**HR Analytics – Understanding the Attrition in HR using Machine Learning**

Employee Attrition is a major cost to an organization, and predicting turnover is at the forefront of needs of Human Resources (HR) in many organizations. 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.

With advances in machine learning, it is possible to not only predict employee attrition but to understand the key variables that influence turnover.

**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?

**Data Preparation:**

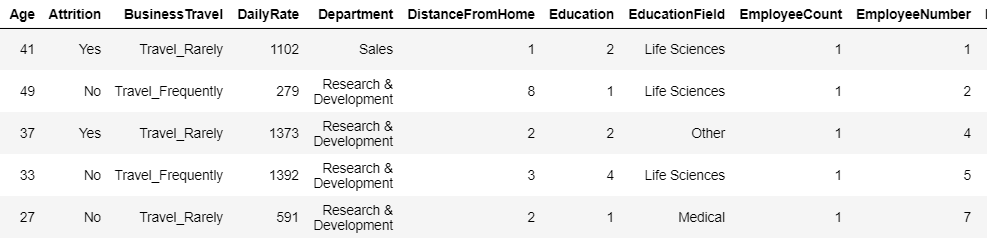
The dataset includes Age, BusinessTravel, DailyRate, Department, DistanceFromHome, Education, EducationField Etc. Let us try to study the factors that lead to employee attrition.

The data can be downloaded from [here](https://github.com/dsrscientist/IBM_HR_Attrition_Rate_Analytics).

#Load the data

data=pd.read\_csv("HR Employee Attrition.csv")

data.head()

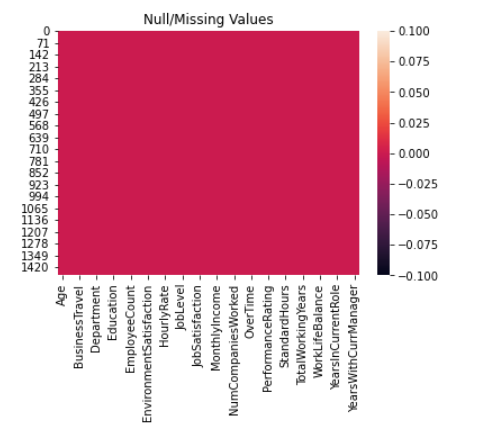


#Visualize and check for Null/Missing Values

sns.heatmap(data.isnull())

plt.title("Null/Missing Values")

plt.show()

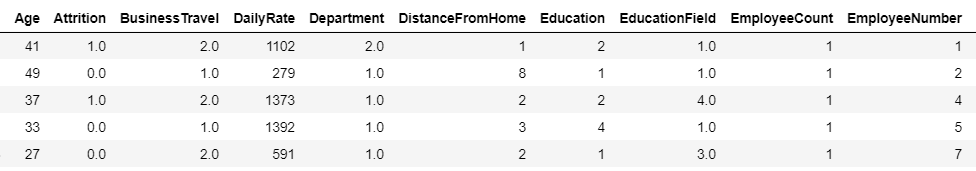


Good that we do not have any missing or null values in the dataset.

**Data Analysis:**

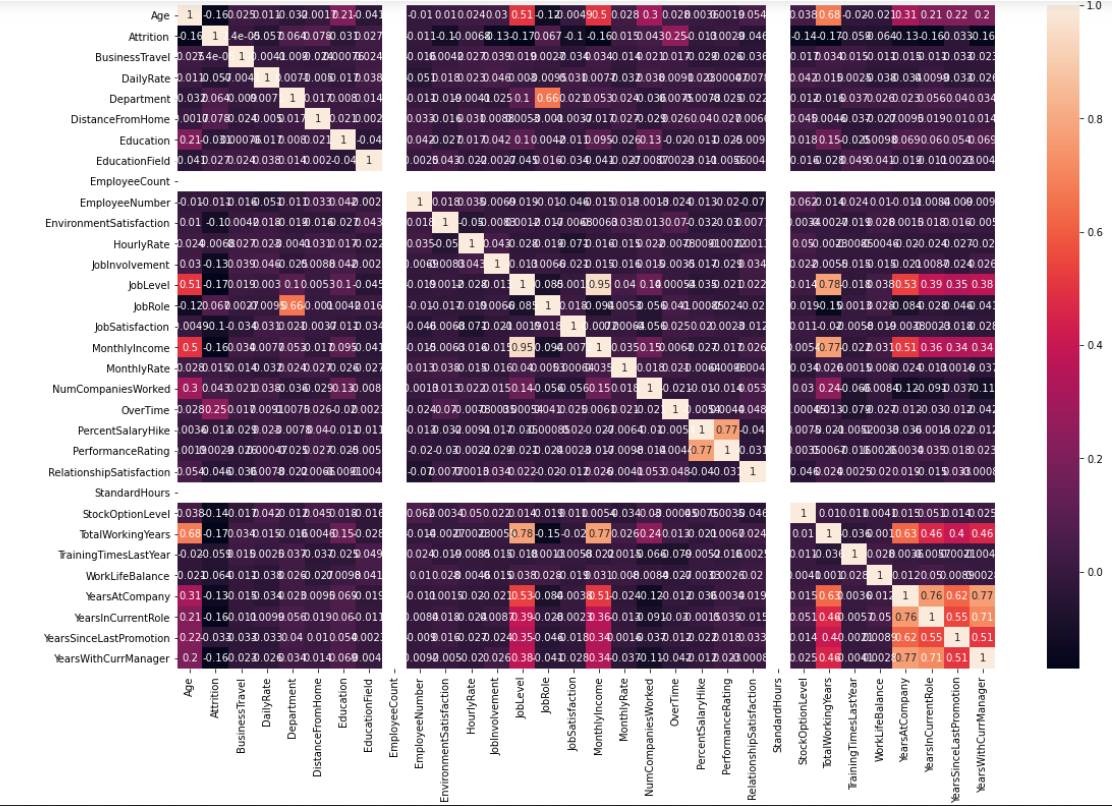
Some of the columns have numerical values, as in the case of Age, DistanceFromHome, DailyRate etc. while the other have text-based (or categorical) values such as BusinessTravel, Department, EducationField etc.

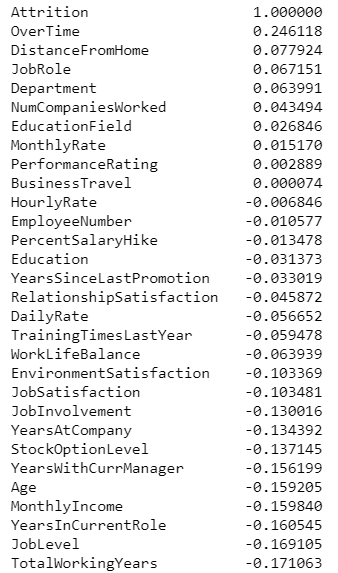
In such scenario, where the data has both numerical and categorical values, we use Ordinal Encoding technique to better handle the categorical datatypes. We import OrdinalEncoder utility from scikit-learn. The OrdinalEncoder will fit and transform all the categorical values into numeric.



Data after Encoding

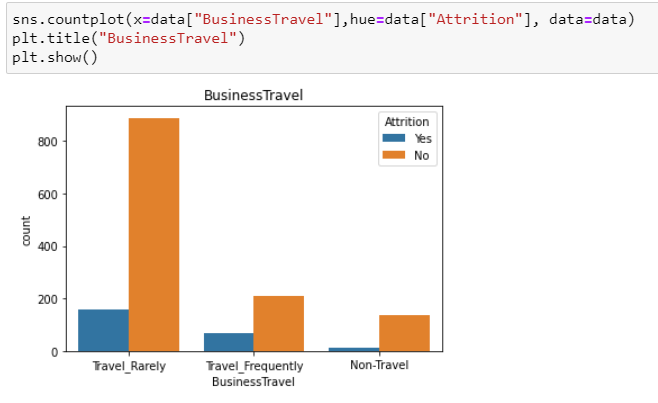
Now we can create a seaborn heatmap to see the correlation between the feature variable and target variable. Also, below we can see the correlation with respect to target variable Attrition.



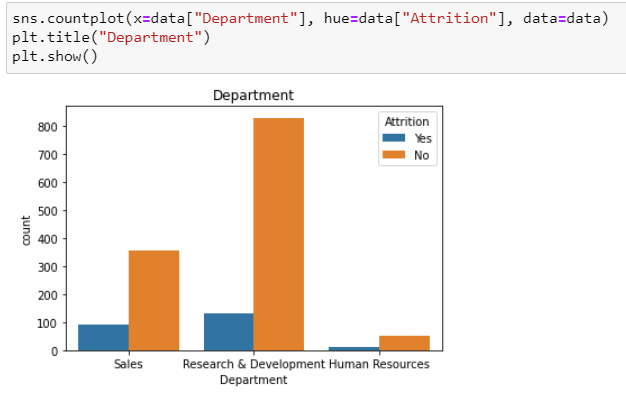


On plotting a count-plot of the BusinessTravel, we can notice that

1. Travel\_Rarely are the ones who got attrition more compared to travel\_Frequently and Non\_Travel employees.
2. Travel\_Rarely has highest number of employees in it.

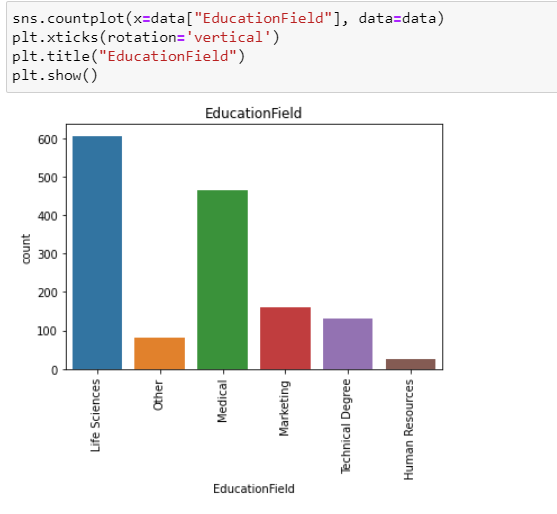


Now let’s find Department-wise performance.



1. Research & Development domain has highest number of employees followed by the Sales and Human Resources are very less.
2. Attrition is higher in Research and Development Area.

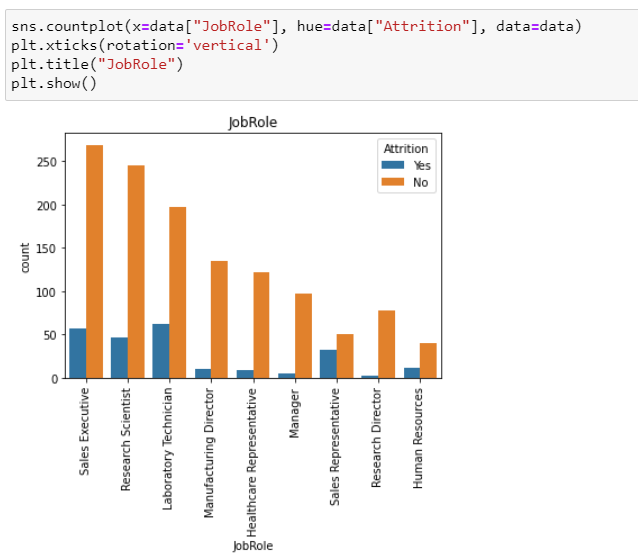
Let’s look into EducationField.



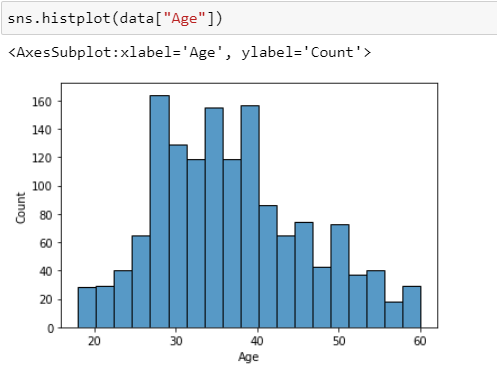
1. Life Sciences is the education of highest number of employees followed by the Medical Field.
2. Human Resources are the lowest number of employees.

Now when we look at the Jobrole versus Attrition, we can see that

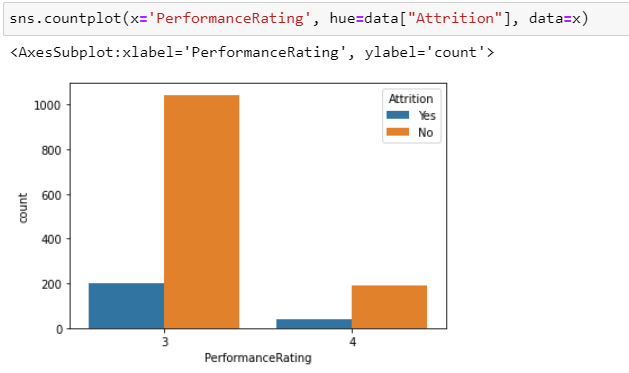
1. Laboratory Technicians are more likely to be attrition followed by the Sales Executives and Research Scientists.
2. Sales Executives and Research Scientists are higher in number compared to others.



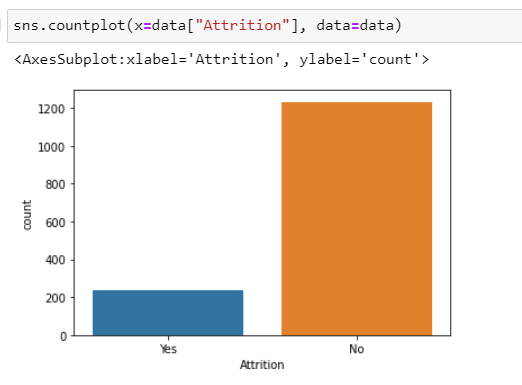
When we consider Age, higher number of employees from 25 to 40 years of Age.



By looking at the PerformanceRating, employees are having only 3 and 4 rating. Attrition is higher in Rating 3 and less in Rating 4.

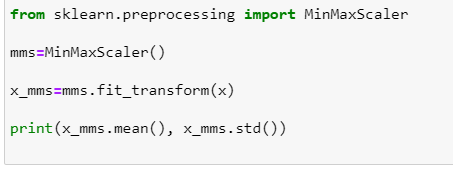


Now, we check the target variable: Attrition. It turned out that it is a categorical value, consisting of binary value Yes or No, making it a typical classification problem.

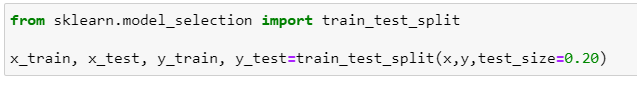


Then we separate feature variables and target variables for splitting up the data set to train set and test set.

Now the whole feature set is scaled using MinMaxScaler utility from sklearn preprocessing library to bring all the datapoints to similar scale.



We then split the dataset into training set and testing set using train\_test\_split utility from the scikit learn model selection library.

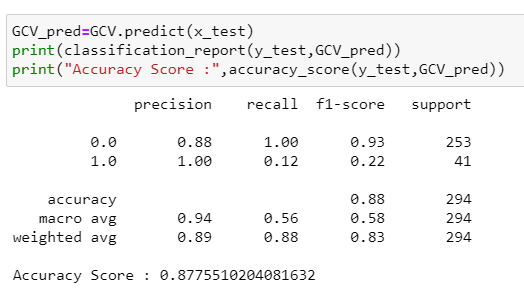


**Model Building:**

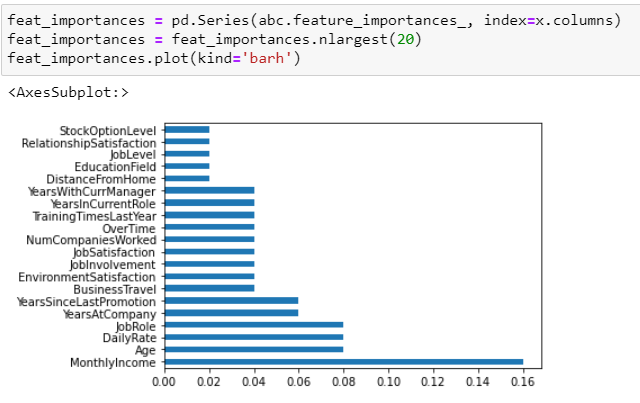
The AdaBoostClassifier is ensemble learning method is chosen over other algorithms because it was initially created to increase the efficiency of the binary classifiers. AdaBoost uses an iterative approach to learn from the mistakes of weak classifiers and turn them into strong ones.

AdaBoost is easier to use with less need for tweaking parameters. Also, it is not prone to overfitting.

We then train the AdaBoostClassifier model and apply the AdaBoostClassier to Testing data set. We check accuracy score with cross validation as well to overcome the problem of overfitting or underfitting of the data. Then we perform Hyper parameter tuning to get the best parameters with highest scores.



Our accuracy score is 88% with AdaBoostClassifier.Our model is performing very good.



Also, when we see the feature importance, we found that monthlyIncome contributes the most in Attrition of the employees.