least difference between the cross-validation score and model accuracy score is in Random Forest Classifier and its F1 andF2 scores are also really good. So, we will be choosing it as our final model.
* As our final model is chosen, to improve the accuracy more we will be hyperparameter tune the mode, by choosing the best parameters.
* We will import GridSearchCV and perform the hyperparameter tuning.



After Hyperparameter Tuning our score is 90.04%, which is considered as a good model. As it’s a classifier problem we need to plot the AUC-ROC curve and check the AUC-ROC score also.





* After checking the AUC-ROC score, it gives us 90.12% which is similar to our accuracy score.
* Now we will be saving the file by importing joblib and saving it with pkl extension. 

**5. Concluding Remarks:**

With the help of this project, we have an idea about how important it is to balance the target variable. Imbalanced target variable can affect our accuracy majorly. We also found that what type of data we can work with and which data we can avoid.

After plotting the correlation heatmap we had a fair bit of idea of certain attributes which are positively correlated like overtime, overtime increases, chances of attrition increases and how other factors like job level and monthly income is negatively correlated, more the job level and monthly income less are the chances of attrition. We saw that how encoding is mandatory for machine learning.

Lastly, we saw the parameters on what basis we must choose our best model, for prediction purpose by checking all scores of accuracies, F1, F2, recall, precision, AUC-ROC score. After going through all these parameters, we chose our best suitable model.