**HR Analytics Project- Understanding the attrition with HR analytic**

* **Introduction:-**

Companies hire new employees every year and spend lot of money and invest time on them to train. It is not only company provides training to their new employees but existing is also trained for effective delivery and to increase the productivity timely. “**Attrition is also an important parameter in a company and it is being reviewed rigorously by the top management on regular interval”.**

A question must be arising in the minds that how “HR analytics” helps in improve employees’ efficiency. HR analytic has significant role in process improvement. We will read further the relation and the contribution of HR analytics in our coming sections. Please read full article to get complete understanding.

This article is containing the following sub-topics

|  |  |
| --- | --- |
| **SR.No** | **Topic Name** |
| 1. | Problem Definition   1. How attrition impact the business 2. How to HR analysis help in understanding probable attrition case |
| 2. | Data Analysis   1. Understanding the data |
| 3. | EDA Concluding Remark |
| 4. | Pre-Processing Pipeline |
| 5. | Building Machine Learning Models |
| 6. | Concluding Remarks. |

**Problem Statement:**  
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?

**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? 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.

**Let’s get into details sub topic wise and understand:-**

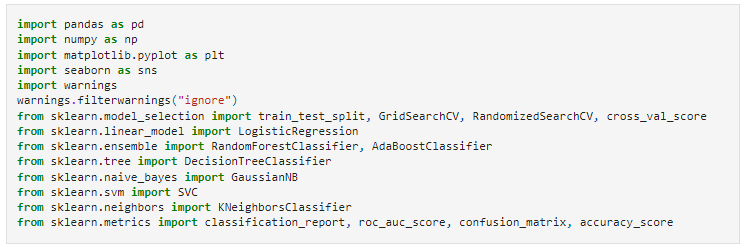
* **Problem definition:-**

As we have already read in the introduction part that companies spend money and invest time to trained new hires and run many training programs internally for existing employees to enhance their work efficiency subsequently.

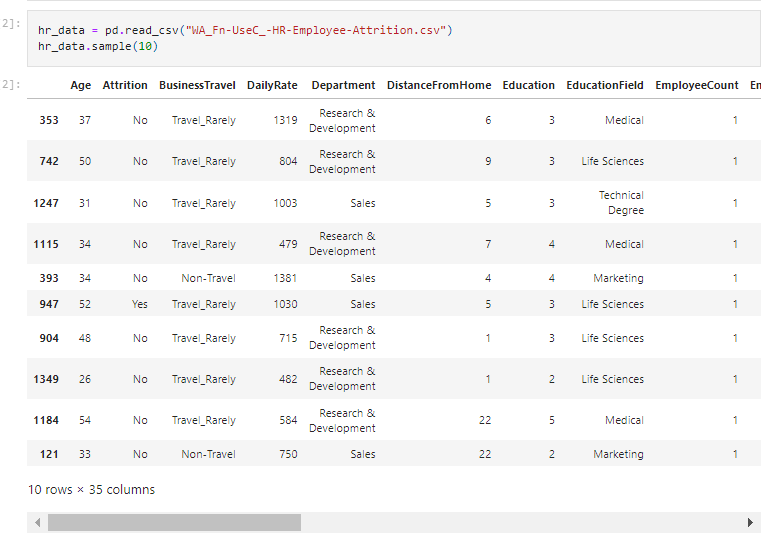
Attrition is considered a major & critical parameter in companies. It is said that old is gold and I believe this example fits perfectly in this case, experience brings lot of good work for the companies. HR conducts timely survey and basis on that gauge employee’s satisfaction and also they conduct many engagement activities to make employees feel good at work. Basis on survey data and historical attrition data, HR analytic brings many innovative idea and initiatives to control the attrition. Let’s get into deep and see how HR analytics provide insight

**Importing libraries:-**

For analyzing data, we would load dataset by using pandas’s read\_csv function , if you want to refresher your idea about pandas, please visit pandas official site and documents. we are also importing important libraries which will help in analyzing and model building.

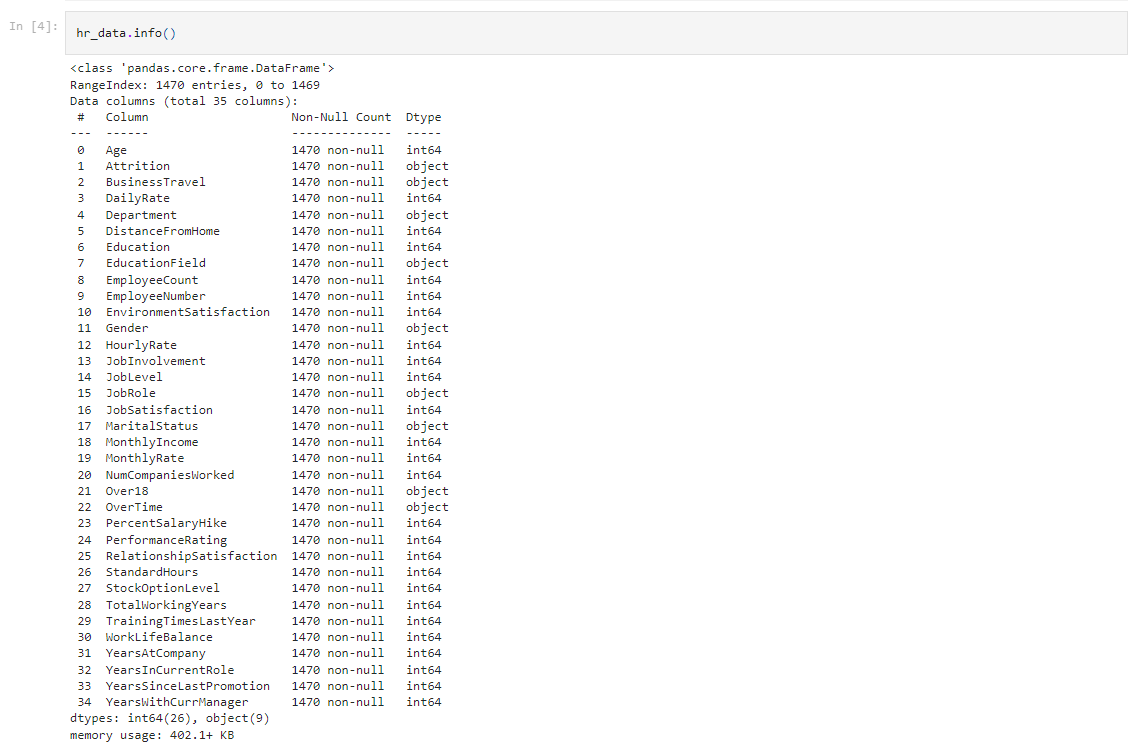


**Loading/gathering data:-**



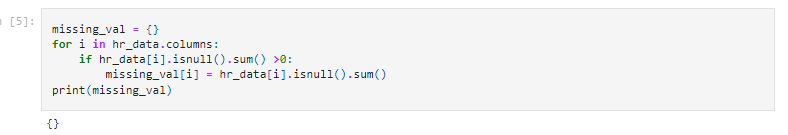
Dataset has been imported by using pandas read\_csv() function. We can see, it has mix of data types. Let’s check the shape of the dataset by calling shape method:-

* **Data Exploration/Analysis:-**



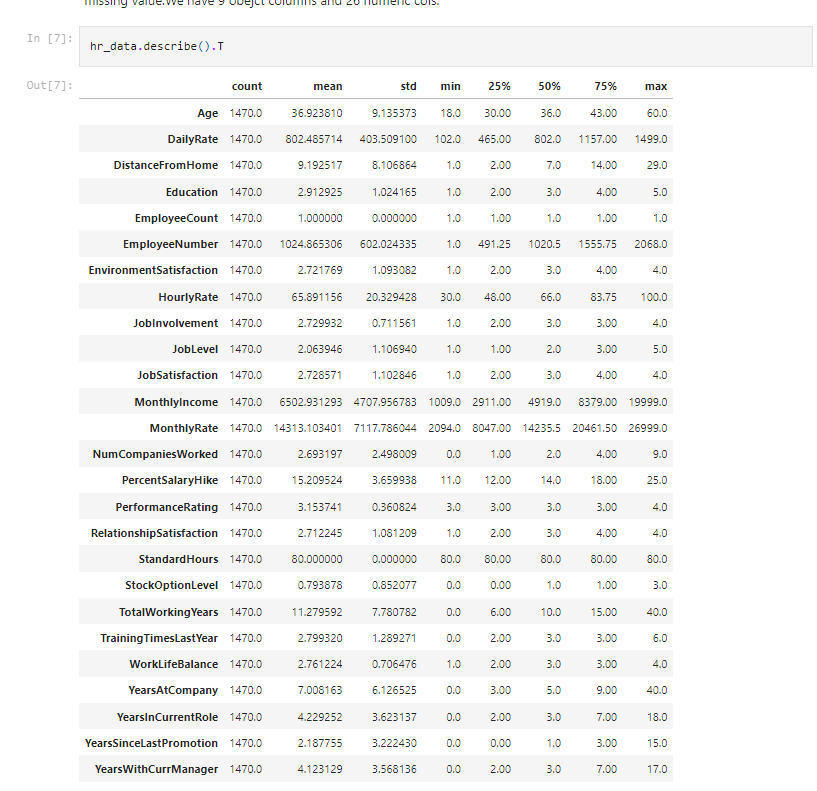
Dataset consists of 35 columns and 1470 rows. I have used hr\_data.info() function to check it.

**Checking missing values:-**



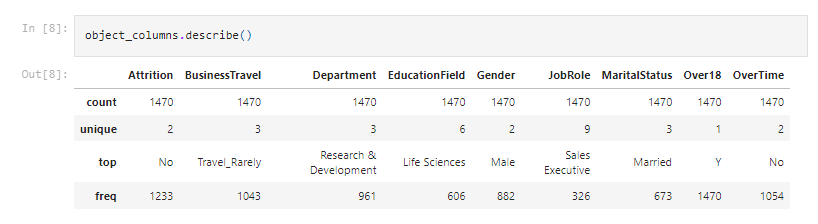
No missing value found in the data. Missing value can be checked by pandas’s .isnull() or isna() function

**Check statistical information by using describe function**

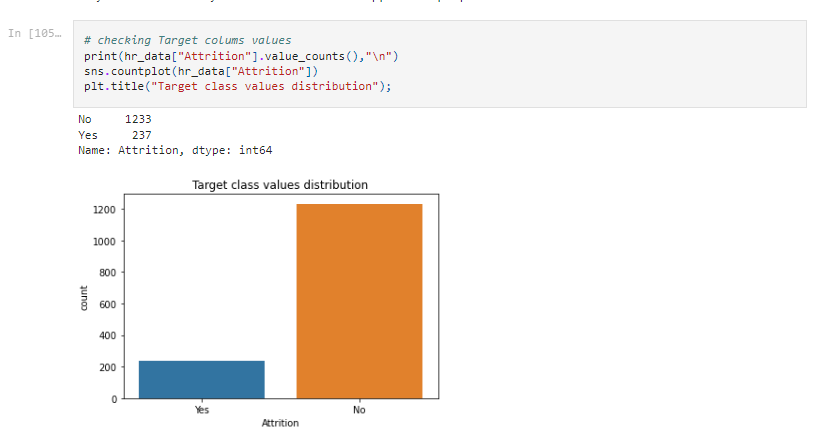


Understanding data’s stats is essentials for further analysis. Pandas describe() function provides insight about descriptive statistic information like mean, median, mode, standard deviation, min and max value and we can check the percentile of data.

Describe function primarily takes numerical data but it can be extended to string data type

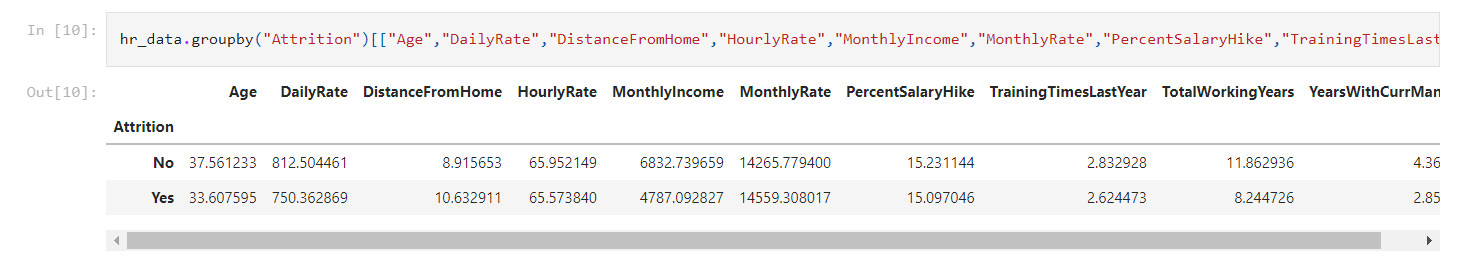


Since it is a classification problem, we are checking target column’s proposition to know that how it looks like



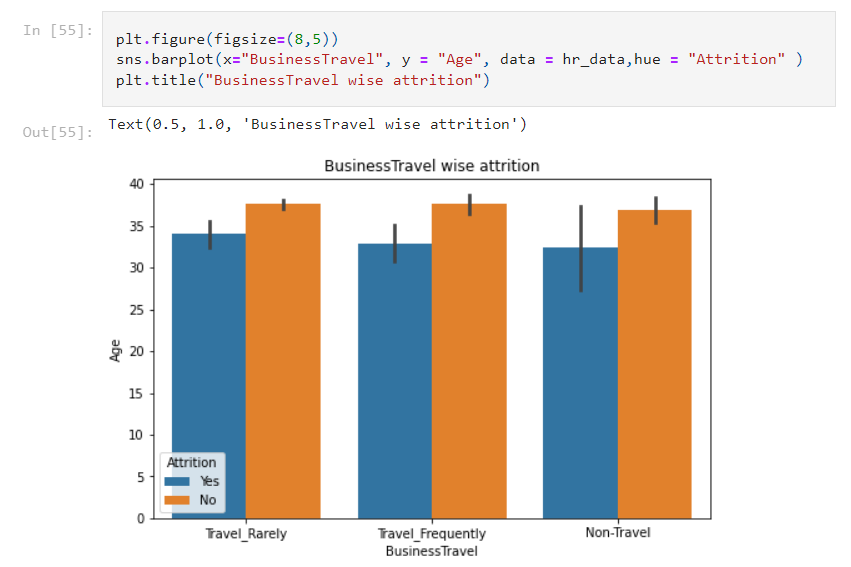
Looking at the graph, No means no attrition and its ratio is high than Yes. It is kind of imbalance problem and it has to be corrected otherwise majority class influence the minor class. It will get corrected in coming section when we start modelling.

**Feature wise attrition analysis:-**

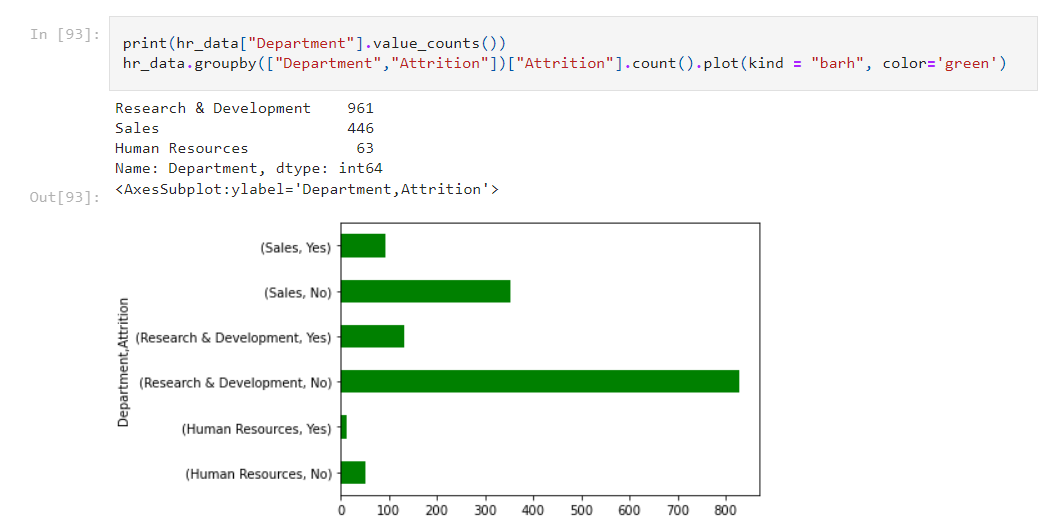


It seems that people who left the company they probably wanted to change or looking for other opportunity. Age is also one factor; they were younger than the other people who didn’t attrite. We can observe couple of more factors that people attrited like average daily rate, distance from home, monthly income.

**Let us explore more things:-**

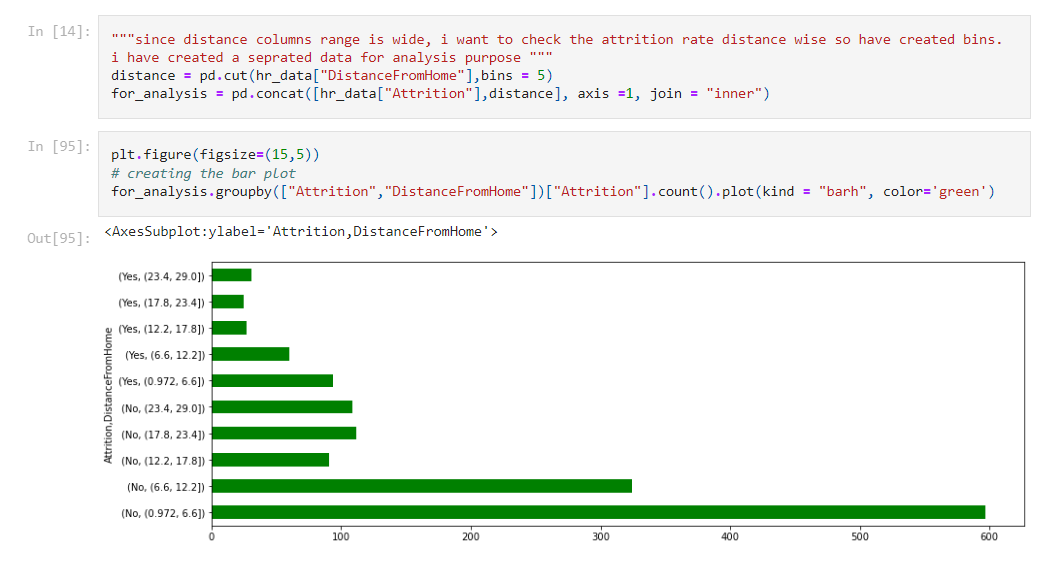


I tried checking the relationship between travel roles and non travel roles, have seen no relation between attrition and travel profile, I thought of may be travelling would be high as people didn’t like to travel frequently



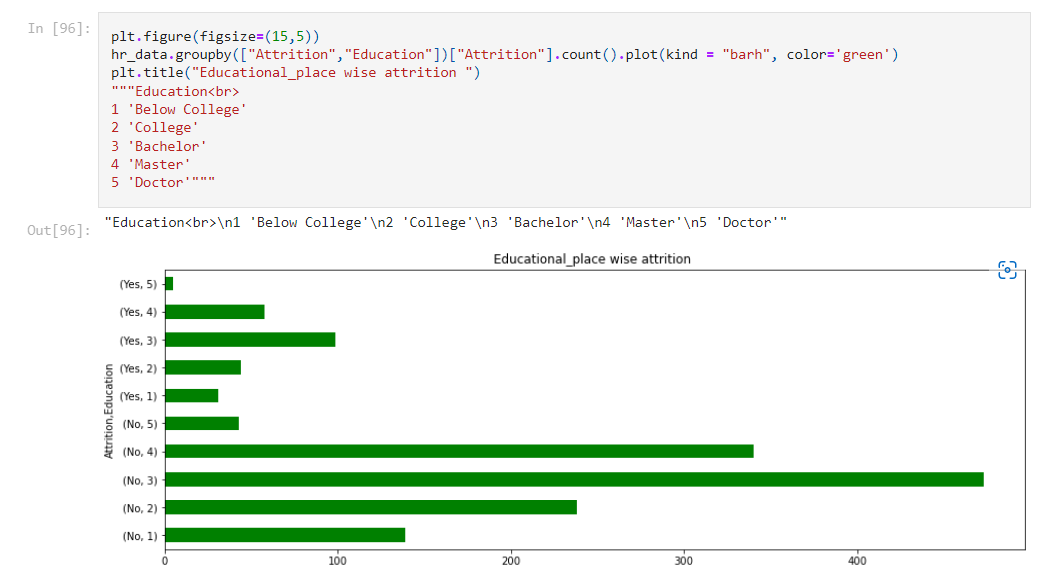
By analyzing the department wise attrition, high attrition is observed in R&D department.

**Distance wise attrition**



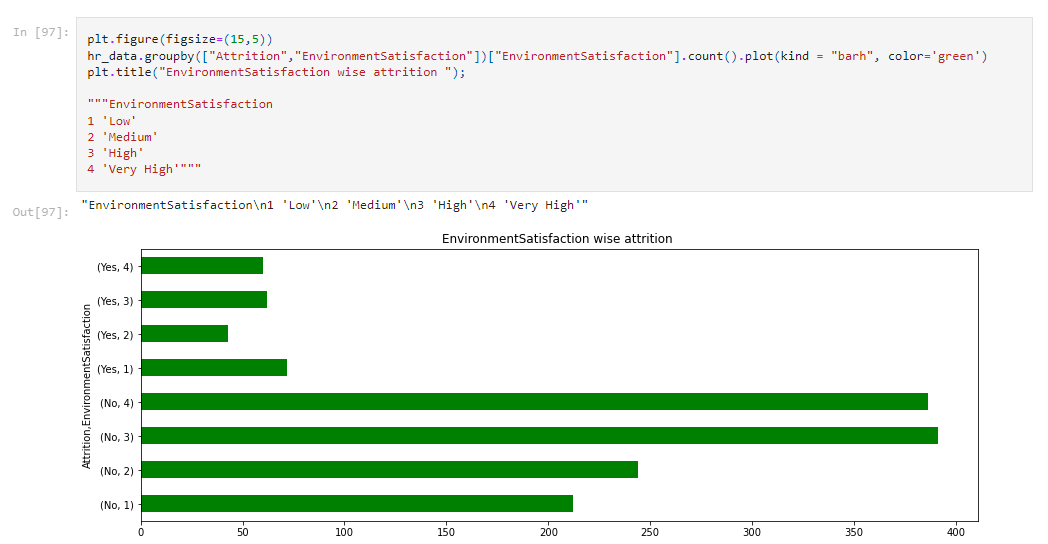
Major Attrition observed from 0.972 to 12.2 KM areas, people who list nearby the company.

**Education wise attrition:**



Maximum attrition happened from bachelor.

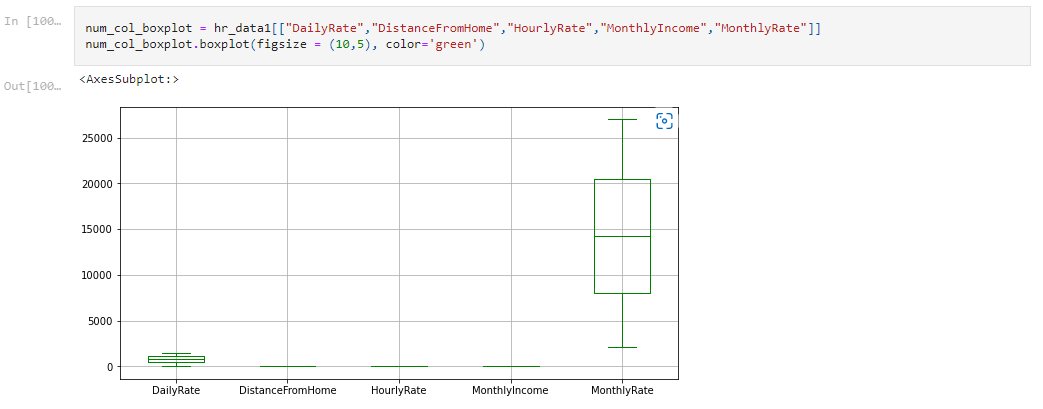
**Environment satisfaction wise attrition:**



A strong co relation has been seen that low environment satisfaction led to high attrition.

We have tried analyzing attrition with multiple component and found interesting information. Let’s check the outlier in the dataset by using box plot.

**Outrlier detection using boxplot**



We see that couple of high values in the monthlyincome column above the top whisker but rest of the columns don’t have outlier.

**Skewness**



Skewness is a measure of the symmetry in a distribution. ... It measures the amount of probability in the tails.

Standard threshold is 0.5 which is considered to be used to modelling, if it goes up we need it to be corrected. There are several ways to correct it, a few operations which is applied to correct them, these are square root transformation, cube root transformation, log transformation (for non zero values), logp1 transformation (if particular column have zero values in it), boxcox transformation for non zero and positive data point, power transformer ( in scikit learn power transformer, we have two method 1. yeo-johnson' which is default one and 2. Boxcox. So apply transformation as per the column nature.

* **Pre-Processing Pipeline**

Label encoding, machine learning algorithm takes numerical input for the learning, thus it becomes important that data should be converted into number. We have a few categorical columns let’s convert them.

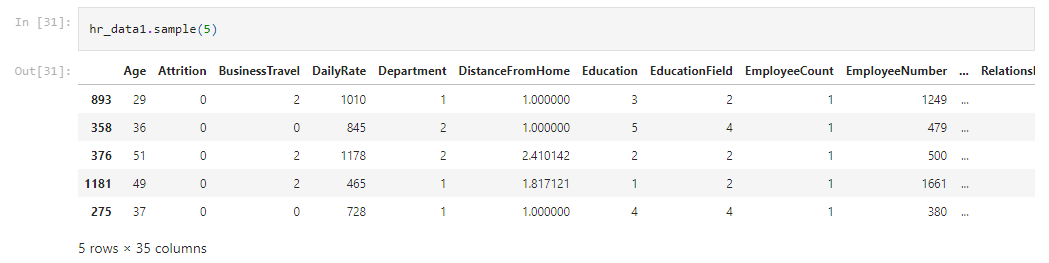


I have used pandas replace method to convert categorical columns into numerical one. One can use scikit learn method as well to convert it.

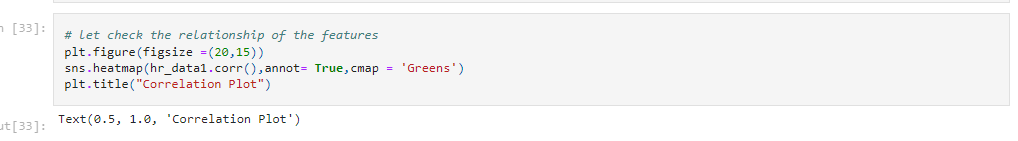
We have converted below categorical columns into numerical:-

1. Attrition ( our target feature)
2. Gender
3. Department
4. Marital status
5. Over time
6. Business travel
7. Education field
8. Job role

**Converted data after label encoding**



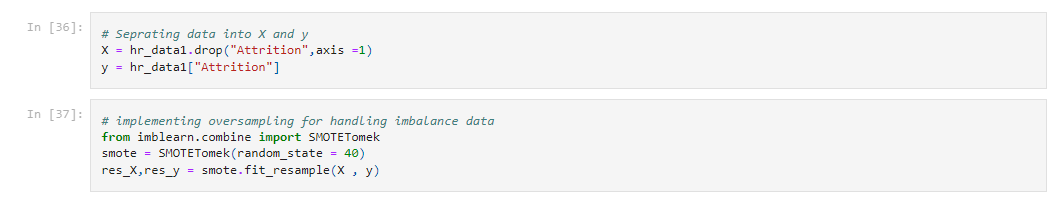
Now all the features are numerical features, we will use correlation plot to check correlation between with target variable and other individual feature. I have used below code snippet to populate correlation.



We have 35 columns and it is difficult to show heatmap in this article. You can visit seaborn heatmap to check how it looks like if you are a beginner.

* **Building Machine Learning Models**

We saw in the above section while checking the proposition of target features, it wasn’t equal, it was data imbalance problem and we discussed that would see the solution so here is the solution:-



For handling imbalance problem, we are using SMOTE method with up-sampling, we have 1470 dataset and we cannot use down sampling because by doing that we would use data and having small size of data we cannot lose that thus up-sampling has been applied.

**Applying ML algorithm**

We will use multiple machine learning algorithms to train model and pick the best one.



training score: 1.0 Random\_state 50

Test score: 0.9047619047619048 Random\_state 50

Accuracy : 0.9047619047619048

Roc\_auc\_score : 0.9045452674049982

confuion\_matrix

[[192 22]

[ 20 207]]

classification\_report

precision recall f1-score support

0 0.91 0.90 0.90 214

1 0.90 0.91 0.91 227

accuracy 0.90 441

macro avg 0.90 0.90 0.90 441

weighted avg 0.90 0.90 0.90 441

training score: 1.0 Random\_state 125

Test score: 0.9229024943310657 Random\_state 125

Accuracy : 0.9229024943310657

Roc\_auc\_score : 0.9237378940861322

confuion\_matrix

[[208 22]

[ 12 199]]

classification\_report

precision recall f1-score support

0 0.95 0.90 0.92 230

1 0.90 0.94 0.92 211

accuracy 0.92 441

macro avg 0.92 0.92 0.92 441

weighted avg 0.92 0.92 0.92 441

training score: 1.0 Random\_state 200

Test score: 0.9138321995464853 Random\_state 200

Accuracy : 0.9138321995464853

Roc\_auc\_score : 0.9157313879536102

confuion\_matrix

[[218 25]

[ 13 185]]

classification\_report

precision recall f1-score support

0 0.94 0.90 0.92 243

1 0.88 0.93 0.91 198

accuracy 0.91 441

macro avg 0.91 0.92 0.91 441

weighted avg 0.92 0.91 0.91 441

training score: 1.0 Random\_state 275

Test score: 0.9160997732426304 Random\_state 275

Accuracy : 0.9160997732426304

Roc\_auc\_score : 0.9188025297495215

confuion\_matrix

[[218 26]

[ 11 186]]

classification\_report

precision recall f1-score support

0 0.95 0.89 0.92 244

1 0.88 0.94 0.91 197

accuracy 0.92 441

macro avg 0.91 0.92 0.92 441

weighted avg 0.92 0.92 0.92 441

training score: 1.0 Random\_state 350

Test score: 0.9070294784580499 Random\_state 350

Accuracy : 0.9070294784580499

Roc\_auc\_score : 0.9075460829493087

confuion\_matrix

[[196 28]

[ 13 204]]

classification\_report

precision recall f1-score support

0 0.94 0.88 0.91 224

1 0.88 0.94 0.91 217

accuracy 0.91 441

macro avg 0.91 0.91 0.91 441

weighted avg 0.91 0.91 0.91 441

training score: 1.0 Random\_state 425

Test score: 0.8956916099773242 Random\_state 425

Accuracy : 0.8956916099773242

Roc\_auc\_score : 0.8958874458874458

confuion\_matrix

[[206 25]

[ 21 189]]

classification\_report

precision recall f1-score support

0 0.91 0.89 0.90 231

1 0.88 0.90 0.89 210

accuracy 0.90 441

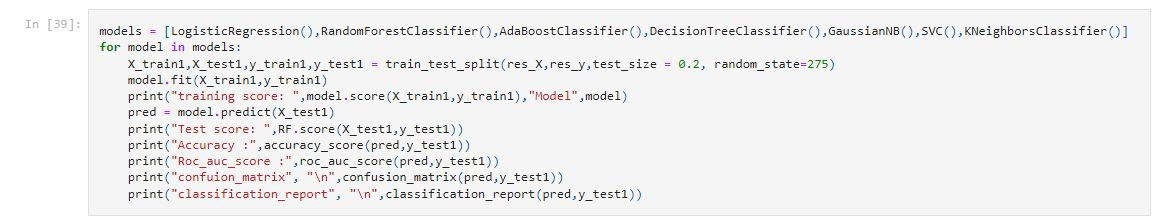
macro avg 0.90 0.90 0.90 441

weighted avg 0.90 0.90 0.90 441

I have used random forest as first algorithm, it is givive 100 accuarcy on training test and approx 91 test set.i have used for loop to find the best random state. We can see that random\_state 275 is giving best accuracy with good recall and precision score out of other random state.

The 100 accuracy on training set will definitely raise question in the mind. To answer this question, i have checked the test score as well as roc\_auc\_score and it has been found that roc\_auc\_score is 92.02 which means that model is able to identify 92.02 data correctly.

**Using multiple algorithms:-**

****

I have used logistic regression, adaboost, decision tree, Gaussian NB, support vector and KNN.

training score: 0.7546848381601363 Model LogisticRegression()

Test score: 0.9886621315192744

Accuracy : 0.7505668934240363

Roc\_auc\_score : 0.7502785572796302

confuion\_matrix

[[176 57]

[ 53 155]]

classification\_report

precision recall f1-score support

0 0.77 0.76 0.76 233

1 0.73 0.75 0.74 208

accuracy 0.75 441

macro avg 0.75 0.75 0.75 441

weighted avg 0.75 0.75 0.75 441

training score: 1.0 Model RandomForestClassifier()

Test score: 0.9886621315192744

Accuracy : 0.9160997732426304

Roc\_auc\_score : 0.9188025297495215

confuion\_matrix

[[218 26]

[ 11 186]]

classification\_report

precision recall f1-score support

0 0.95 0.89 0.92 244

1 0.88 0.94 0.91 197

accuracy 0.92 441

macro avg 0.91 0.92 0.92 441

weighted avg 0.92 0.92 0.92 441

training score: 0.8966496308915389 Model AdaBoostClassifier()

Test score: 0.9886621315192744

Accuracy : 0.8775510204081632

Roc\_auc\_score : 0.8776720864971939

confuion\_matrix

[[204 29]

[ 25 183]]

classification\_report

precision recall f1-score support

0 0.89 0.88 0.88 233

1 0.86 0.88 0.87 208

accuracy 0.88 441

macro avg 0.88 0.88 0.88 441

weighted avg 0.88 0.88 0.88 441

training score: 1.0 Model DecisionTreeClassifier()

Test score: 0.9886621315192744

Accuracy : 0.81859410430839

Roc\_auc\_score : 0.8189804147465438

confuion\_matrix

[[183 34]

[ 46 178]]

classification\_report

precision recall f1-score support

0 0.80 0.84 0.82 217

1 0.84 0.79 0.82 224

accuracy 0.82 441

macro avg 0.82 0.82 0.82 441

weighted avg 0.82 0.82 0.82 441

training score: 0.8239636570130607 Model GaussianNB()

Test score: 0.9886621315192744

Accuracy : 0.8208616780045351

Roc\_auc\_score : 0.8207454029371838

confuion\_matrix

[[186 36]

[ 43 176]]

classification\_report

precision recall f1-score support

0 0.81 0.84 0.82 222

1 0.83 0.80 0.82 219

accuracy 0.82 441

macro avg 0.82 0.82 0.82 441

weighted avg 0.82 0.82 0.82 441

training score: 0.5173197047132311 Model SVC()

Test score: 0.9886621315192744

Accuracy : 0.5056689342403629

Roc\_auc\_score : 0.5090054898648648

confuion\_matrix

[[ 98 87]

[131 125]]

classification\_report

precision recall f1-score support

0 0.43 0.53 0.47 185

1 0.59 0.49 0.53 256

accuracy 0.51 441

macro avg 0.51 0.51 0.50 441

weighted avg 0.52 0.51 0.51 441

training score: 0.8341851220897217 Model KNeighborsClassifier()

Test score: 0.9886621315192744

Accuracy : 0.782312925170068

Roc\_auc\_score : 0.794097089846232

confuion\_matrix

[[158 25]

[ 71 187]]

classification\_report

precision recall f1-score support

0 0.69 0.86 0.77 183

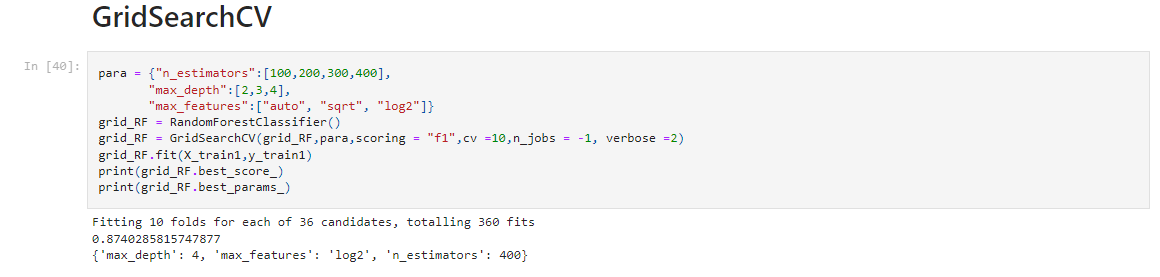
1 0.88 0.72 0.80 258

accuracy 0.78 441

macro avg 0.79 0.79 0.78 441

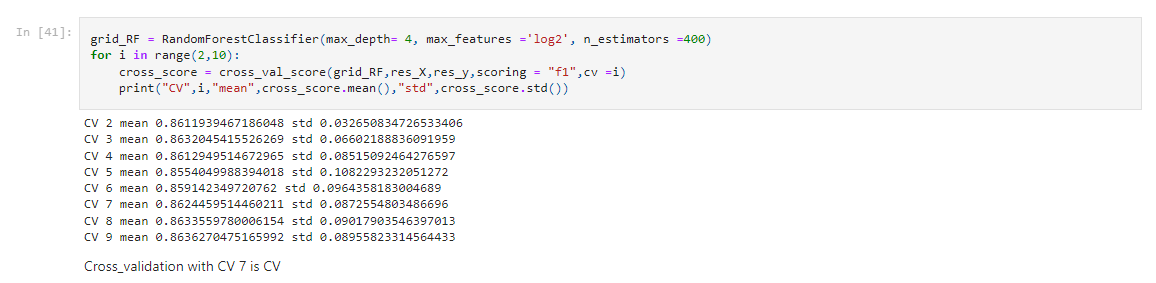
weighted avg 0.80 0.78 0.78 441

Random forest is giving better result as compare to other algorithms, so i have decided it to select as best and process further operation like hyper parameter tuning and cross validation

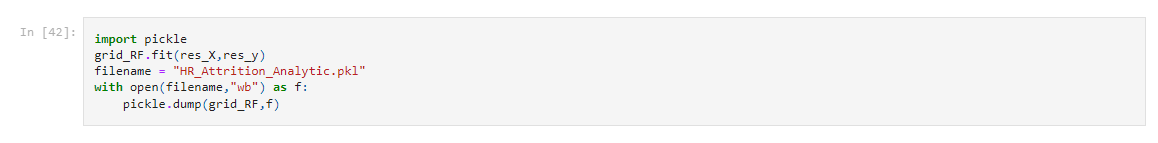


I have used Gridsearch CV to find best parameter of random forest.

Now, we are going to perform cross validation, cross validation train model on all the data points so that model learn every pattern and provide average score of all the models. We get score by using mean function and we check the standard deviation as well of the score. Which ensure us that given score doesn’t have risk factor. Here is the code snippet:-



**Model saving**



**Conclusion:-**

Dataset was quite clear, there was no missing value. It was mix of categorical and numerical features. We have performed multiple analyses to check that which factor plays important role in attrition. We checked outlier and found that few columns have some extreme value but it is very close to upper whisker and we didn’t try treating them because the ensemble methods will deal with them. I have checked correlated of each features and found that couple of features were correlated so have deleted them.  
As we saw at the initial phase of analysis that data was imbalance, we have corrected that by applying oversampling technique and then Model was trained. Random forest has given best F1 score and has taken it for final model.