HR Analytics – Project Report

(From Machine Learning – 1st Evaluation Project)

Contents:

1) Problem Definition

2) EDA and its Concluding Remarks

3) Pre- Processing Pipeline

4) Machine Learning Models

5) Concluding Remarks

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?

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 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.

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.

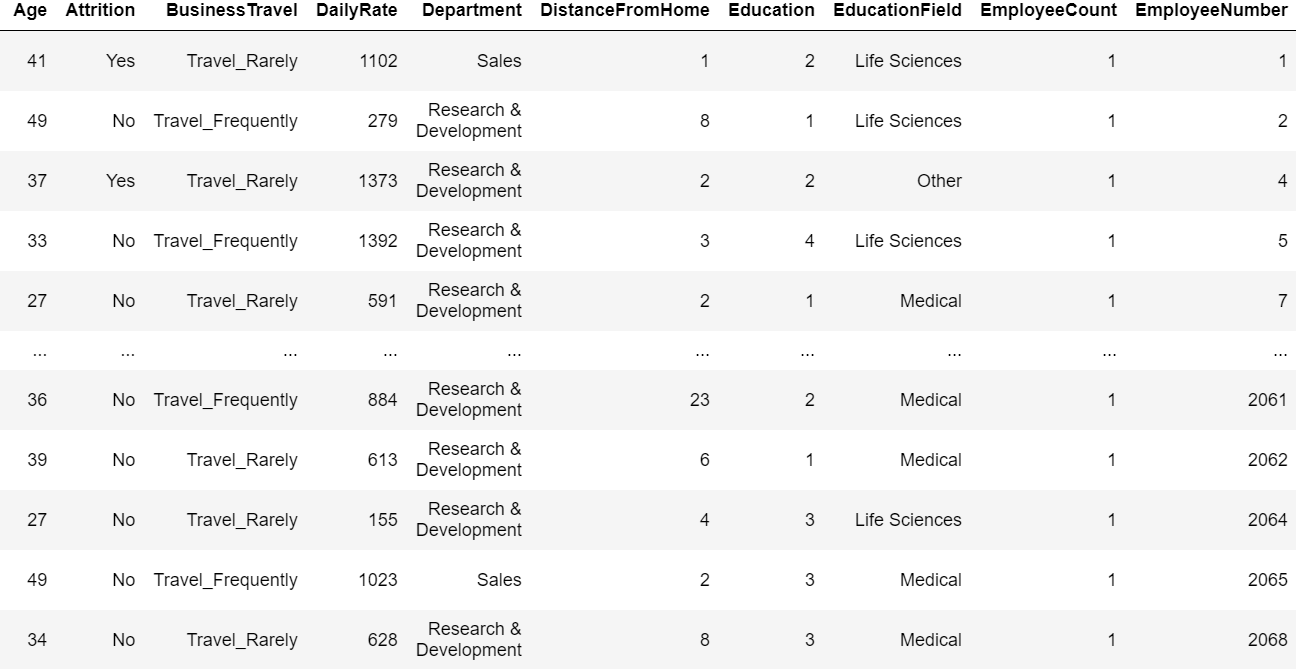
Problem Description

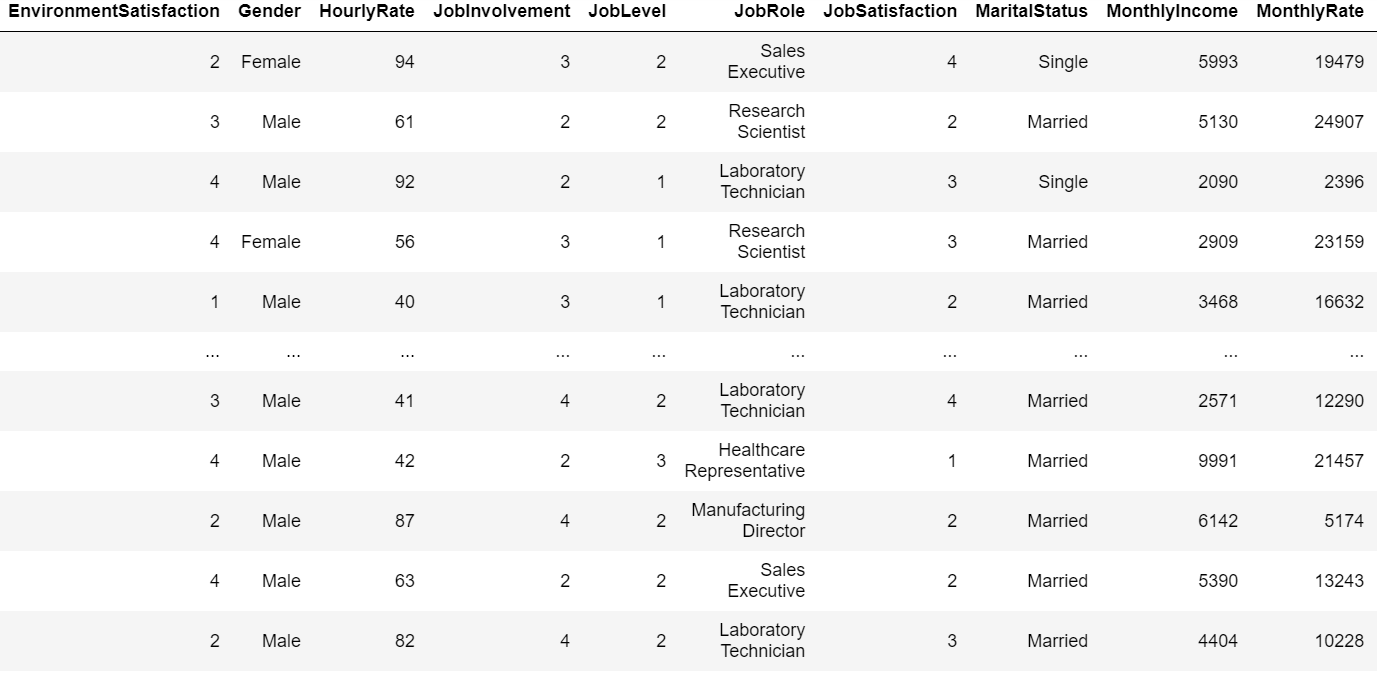
Now that we have understood what HR analytics mean and what is the meaning of Attrition in HR and how does it affect the companies overall performance. The main aim of this whole project is to understand the attrition rate of the company and analysing this attrition rate to inform the HR of the particular company to take necessary measures to increase the effectiveness of the existing employees.

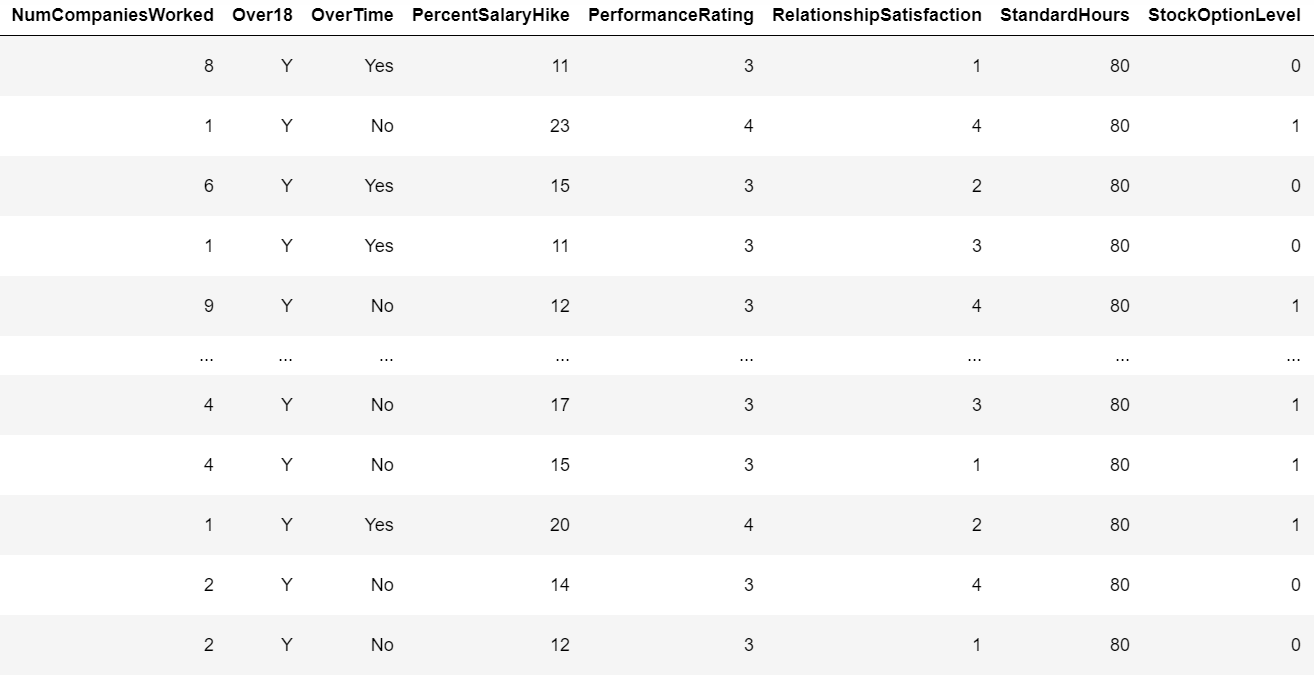
Data Analysis (EDA)

Now, that we have understood what our main aim of this whole project, let us take a look at our dataset and what are the different features (independent variables) present which helps us in understanding the various reasons why there is attrition or not in the company.

a) Dataset info









As you can see from the snaps above, we see the different columns in the dataset which can be helpful for our analysis. All the columns of our dataset have relevant information about the companies employees.

b) Feature Information

Information about some of the features in the dataset (Mostly given By HR of the company)

* *Age* - Age of the Employee
* *Gender* - Gender of the Employee
* *Attrition* - Whether the employee is going to leave the company or not (Yes/No) – (Dependent Variable)
* *Business Travel*  - Whether the employee travels for a business meeting or consultation
* *Department* - Department the employee belongs to
* *Distance from Home* - Travelling Distance from home to office
* *Education* - Level of Education of Employee (ranging from 1 to 5)
* *Education Field* - Field in which the Employee has completed his/her education from.
* *Employee Number* - Distinct Numbers given to Employees by the company.
* *Environment Satisfaction* – Workspace/Environment Satisfaction rating from the Employee (Range = 1 to 5)
* *Job Involvement*  - Involvement of Employee in his/her job (1 to 5)
* *Job Level* - Level of Job Employee is working in (1 to 5)
* *Job Role* - Various Job Roles of Employees in the company.
* *Job Satisfaction* - How satisfied is the employee with his/her job (1 to 5)
* *Marital Status* - Marital status of the employee.
* *Monthly Income* - Monthly Income of the employee.
* *NumCompaniesWorked* - Number of Companies employee has worked before joining their present company.
* *Over 18* - Is the Employee Over18 or not (Y/N)
* *OverTime*  - Is the Employee Working OverTime (Yes/No)
* *PerecentSalaryHike* - Percentage of Salary Hike of the Employee.
* *PerformanceRating* - Performance Rating of the Employee (1 to 5)
* *StandardHours* - Standard Hours an Employee should complete.
* *TotalWorkingYears* - Years Employee has been working (Both past/present companies)
* *TrainingTimesLastYear* - Times Employee has gone through Training in the company
* *WorkLifeBalance* - Work-Life Balance of the employee (1 to 5)
* *YearsatCompany* - Number of years Employee has been working in the company.
* *YearsinCurrentRole*  - Number of years Employee has been working in the same role in the same company.
* *YearsSinceLastPromotion* - Number of years since he/she (employee) had a promotion.
* *YearsWithCurrManager* - Number of years employee had been working with the current manager.

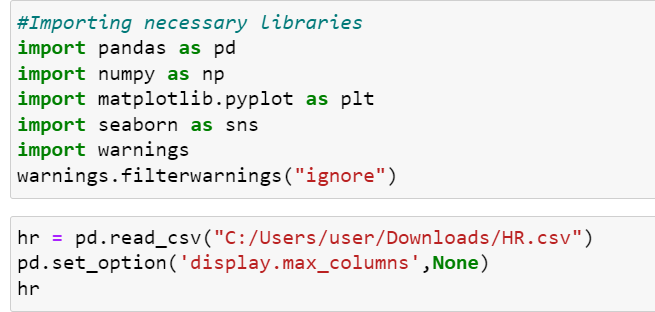
These are some of the information about different features in the dataset.

(Link to Dataset - <https://github.com/dsrscientist/IBM_HR_Attrition_Rate_Analytics>)

EDA PROCESS

STEP -1

Importing Dataset

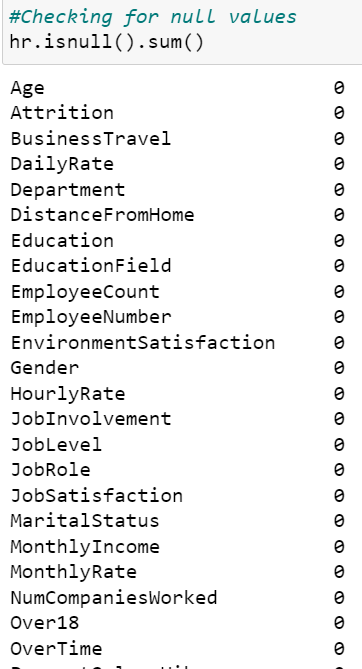


From the snap above, we are importing the necessary libraries needed for our analysis and then we are importing the csv file (read\_csv()) where our data is present.

STEP – 2

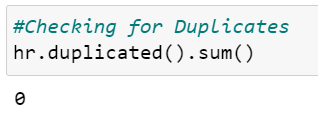
Data Cleaning (i.e removing unnecessary features from the dataset, checking for null values , checking for duplicate values)

Checking for null values



From the above code snippet, we see the code for checking null values in the dataset for each column. If there was any null values present we can either drop those columns or use mean, median or mode imputation method to get rid of those null values.

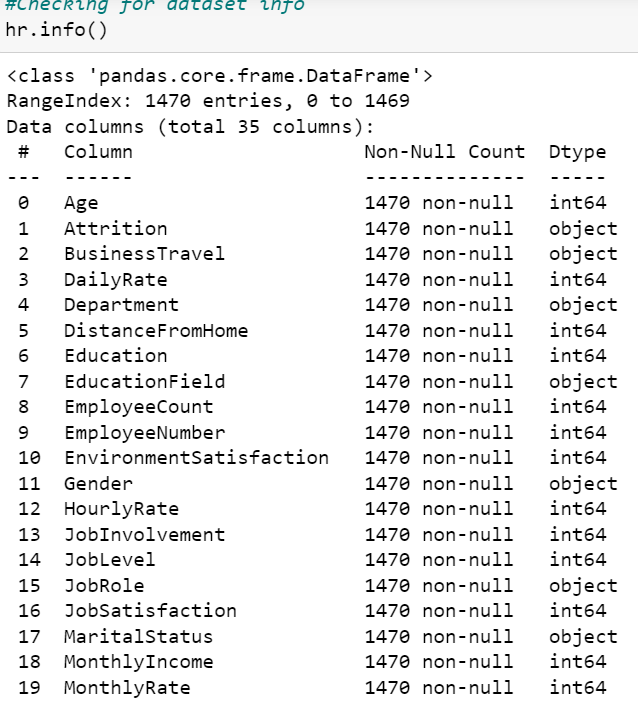
Checking for Duplicate values



We see that from the above code snippet, there are no duplicate values present in the dataset. Which means that there are 0 rows present having the same values in each column.

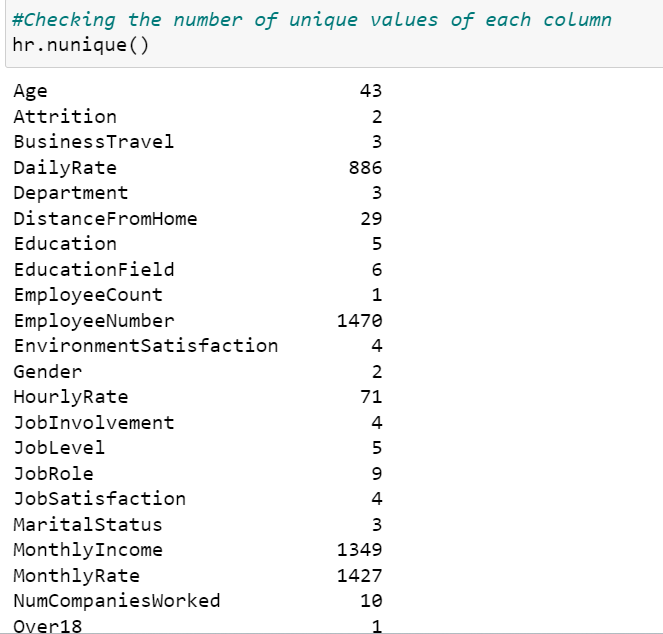
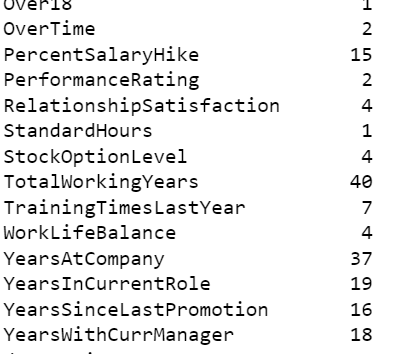
Checking for Dataset Info

We are mainly checking for the different data types each column possesses.



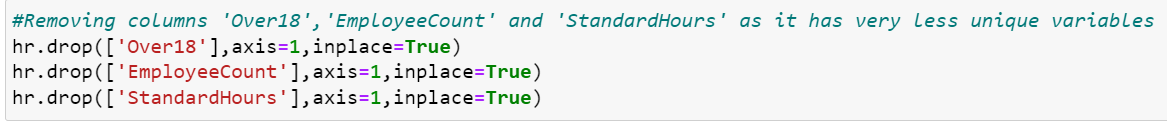
From the .info() function we see the various information like range, dtype, non-null count of the dataset.

We see that there 1470 rows and a total of 35 columns present in the dataset and from the Dtype column we see the various data types like object (str) type , int64 (int) type etc.

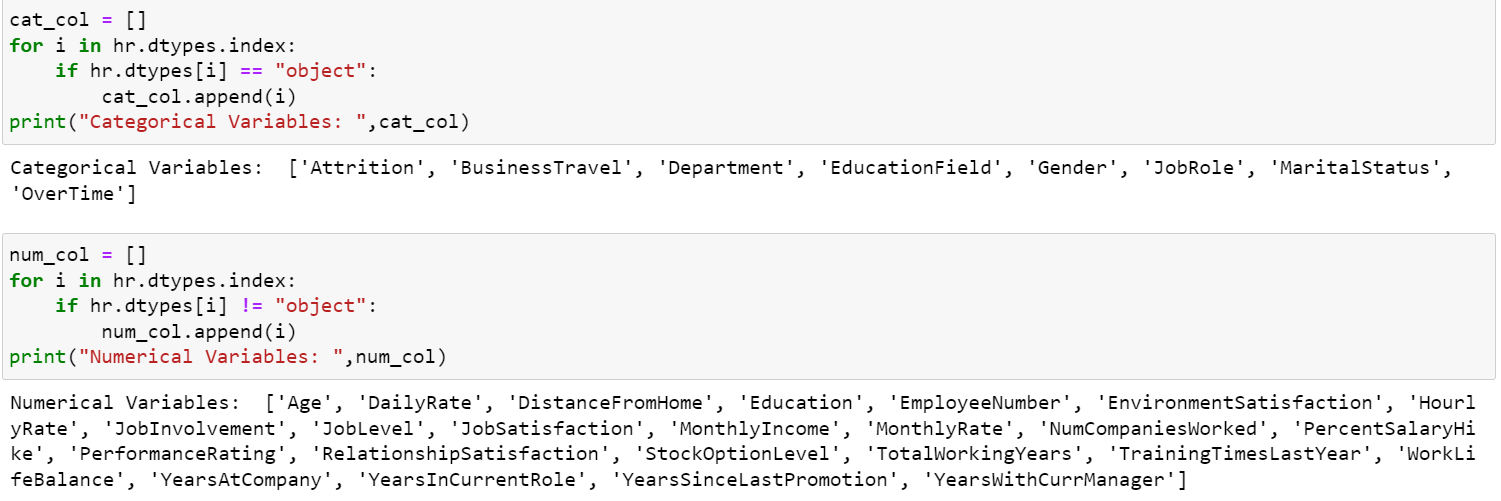
We also check for number of unique values present in the dataset using the code above. We can drop columns which have very less or very high unique values such as (columns like Over18, StandardHours and EmployeeNumber)

Removing unnecessary variables

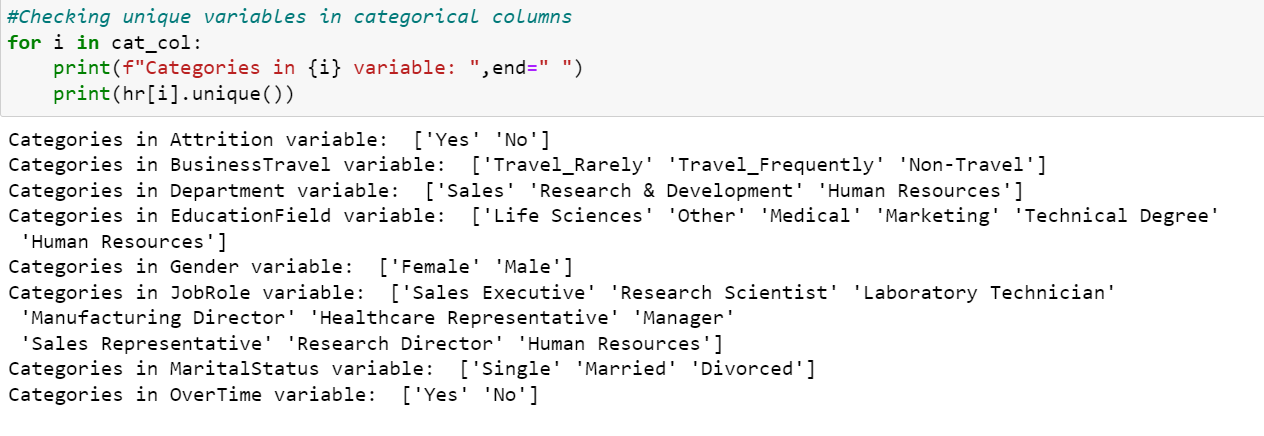


Now, we drop the columns which are having very less unique values.

Columns like ‘Over18’ and ‘StandardHours’ have only one unique value and also does not affect the analysis and prediction purpose.



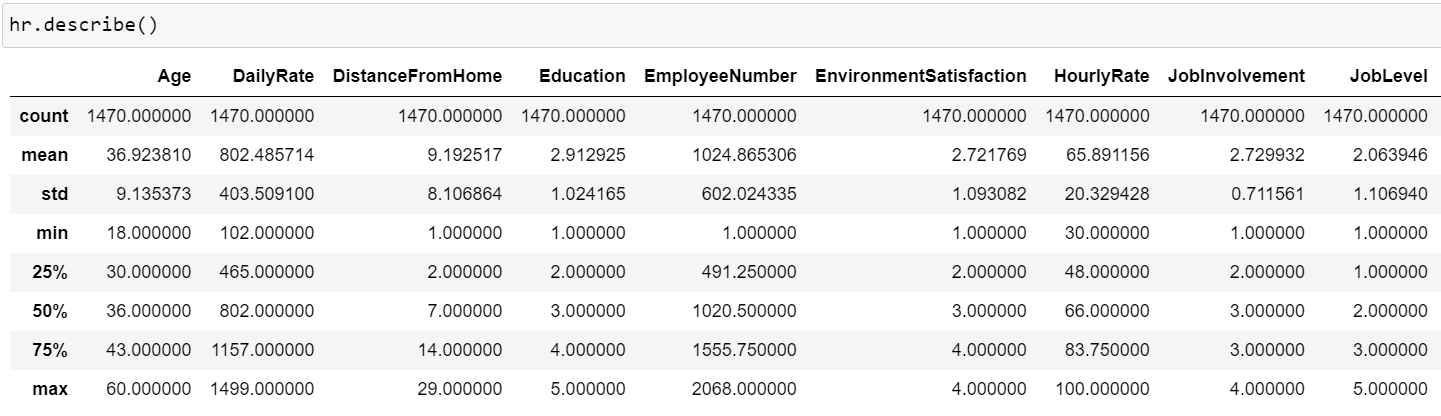
Now, we split the columns into categorical and numerical and store them in a list for further analysis purpose.

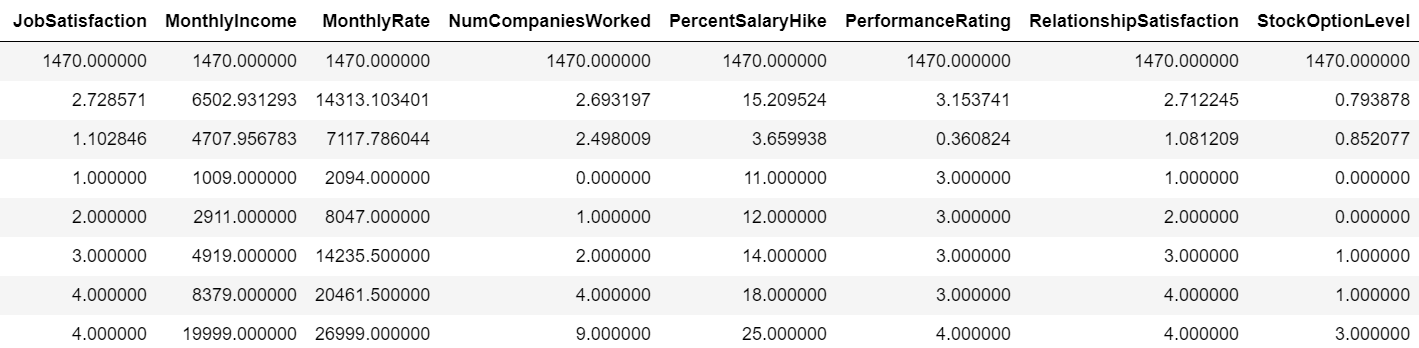


We can see the different unique variables present in the categorical columns.

UNIVARIATE ANALYSIS

a) Non-Graphical







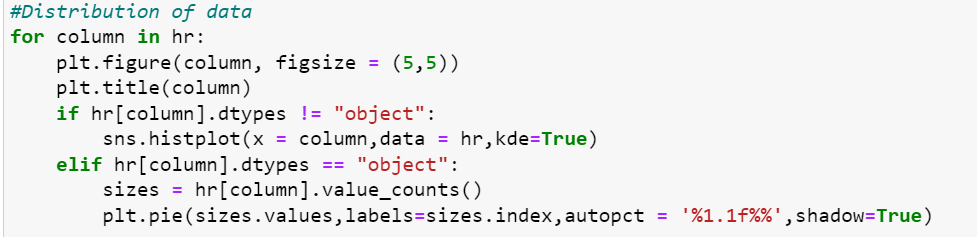
-- Take-aways:

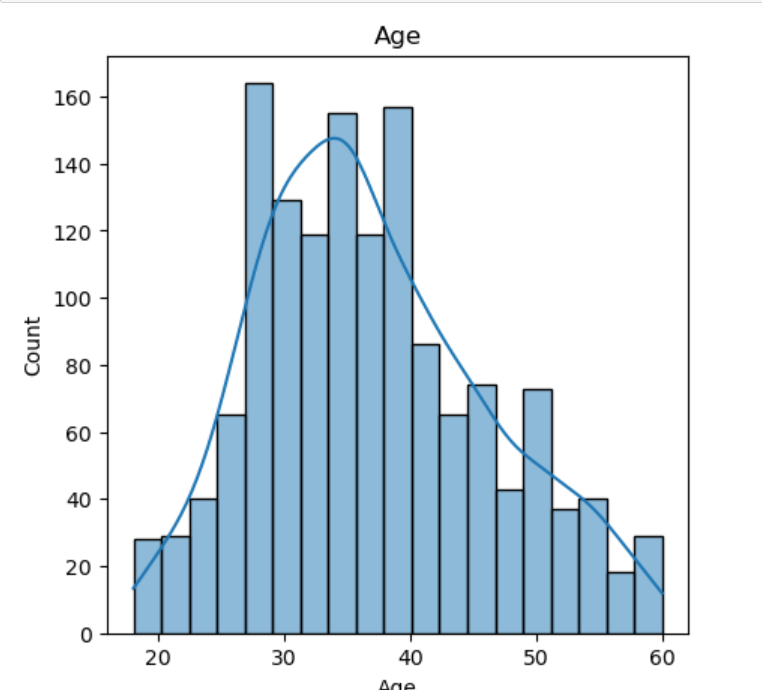
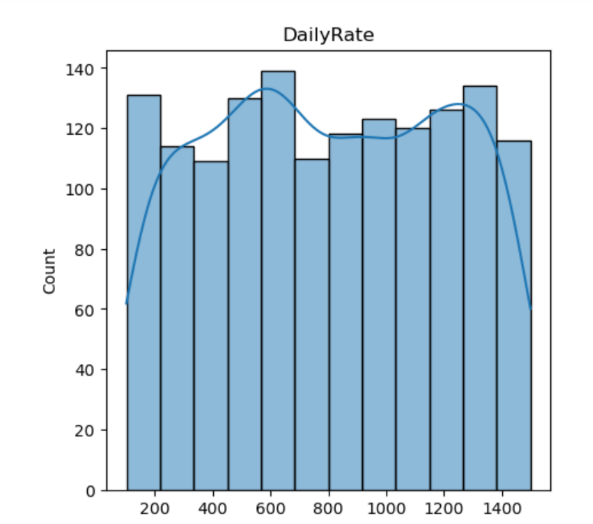
1. The mean age of employees in the given data is around 37 years.
2. The mean distance from home to the workplace of employees is around 9km.
3. The mean Job Satisfaction level (i.e rated from 1-5) of employees in the current company is around 2-3.
4. The average monthly income of employees in the current company ranges up to 6,500.
5. The average number of companies that the particular employee used to work for is around 2-3.. and the average years that the employee worked for is around 11 years..(2-3 companies in 11 years)..Therefore, the dataset consists of really experienced employees.
6. The average Performance Rating of employees given by the company is around 3 (out of 5)..which is a really good rate.
7. Average years that the employee has been working for this company is 7 years, and the average years of experience that the employee has in his current role is around 4 years. Therefore, employees are very reliable in this company and the HR's have done a really in retaining these employees for these many years..
8. There might be very light presence of outliers, as the difference between the maximum value and the third quartile is very less in these columns.

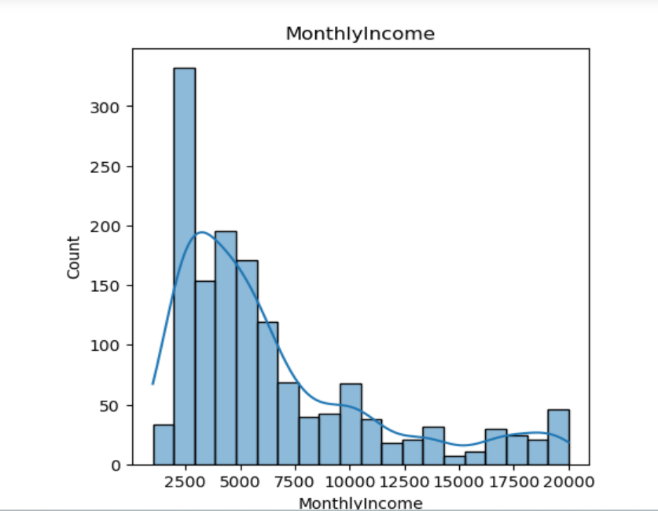
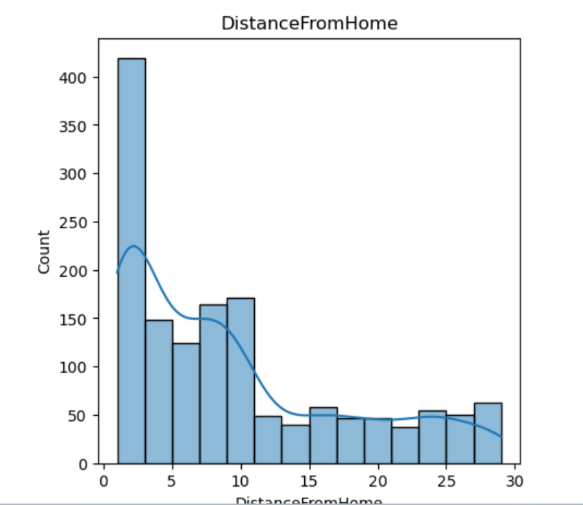
b) Graphical

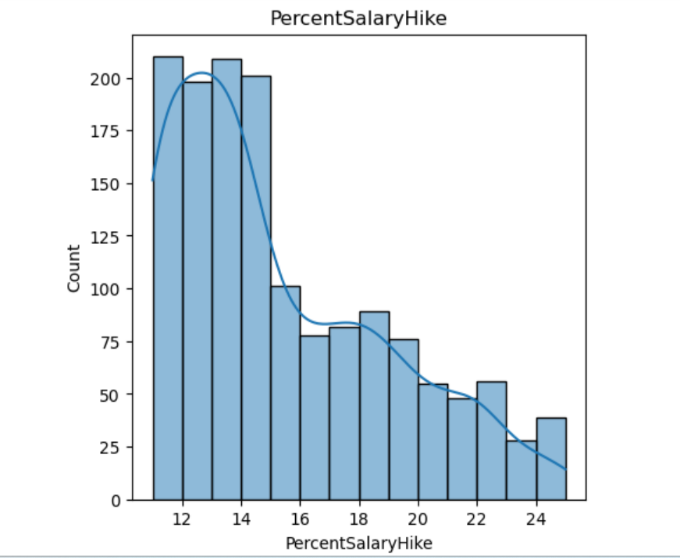
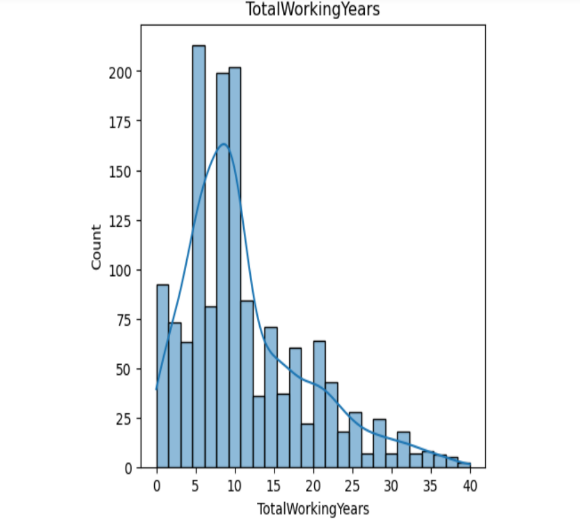
Now, that we have observed the Non-Graphical Univariate Analysis part let’s move on to the Graphical part of our Univariate Analysis ( i.e distribution of data across all the columns – both numerical and categorical)

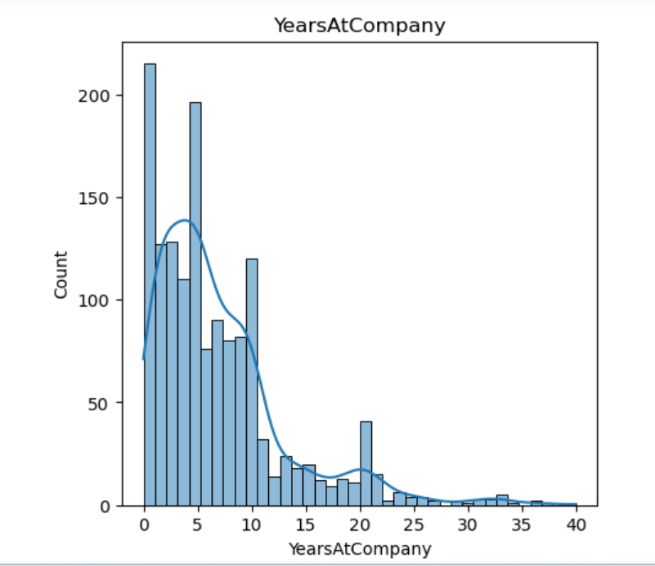
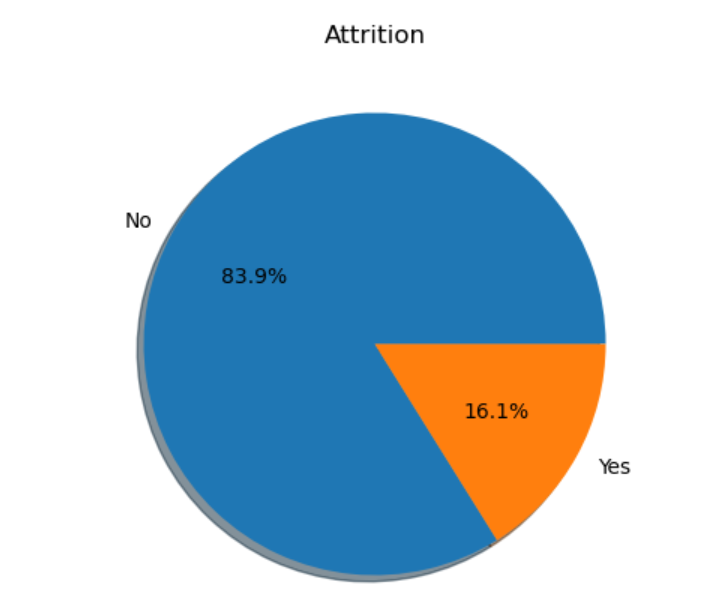
For the numerical columns, we shall be analysing our distribution of data through histogram plots. For the categorical columns, we shall be analysing our distribution of data through Pie Charts.

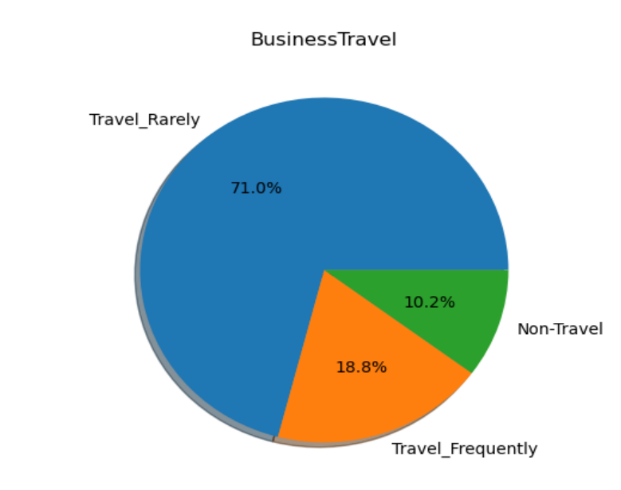
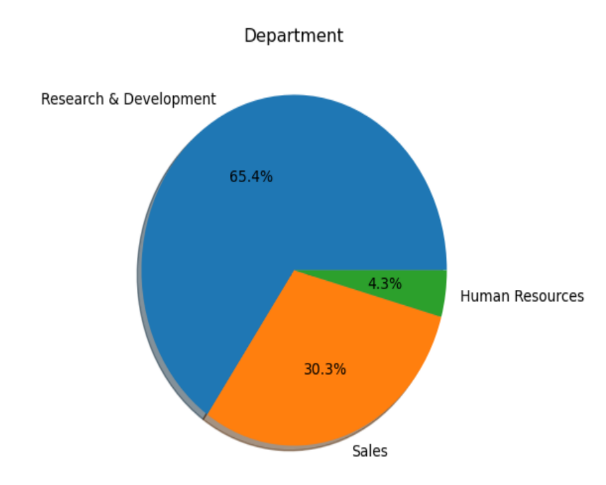


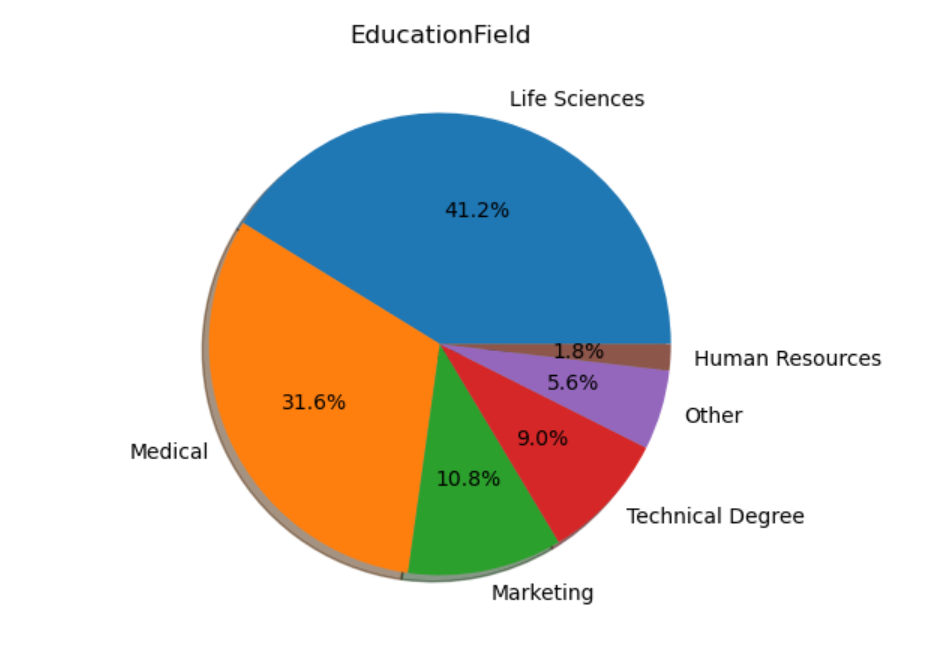
 

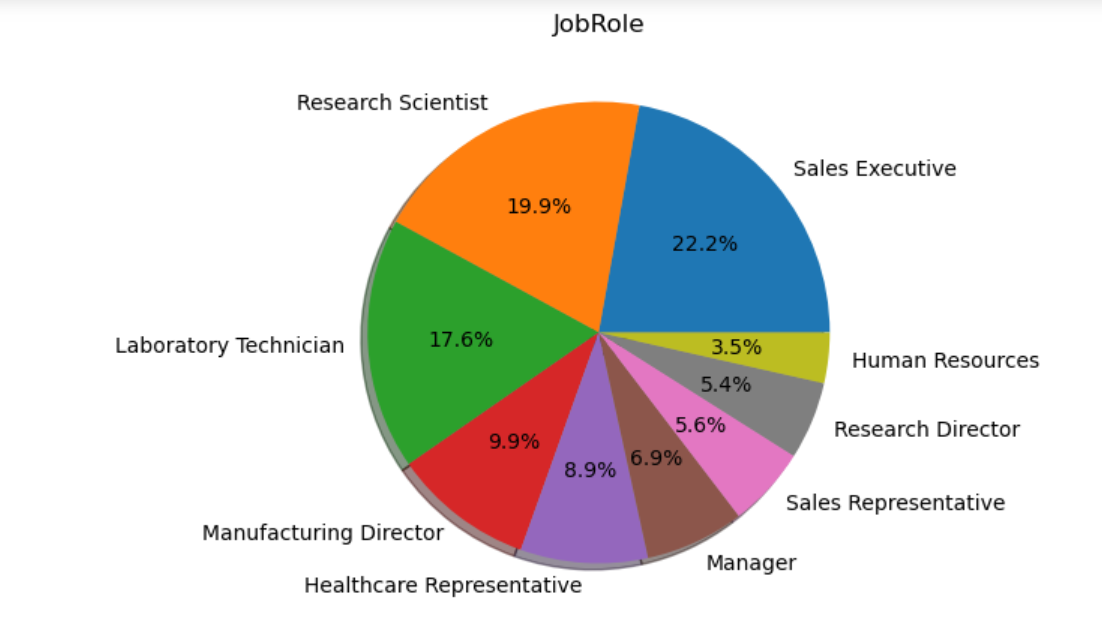
 

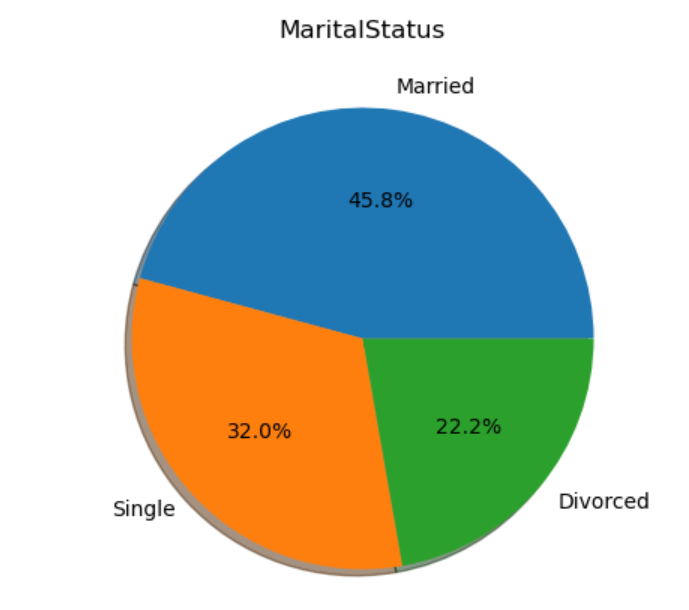
 







These above images are some of the outputs of the code snippet given above, which gives us the distribution of data across both numerical and categorical variables.

From these images:

1. As we can see from the distribution of both numerical and categorical columns in the data, Numerical columns = histplot and Categorical columns = pie charts
2. Numerical columns have histplot as their plot for showing the distribution of data and also the skewness of data, most of the numerical columns are right skewed (i.e mean > median) almost 9 columns and the rest are either left skewed or perfectly distributed (i.e mean = median, columns like age,Daily rate)
3. Categorical columns have pie chart as their plot for showing the distribution of data

a) Employees classified with buisness travel as 'rarely' are much in number than non-travelling and frequent travellers.

b) Employess in the department of Research & Develpoment are much higher than employees working in Sales and Human Resources.

c) Percentage of employees not working overtime is way higher than those who are working overtime.

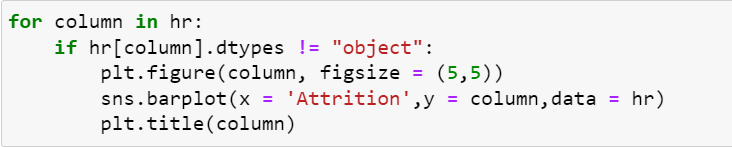
d) The percentage of employees in the Sales Representative Job role in the highest in the dataset, with Research Scientist job role coming at a close second.

BIVARIATE ANALYSIS

Now we shall see how the variables(features) are related to each other and what can we derive from them. Also we shall check how the independent variables are correlated to the dependent feature.

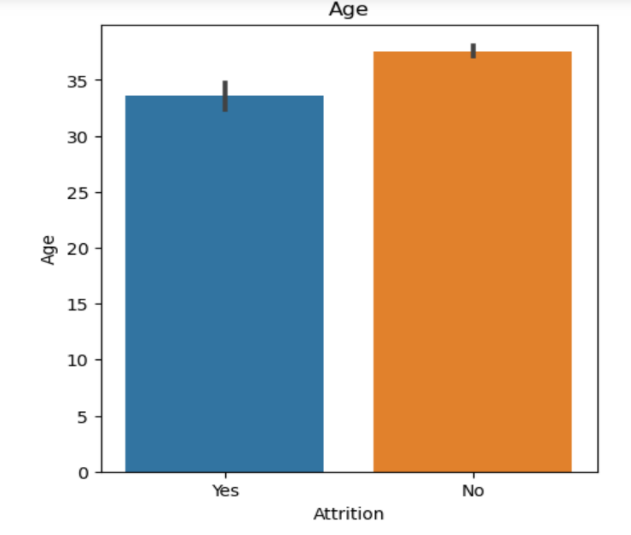
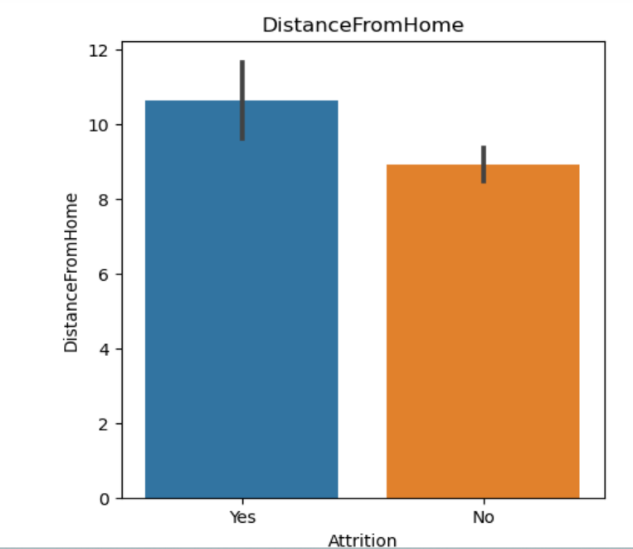
a) Relation between numerical features and dependent feature

Code snippet:



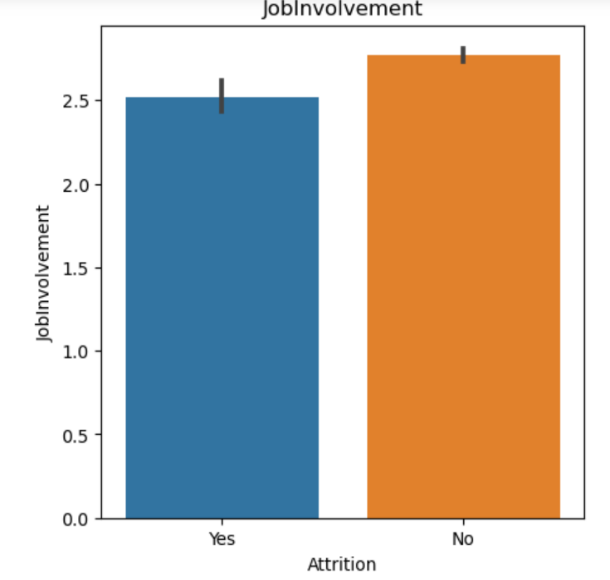
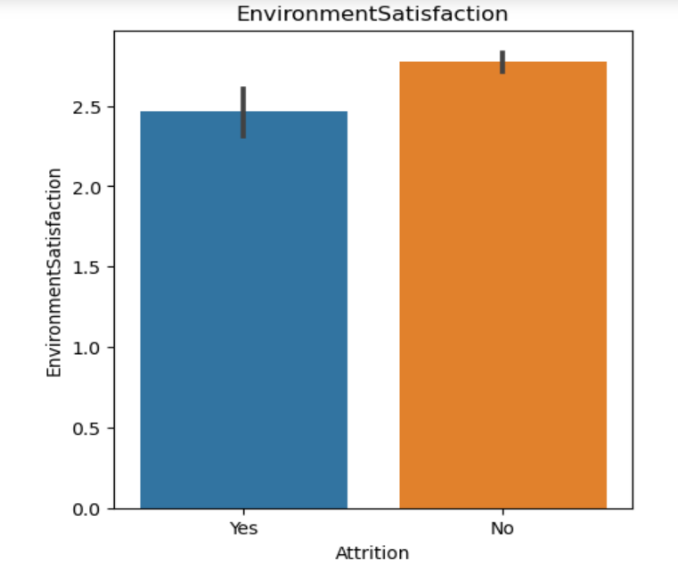
The above code gives us the realtion between numerical features and dependent feature with the help of barplots with dependent feature on the x-axis and numerical features in the y-axis.

Output:

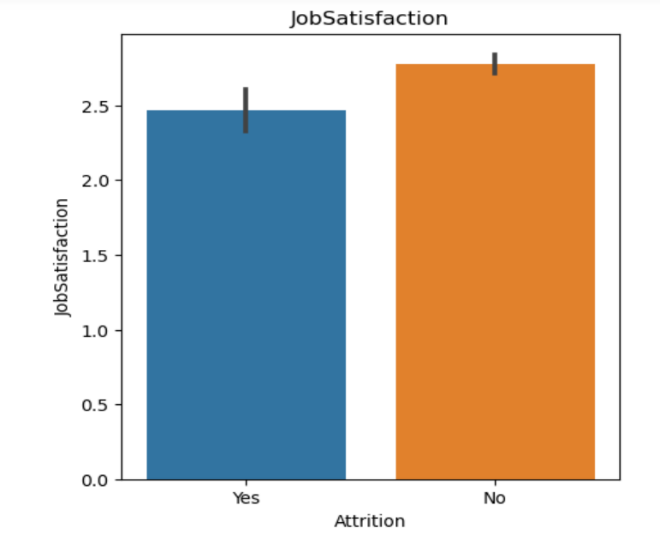
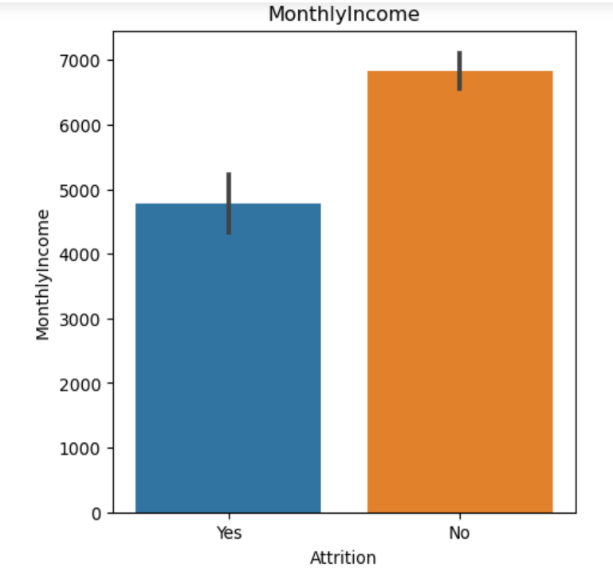
Employees above the age of 35 are tend to not attrition (or leave the company) more than Employees under the age of 35.

And Also employees living > 9 km from workplace are tend to attrition more than the employees living < 9 km.



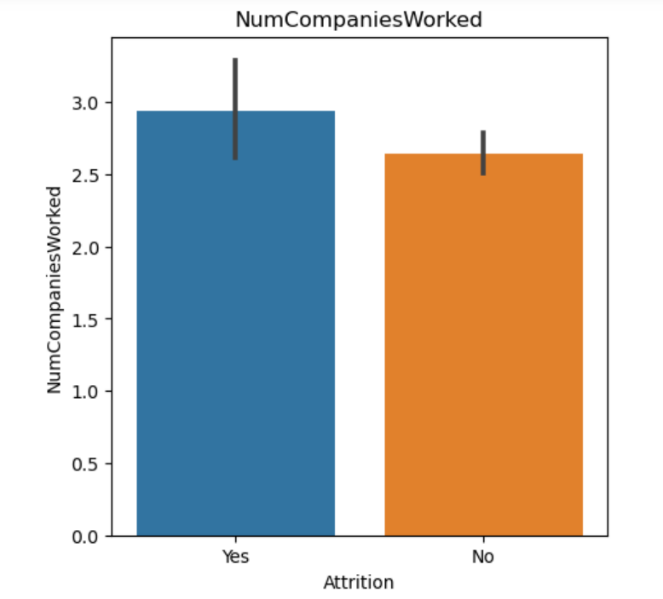
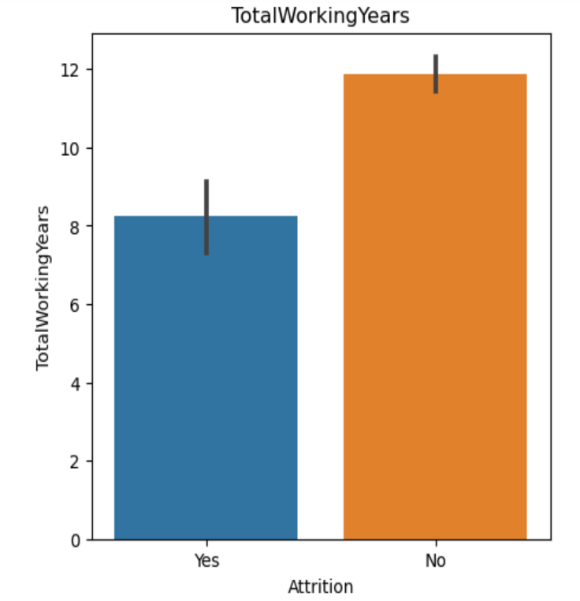
Employees who voted for the environment satisfaction rating (1 to 5) of more than 2.5 are tend to not attrition (or leave the company) more than Employees who voted for the rating less than 2.5.

And again same with the Job Involvement rating too. Employees with the Job Involovement rating of more than 2.5 are tend to not attrition from the company than the employees with the Job Involvement rating of less than 2.5.

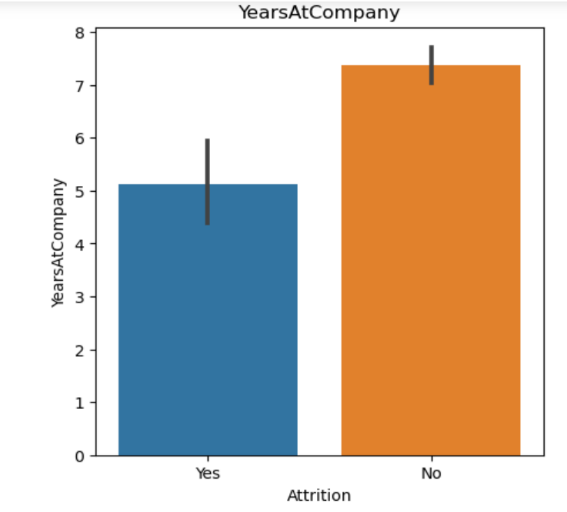
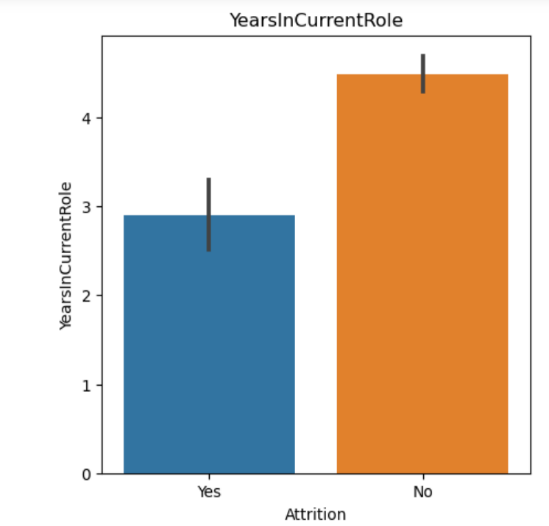
Employees who voted for the job satisfaction rating (1 to 5) of more than 2.5 are tend to not attrition (or leave the company) more than Employees who voted for the rating less than 2.5.

Employees with a monthly income less than 5500 are tend to leave the company more than the Employees with a monthly income of greater than 5500.

Employees who have worked for more than 3 companies are more likely to attrition from their current company than the Employees who have worked for less than 3 companies.

But, If an Employee has worked more than 9 years he/she is more likely not to attrition from their current company than the Employees who have worked for less than 9 years.

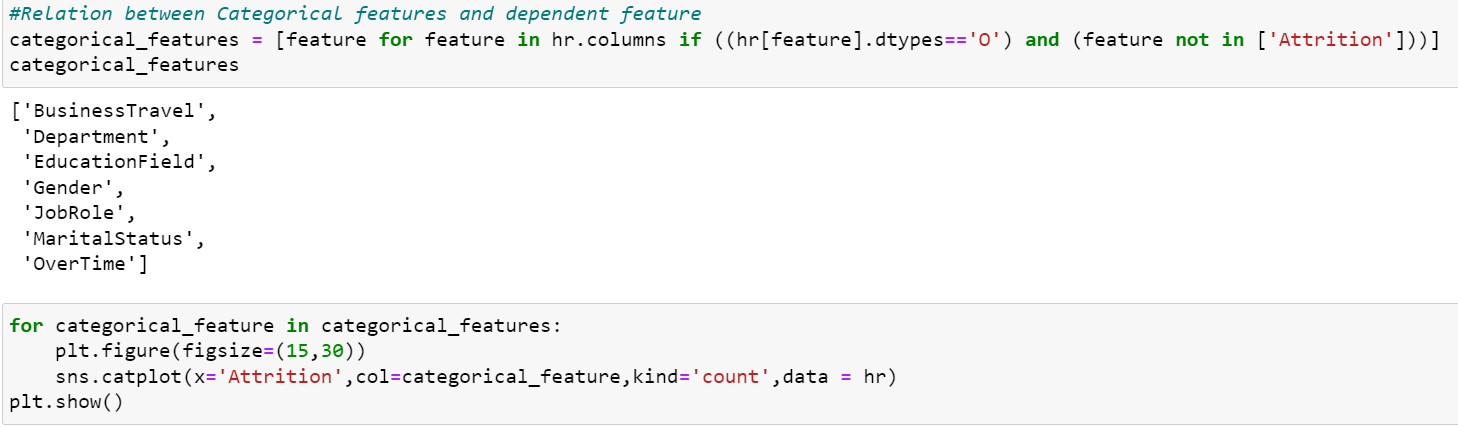
Employees staying in the current company for more than 5-6 years are more likely to not leave the company than the Employees staying in the current company for less than 5-6 years.

Employees working in the same current role in the company for more than 3 years are more likely to not leave the company, than the employees working in the same current role in the company for less than 3 years.

Now, these were some of the examples of the relation between numerical features and the dependent feature.

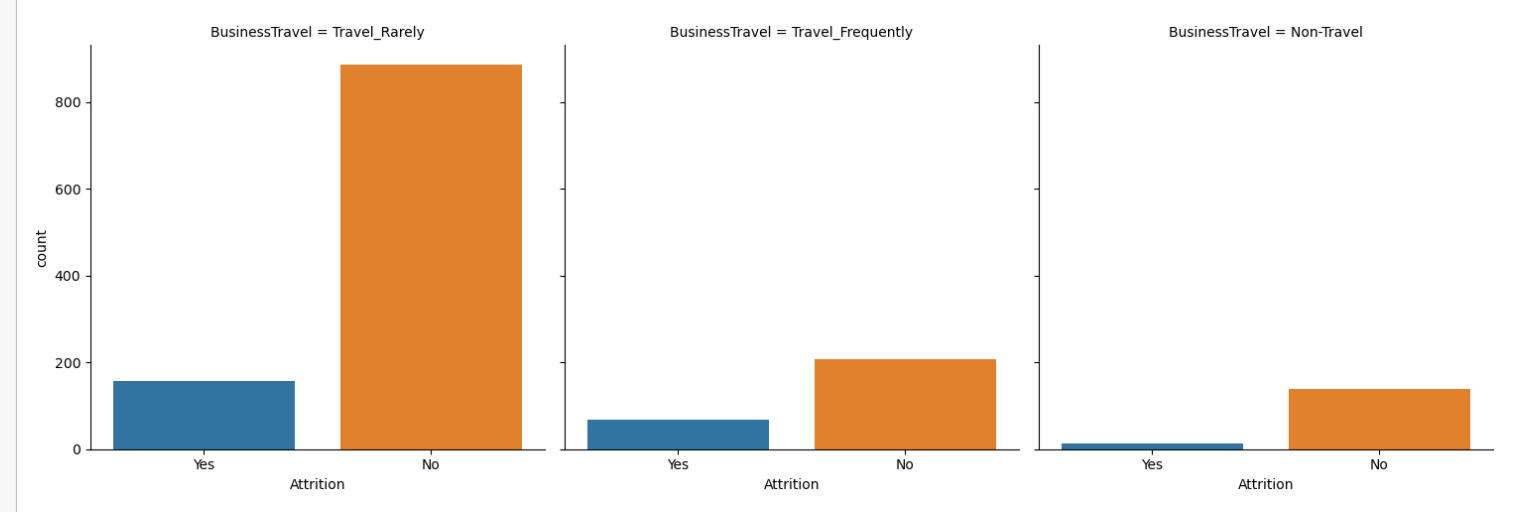
b) Relation between categorical features and dependent feature

Code Snippet:

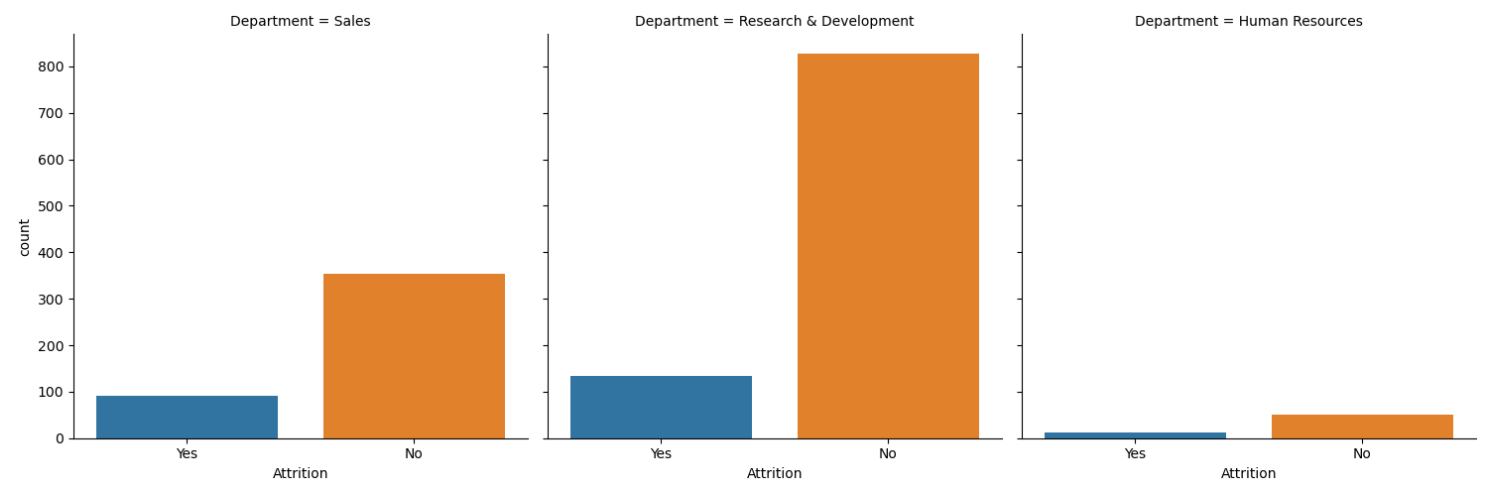


From the above code, we need to divide the features into categorical features and use these categorical features to find the realtion between them and dependent feature using catplots.

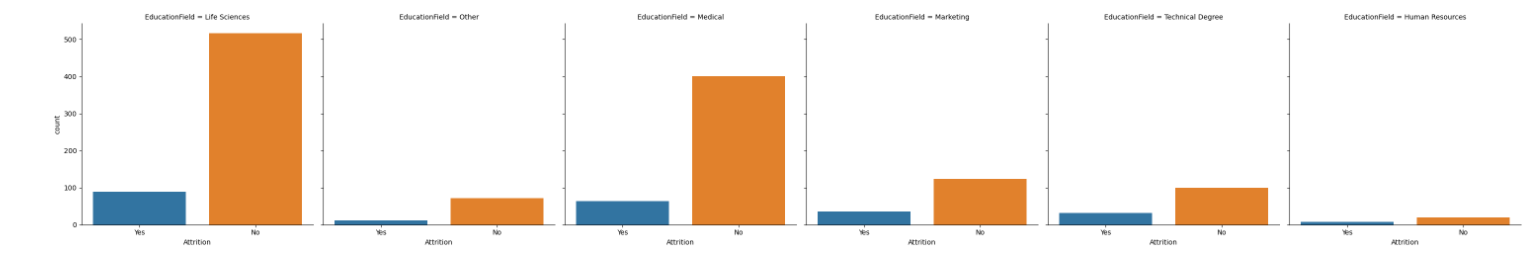
Output:



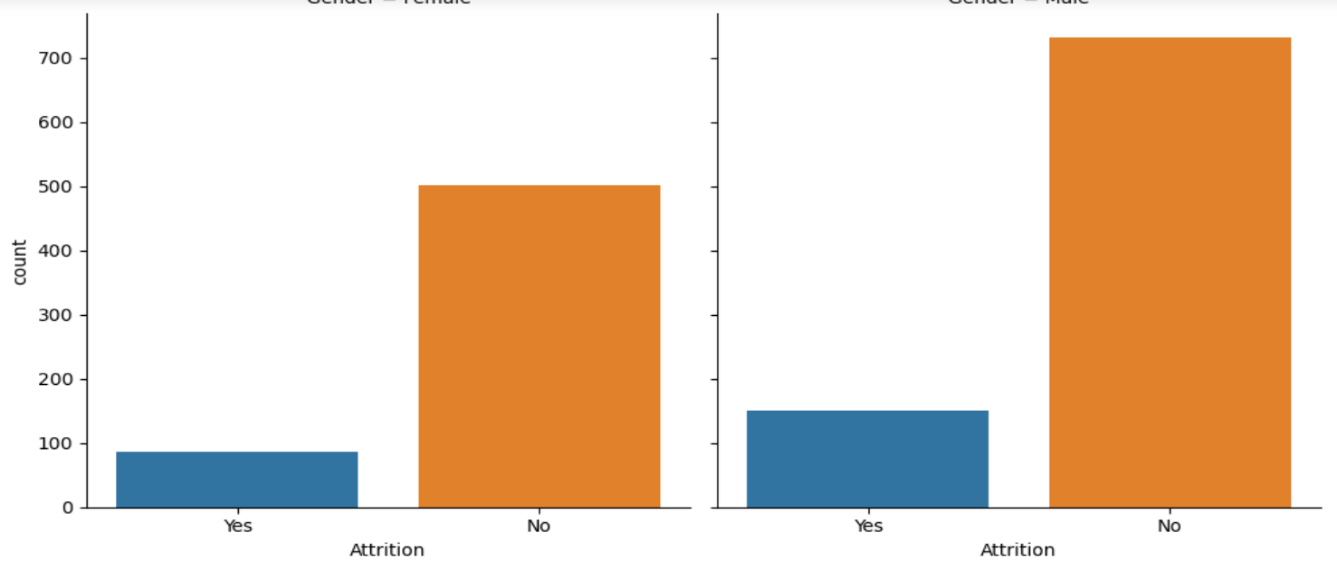
Categorical feature – Buisness Travel (Travel\_Rarely , Travel\_Frequently , Non- Travel)



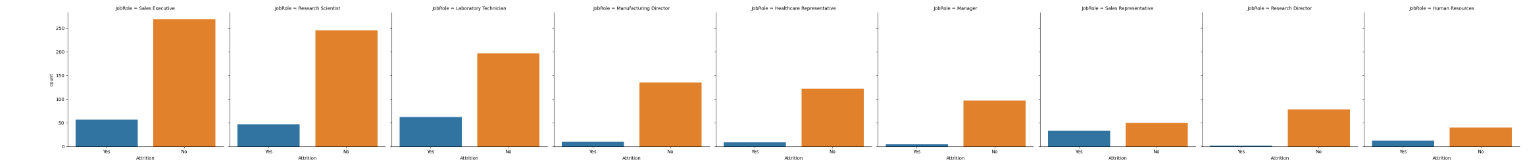
Categorical Feature – Department (Sales , Research and Develpoment , Human Resources)



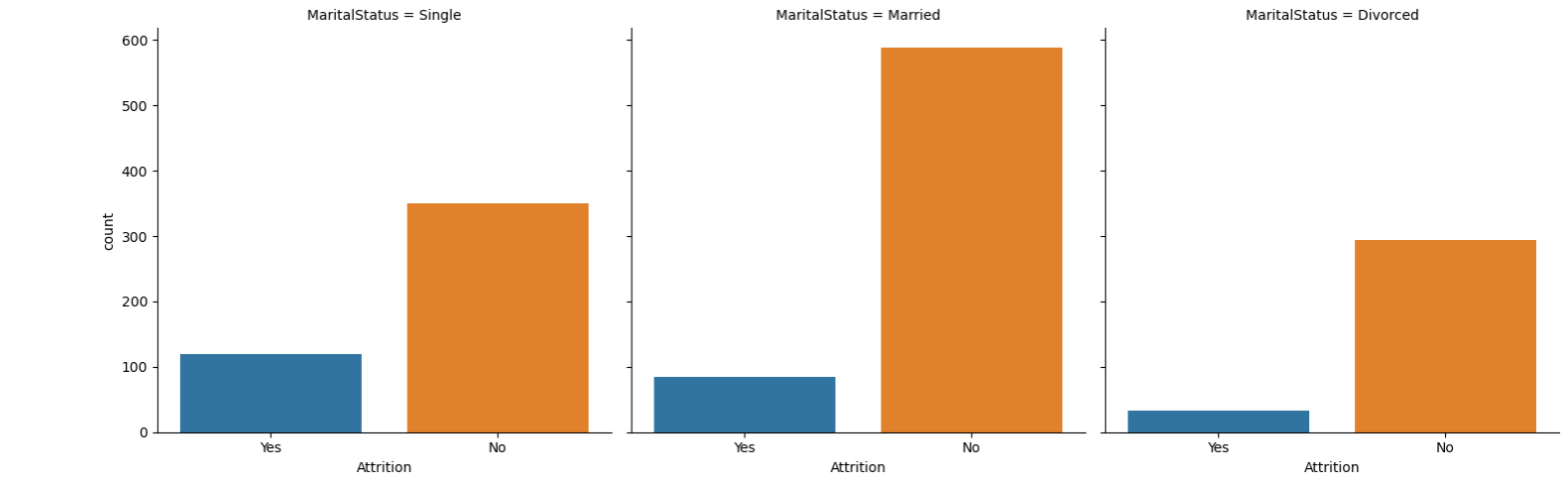
Categorical Feature – EducationFeild (Life Science , Other , Medical , Marketing , Technical Degree , Human Resources)



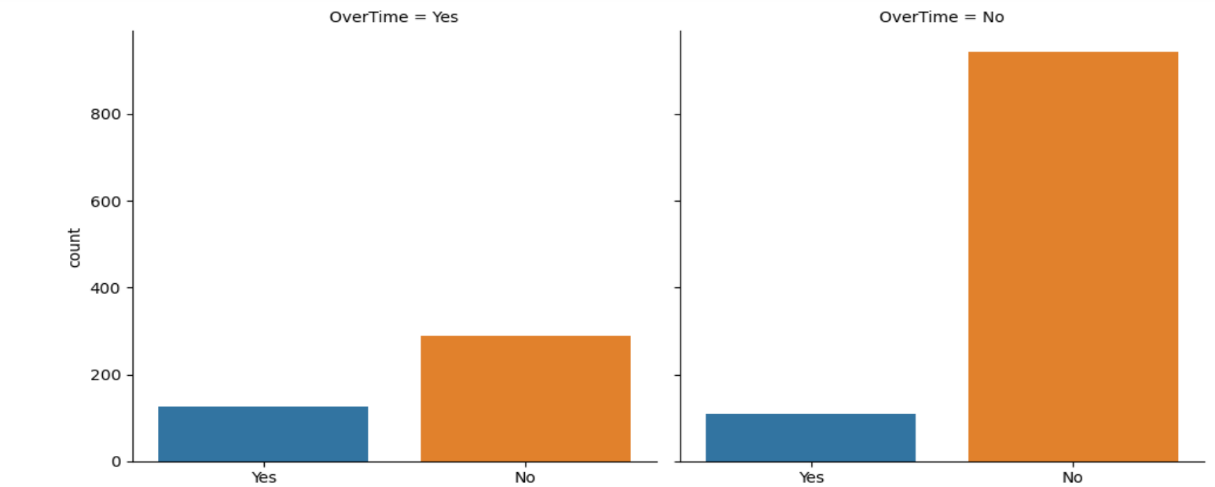
Categorical Feature – Gender (Male , Female)



Categorical feature – Job Role (Sales Executive , Research Scientist , Laboratory Technician , Manufacturing Director , Healthcare Representative , Manager , Sales Representative , Research Director , Human Resources )



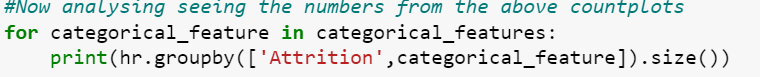
Categorical feature – Marital Status ( Single, Married , Divorced)



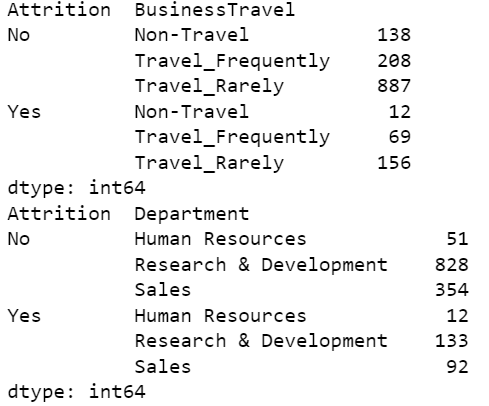
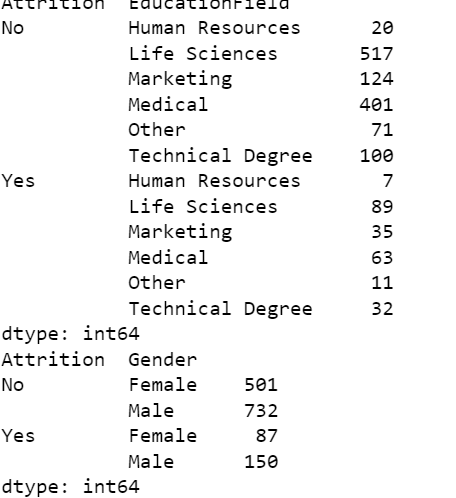
Categorical feature – OverTime (Yes , No)

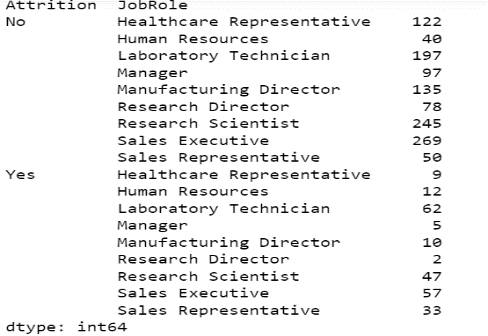
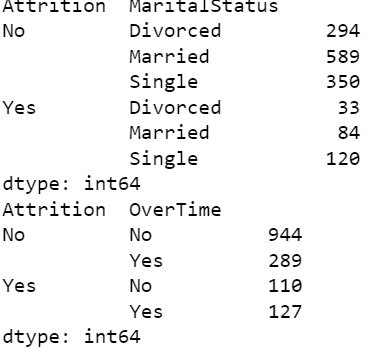
We can analyse these above realtion between categorical and dependent feature by raw numbers too for better understanding.

Code:



Example output:

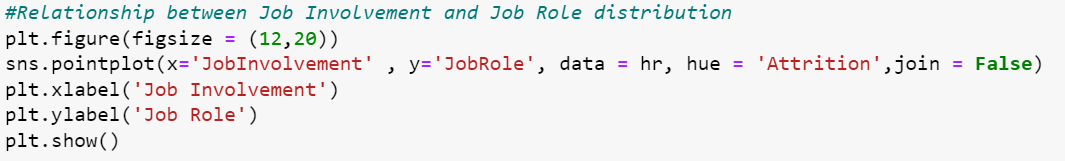
 

MULTIVARIATE ANALYSIS

Understanding Relation between one or more independent features with the dependent feature.

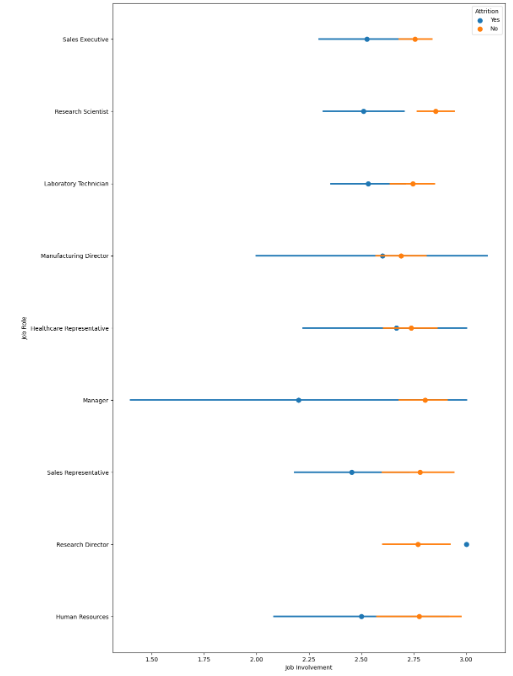
In this section, we are going to see the relation between one or more independent features with the dependent feature using pointplot.

i) Code:



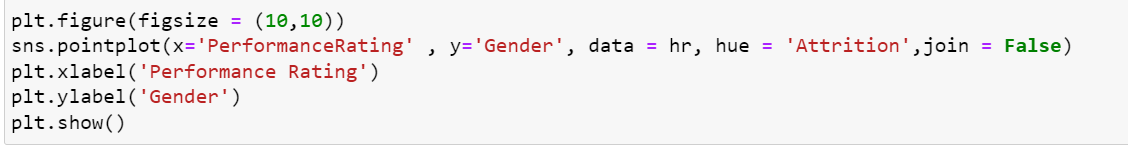
Here we can see the relation between two independent features ‘JobInvolvement’ and ‘JobRole’ with the dependent feature(‘Attrition’) as our reference.

Output:



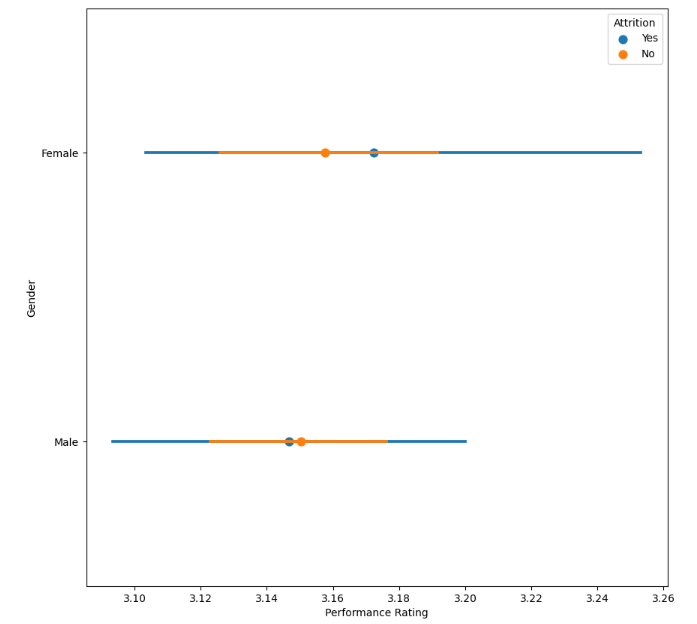
Employees with more Job Involvement are less likely to leave the company (or less attrition rate), mainly Job Roles like Sales Executive,Research Scientist etc..

ii) Code:



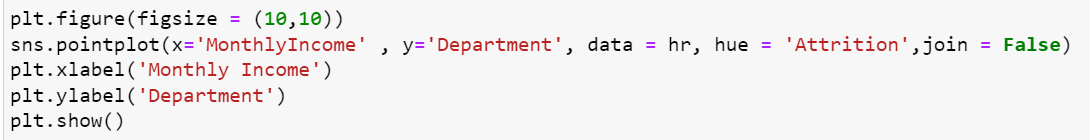
Here we can see the relation between two independent features ‘PerformanceRating’ and ‘Gender’ with the dependent feature(‘Attrition’) as our reference.

Output:



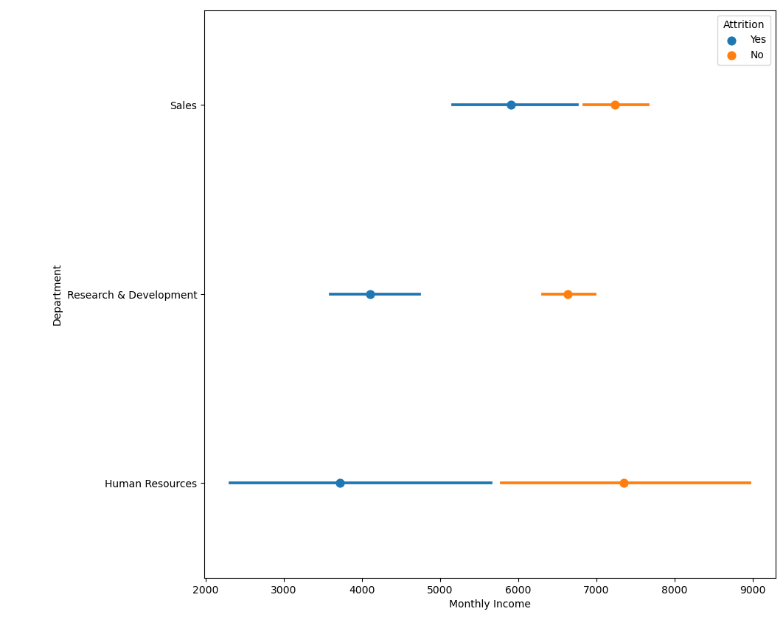
 Employees with high Performance Rating are likely to leave the company, as they feel like they are doing more to the company but are either getting paid less or some other problems. Eventually, Female Employees have higher Performance Rating than Male Employees and have higher attrition than men.

iii) Code:



Here we can see the relation between two independent features ‘MonthlyIncome’ and ‘Department’ with the dependent feature(‘Attrition’) as our reference.

OUTPUT:



Employees with high monthly Income are less likely to leave the company(less attrition rate). And the department that offers high monthly income are the Human Resources department.

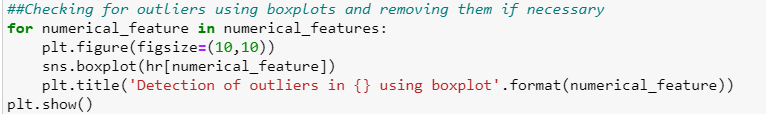
DATA PRE-PROCESSING (PRE-PROCESSING PIPELINES)

Step 1 :

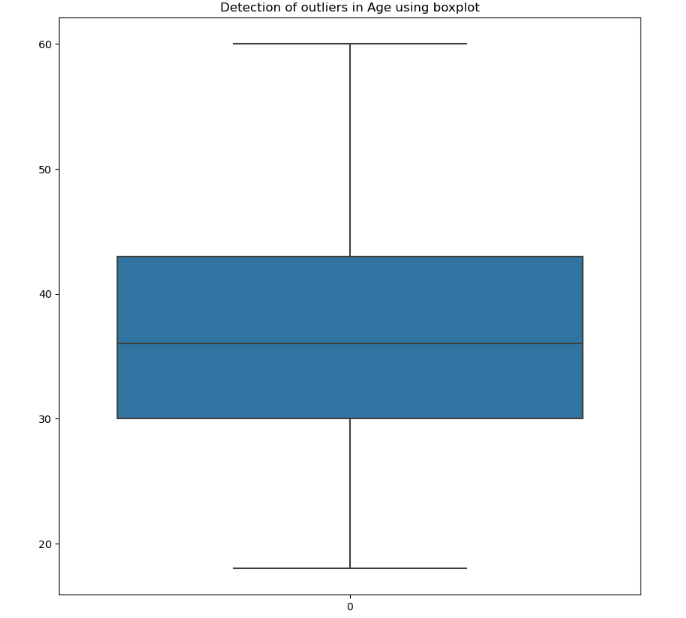
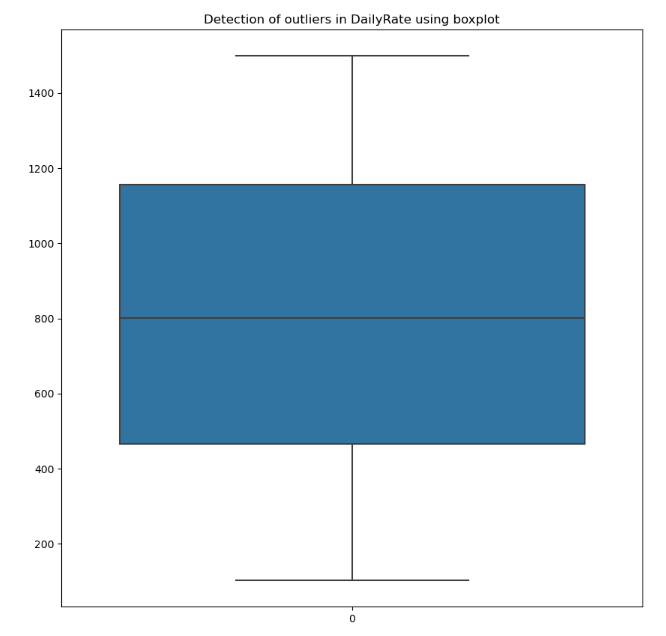
CHECKING FOR OUTLIERS

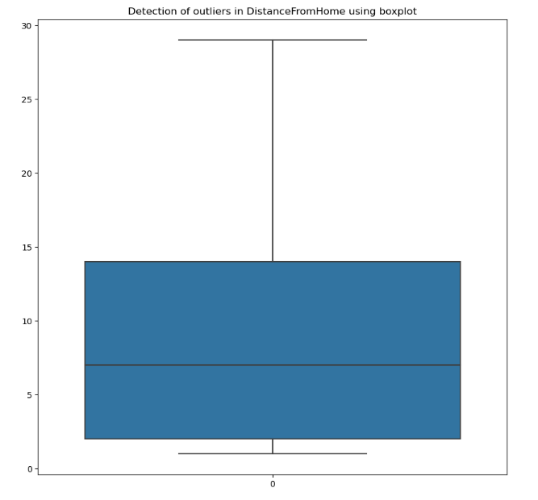
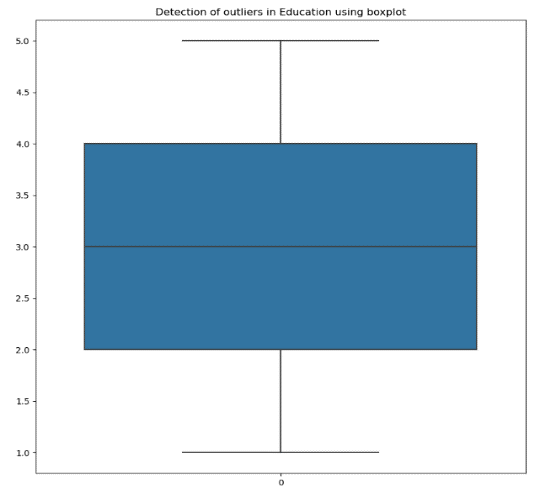
We can check for outliers using various methods, but here we are going to check for outliers using boxplot for each numerical feature and treat them (if necessary)

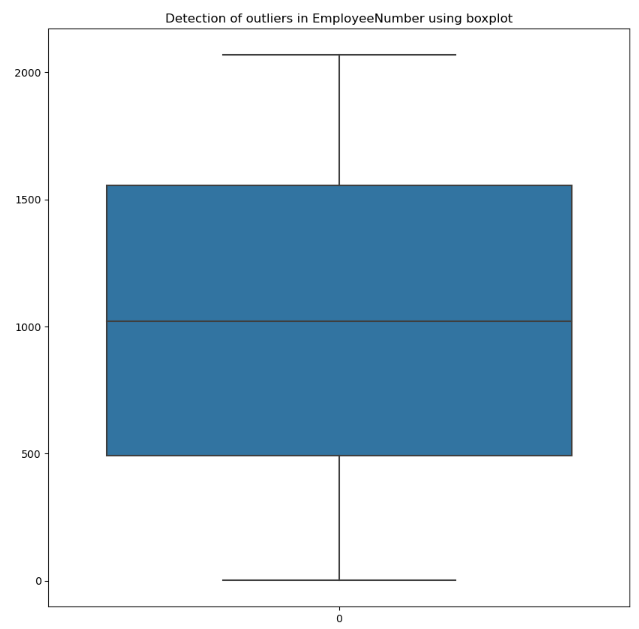
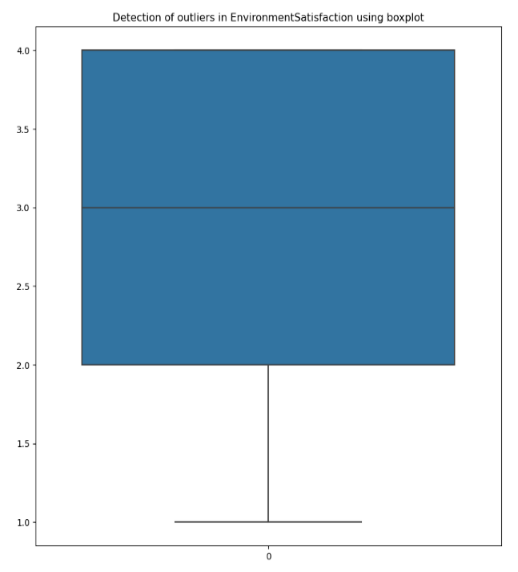
Code:

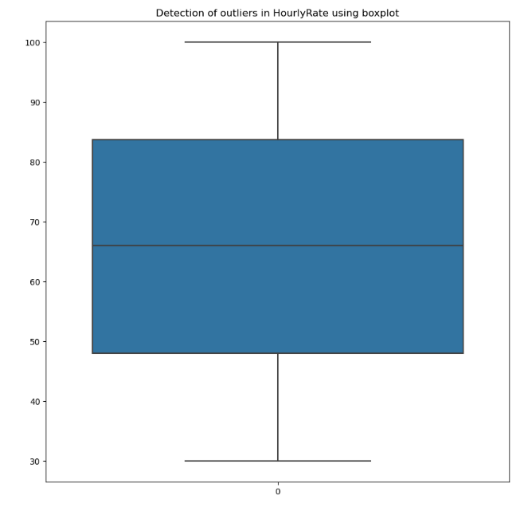
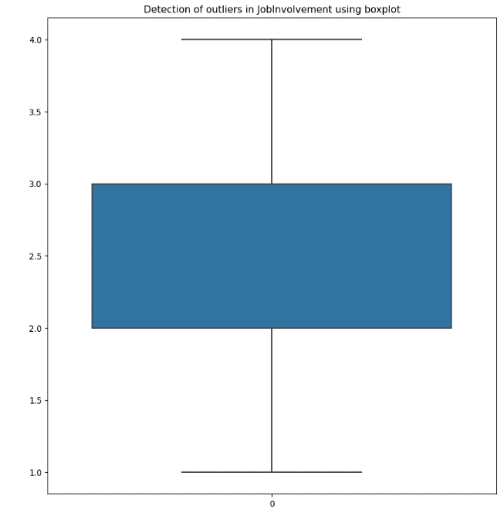


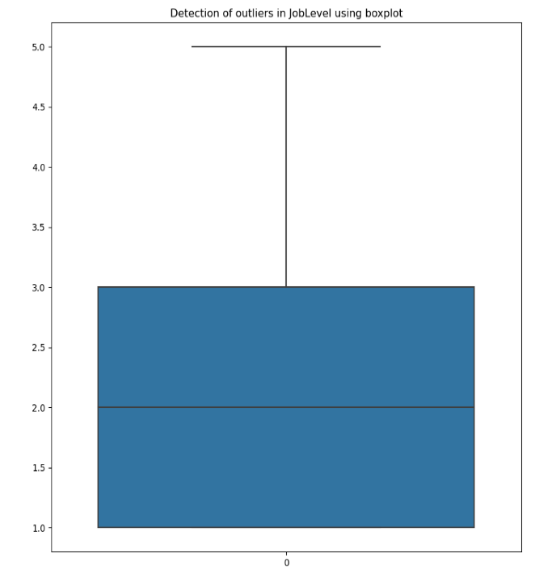
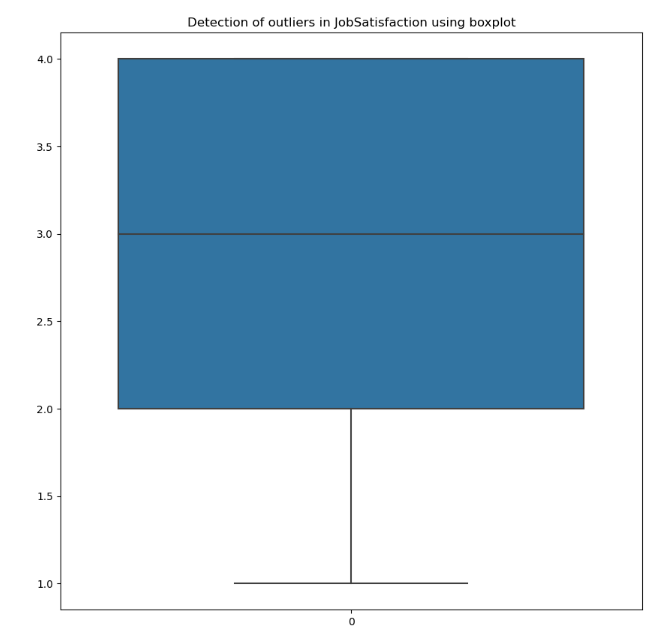
Output:

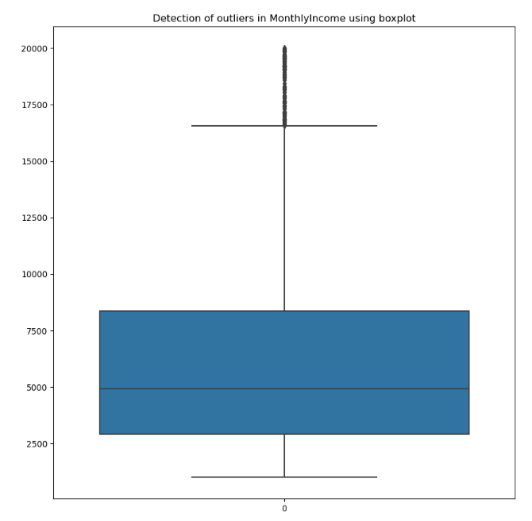
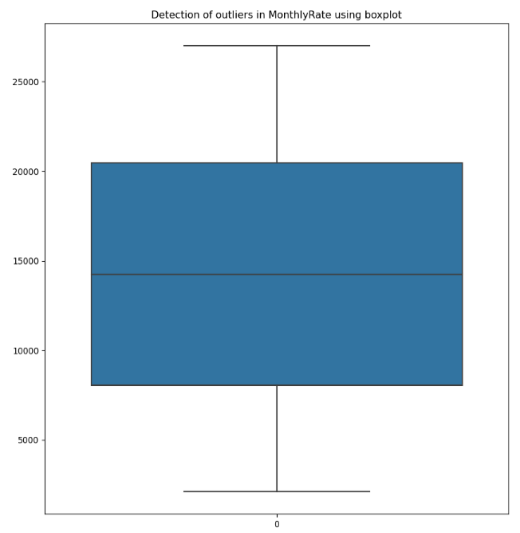
 

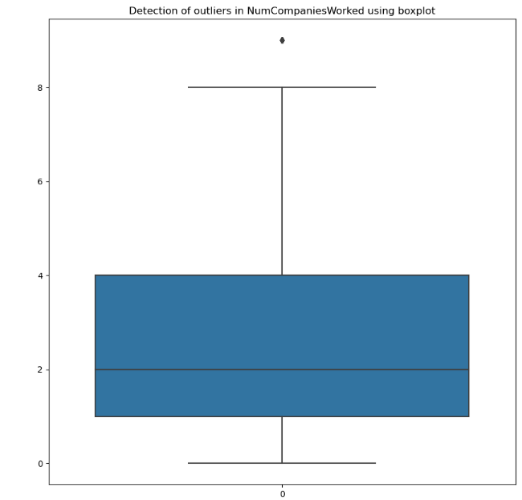
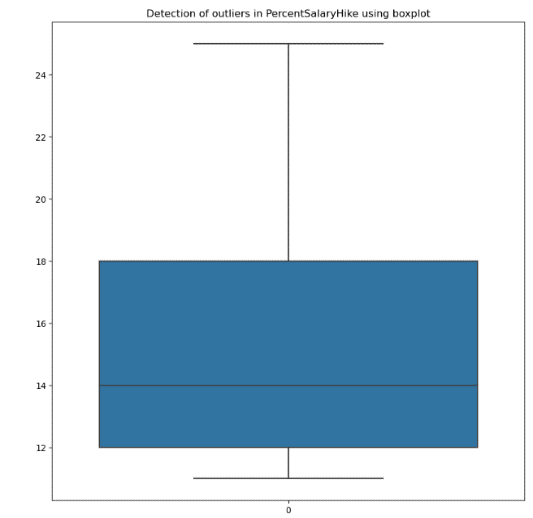
 

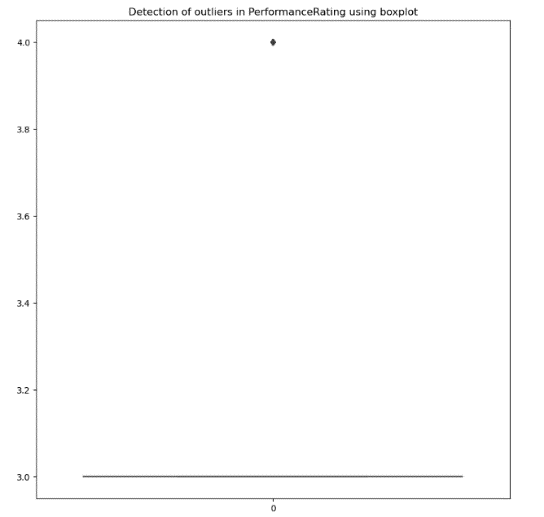
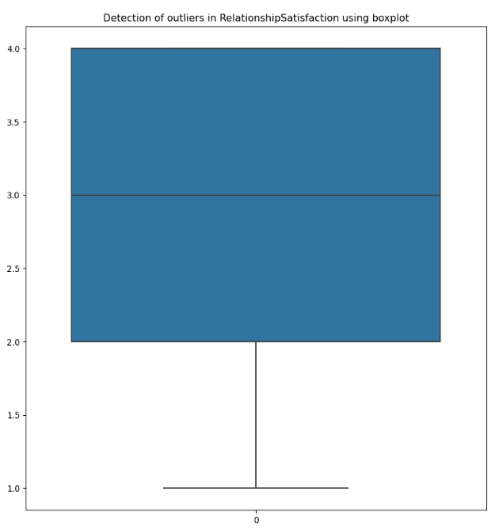
 

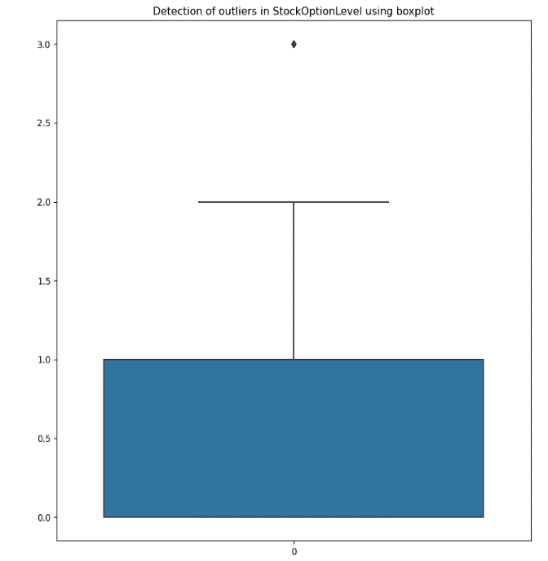
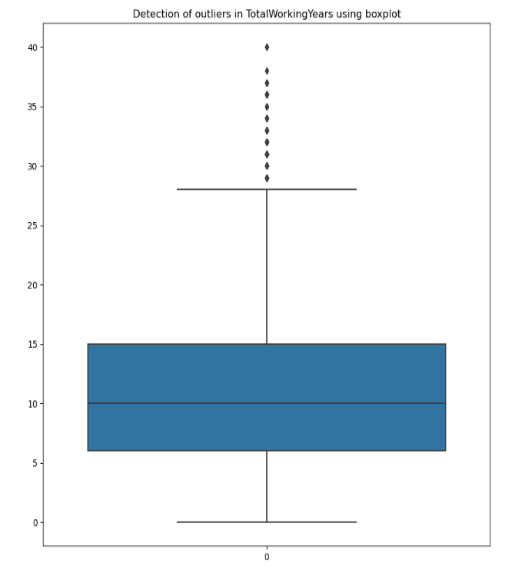
 

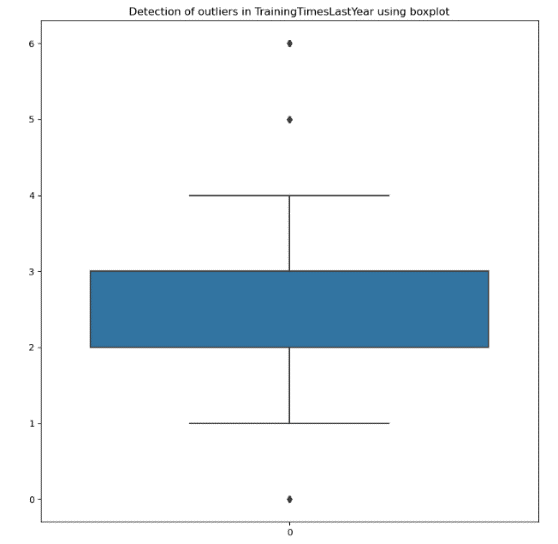
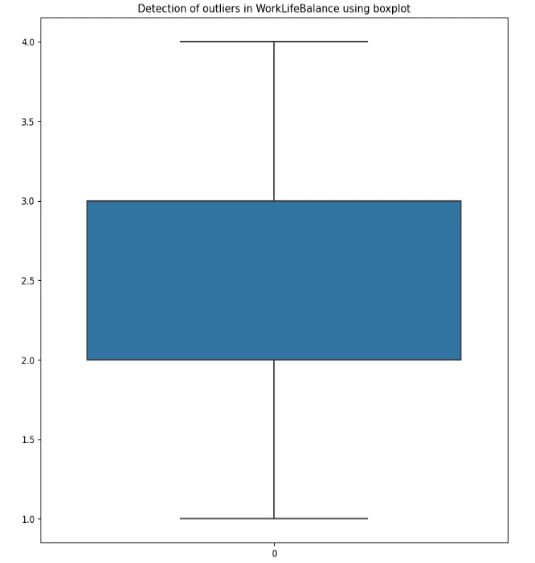
 

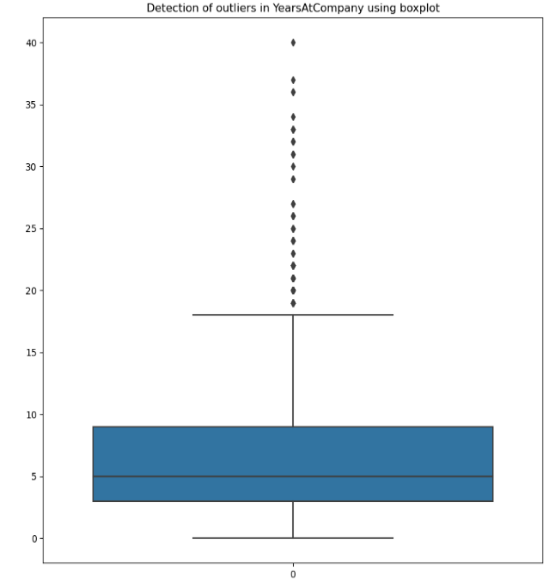
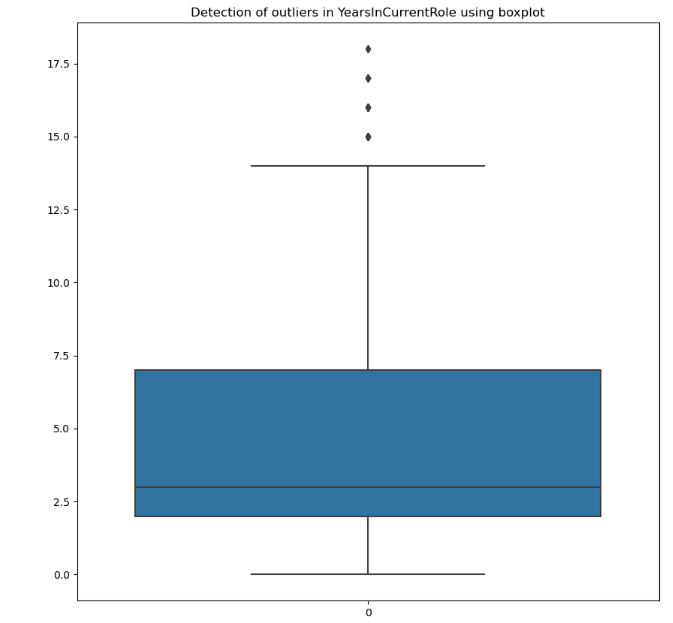
 

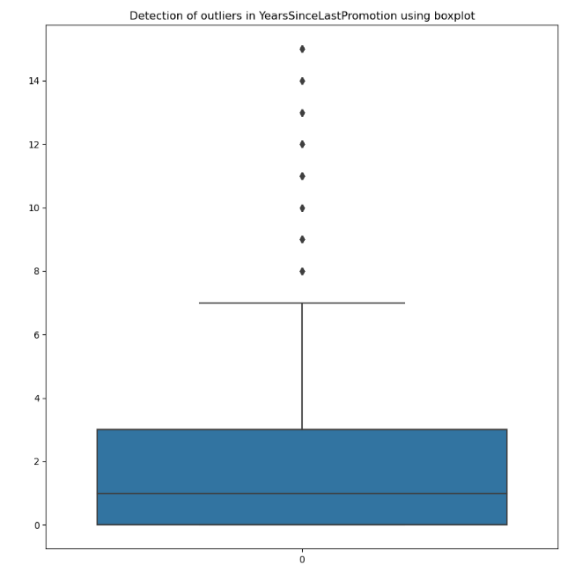
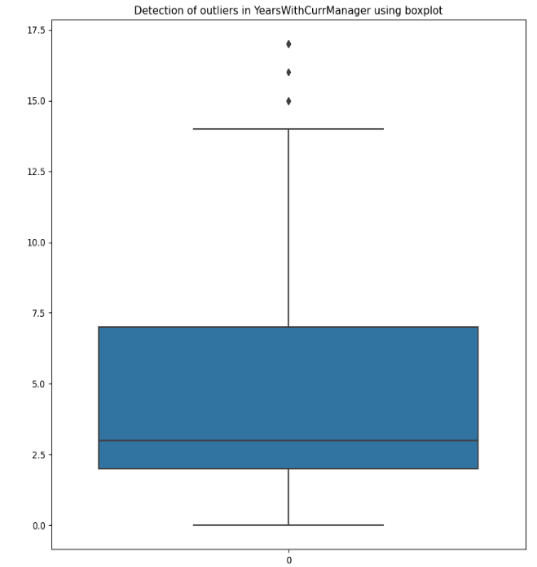
 

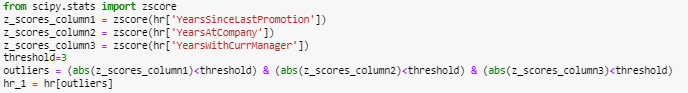
 

These are all the boxplots of all the numerical columns in the dataset. As you can see there are some outliers present in some features, but we are going outliers from those columns which are not realistic or which affect our prediction process.

We will be removing outliers in the following columns as some data entries does not look realistic :

1. 'Years at Company'
2. 'Years Since Last Promotion'
3. 'Years with current manager'

We will be removing outliers using the z- score method , the code is as follows:



Outliers are removed successfully.

We remove the column ‘EmployeeNumber’ as it is not needed for our prediction process.



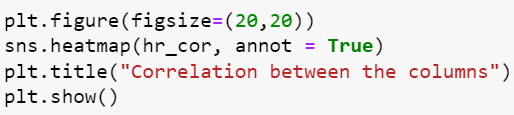
Step 2 :

CORRELATION ANALYSIS

Now, we check correlation among the numerical columns and mainly check for multicollinearity among these columns and we treat the columns which have multicollinearity by dropping those respective columns cause as we know multicollinearity reduces the precision of the model.

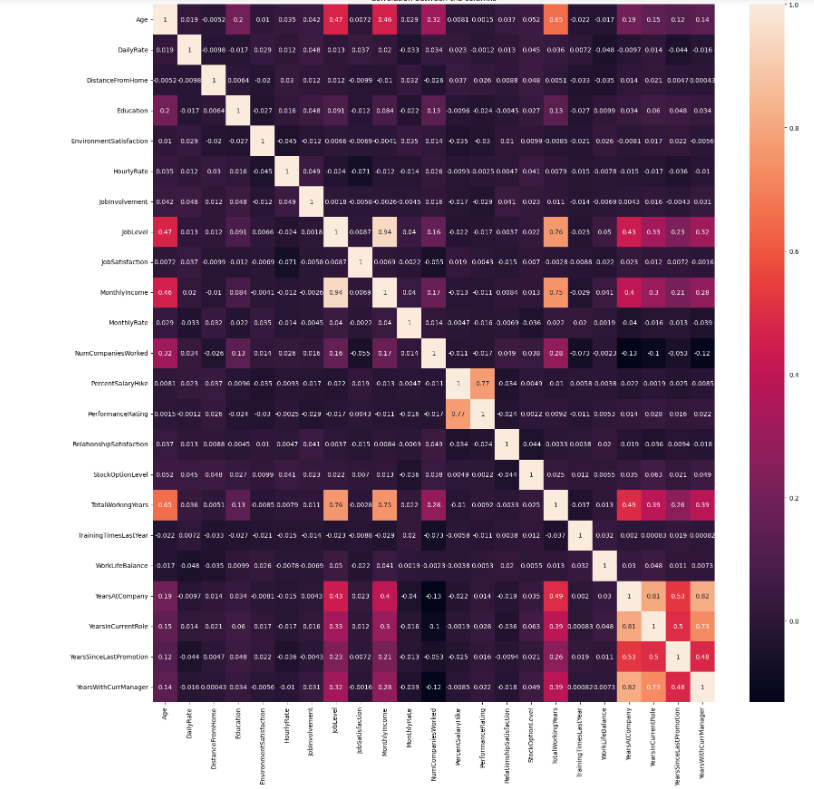
Code:





Output:

Checking correlation using heatmap, with the light shaded color indicating positive correlation and the dark shaded color indicating negative correlation.



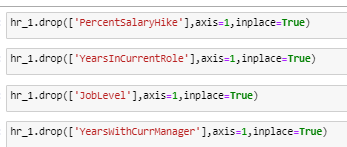
There are signs of multicollinearity present in the dataset.

Therefore,

We will be removing the column 'PercentSalaryHike' as it is multicollinear and this particular feature does not have much effect on the prediction process.

We will also be removing the column 'YearsInCurrentRole' and 'Job Level' as it is also multicollinear and these particular features does not have much effect on the prediction process and also very less info about these features.

We will be removing the column 'Years with Current Manager' as we know very less info about this feature.



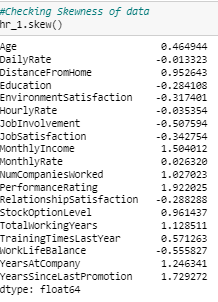
By removing these columns, we have reduced multicollinearity in the dataset successfully.

Step – 3

CHECKING FOR SKEWNESS

As we know checking and removing skewness from our dataset is important as it affects the accuracy of predictive models. Skewness scales from -1 to 1 anything above or below the threshold is either positively or negatively skewed.

Code:



From the above code, we see that there is not much skewness in the dataset and we can move on to the next step.

Step – 4

LABEL ENCODING

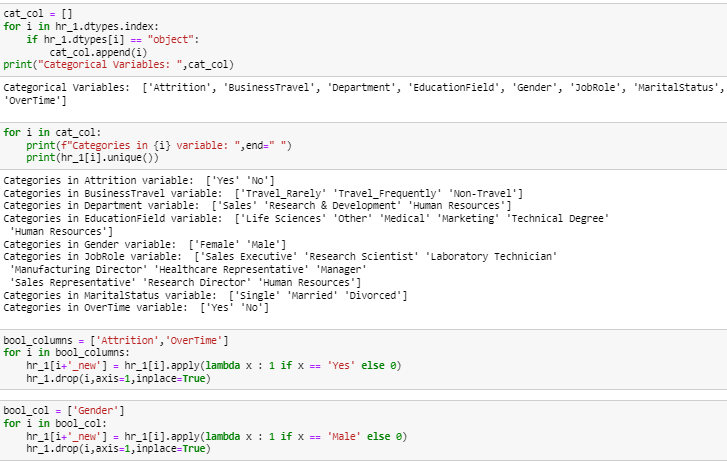
Now, we label encode the categorical variables (i.e from string to int) so that it will be easier for the prediction purpose.

For Two-class variable categorical columns, we convert them into binary variable (i.e either 0 or 1)

For Multi-class variable categorical columns, we create dummy variables or one-hot encoding.

i) For Two-class variable columns:

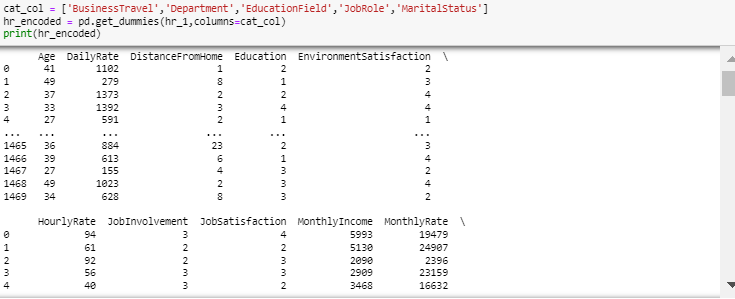
Code:



As, we can see from the above code we have successfully encoded the Two-class categorical columns.

ii) For multi-class variable columns

Code:

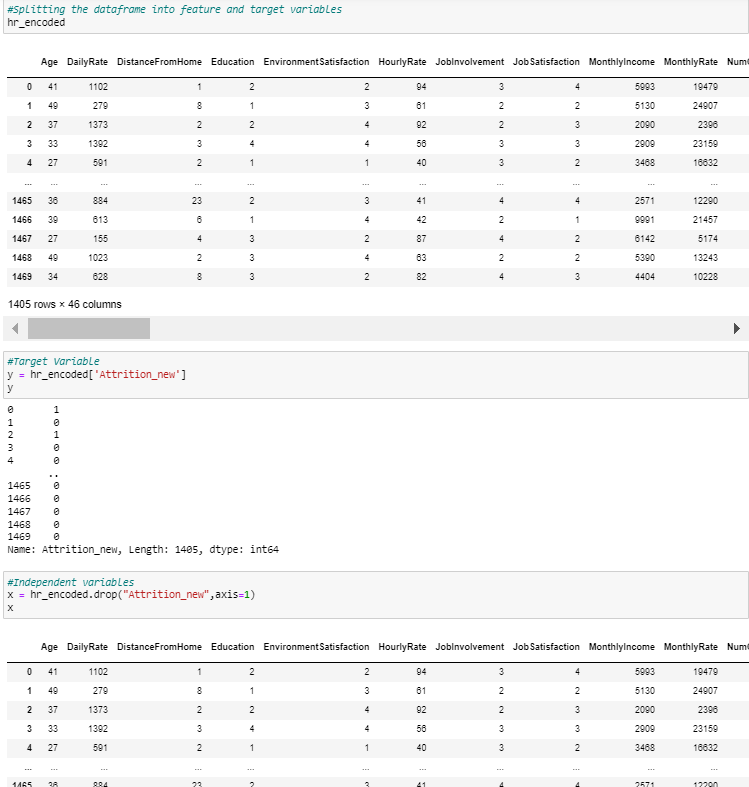


Now, from the code we see that we have created a list of multi-class variable columns and used the dummy variable method to encode the columns. We have encoded the dataset successfully.

Step – 5

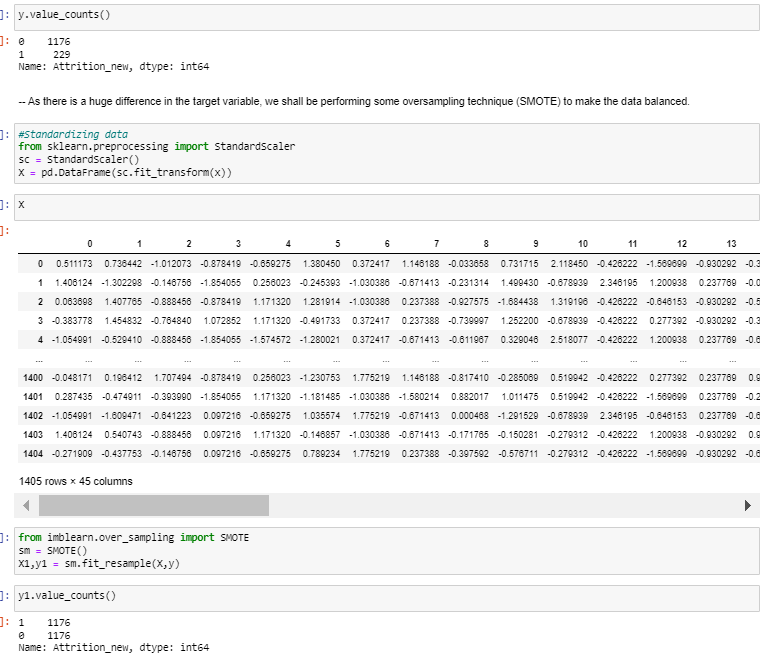
SPLITTING THE DATA INTO TARGET AND FEATURES

Now, we split the encoded data into target variable (y) and independent variables (x).



Step – 6

Standardizing the variables and splitting them into Train and Test Dataset

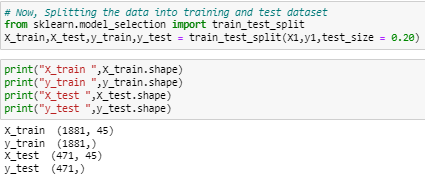


From the above code we see that the data has been split into independent and target variables, as this is a classification problem we have to check for count of target variables.

For Bi - variate classifiaction, we have to check if both the classes are equally balanced or not. If they are then we move towards splitting the data into test and train dataset. If they are imbalanced or oversampled (in our case) then we have to balance these classes, as it will hinder our prediction process by becoming overfit or underfit.

In order to balance the target variable, we use the SMOTE method which is used to balance datasets by generating synthetic samples for the minority class (in our case – 0).

For the independent variables, we Standardize the variables using StandardScaler which scales the data to have a mean of 0 and a standard deviation of 1. We Standardize the data to achieve better accuracy and reliability to our model.



Now, we split the scaled data using the above code.

We split the dataset into 80% train data and 20% test data (test\_size = 0.20).

BUILDING MACHINE LEARNING MODELS

Now that our dataset is cleaned, encoded and scaled, we move on to the next step in our prediction process i.e building machine learning models to predict the attrition rate in the company so that the HR of that particular company take intiatives to retain their employees as much as possible.

So, as we know our target variable(attrition) is a bi-variate classfication type with 1 being that employee would like to leave the company (Yes) and 0 being that employee would like to stay in the company (No).

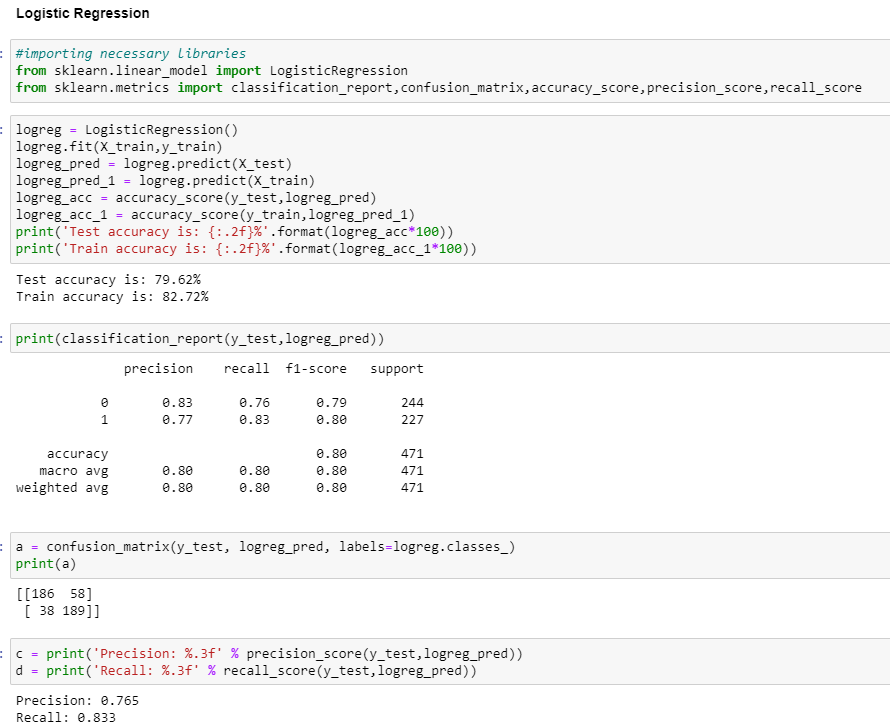
As our target variable is of classification type, we shall build several classification models to know how the accuracy with different model changes and select the model with the highest accuracy and reasonable performance metrics (such as precision score, recall score, classification report, confusion matrix) and based on the accuracy whether the model is performing well with unknown data (cross- validation) and so on.

After Finalising our model, we can also make adjustments with our model parameters and increase the accuracy of our model for better prediction rates (Hyper - Parameter Tuning).

**1) MODEL TRAINING**

Code:

i) First Model - Logistic Regression



Logistic Regression:

Test accuracy = 79.62% , Train accuracy = 82.72%

Accuracy is pretty good for both the train and test sets. It indicates that the model correctly classified approximately 79.6% of the instances in the test set and approximately 82.7% of the instances in the train set.

As we can see, the test accuracy is little lower than the train accuracy but there is not a huge difference between these two. Therefore, there are no signs of overfitting of the model.

From the confusion matrix, the desired result is to have a high number of true positives(TP) [top left] and true negatives(TN) [bottom right] while minimizing the number of false positives(FP)[bottom left] and false negatives(FN)[top right]. So, from the confusion matrix above we see that the number of TP-(186) and TN-(189) is much greater than FP-(38) and FN-(58) which is a really good sign for our prediction purpose.

Precision – measures the proportion of true positive predictions out of all positive predictions made by the model . In this case, a precision score of 0.765 indicates that the classification model achieved a precision rate of 76.5% on the test dataset.

Recall – measures the proportion of true positive predictions out of all true positive instances in the data set . In this case, a recall score of 0.833 indicates that the classification model achieved an 83.3% recall rate on the test dataset.

ii) Second Model – Decision Tree Classifier



Decision Tree Classifier:

Test accuracy = 84.93% , Train accuracy = 100%

Accuracy is pretty good for both the train and test sets. It indicates that the model correctly classified approximately 84.93% of the instances in the test set and approximately 100% of the instances in the train set.

As we can see, the test accuracy is way lower than the train accuracy and there is a huge difference between these two. Therefore, there are signs of overfitting in this model.

From the confusion matrix above we see that the number of TP-(209) and TN-(191) is much greater than FP-(36) and FN-(35) which is a really good sign for our prediction purpose.

A precision score of 0.845 indicates that the classification model achieved a precision rate of 84.5% on the test dataset. A recall score of 0.841 indicates that the classification model achieved an 84.1% recall rate on the test dataset.

iii) Third Model - Random Forest Classifier



Random Forest Classifier:

Test accuracy = 95.12% , Train accuracy = 100%

Accuracy is pretty good for both the train and test sets. It indicates that the model correctly classified approximately 95.12% of the instances in the test set and approximately 100% of the instances in the train set.

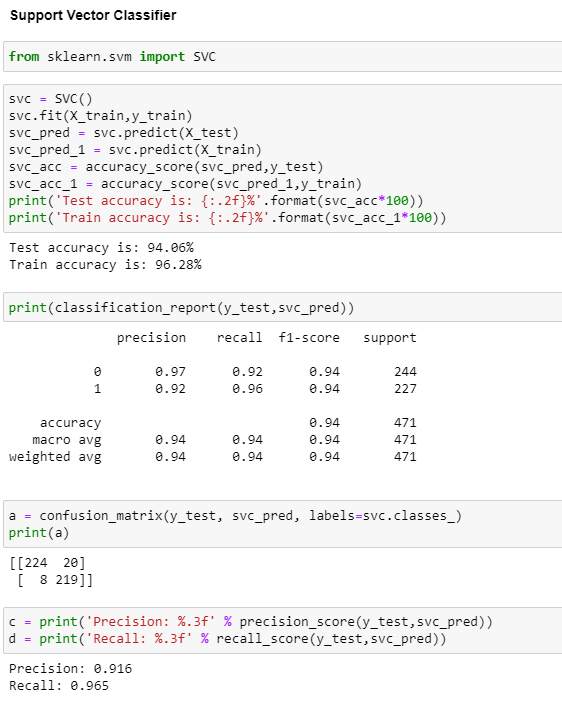
As we can see, the test accuracy is little lower than the train accuracy but there is no huge difference between these two. Therefore, there are no signs of overfitting in this model.

From the confusion matrix above we see that the number of TP-(239) and TN-(209) is much greater than FP-(18) and FN-(5) better than the first two models, which is a really good sign for our prediction purpose.

A precision score of 0.977 indicates that the classification model achieved a precision rate of 97.7% on the test dataset. A recall score of 0.921 indicates that the classification model achieved an 92.1% recall rate on the test dataset.

iv) Fourth Model – Support Vector Classifier

Code:



Support Vector Classifier :

Test accuracy = 94.03% , Train accuracy = 96.28%

Accuracy is pretty good for both the train and test sets. It indicates that the model correctly classified approximately 94.03% of the instances in the test set and approximately 96.28% of the instances in the train set.

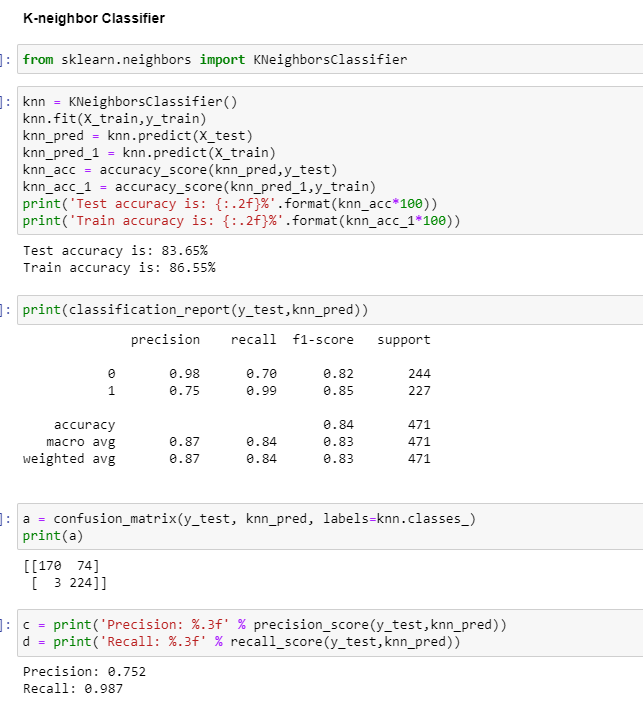
As we can see, the test accuracy is not much lower than the train accuracy(almost equal) . Therefore, there is absolutely no signs of overfitting in this model.

From the confusion matrix above we see that the number of TP-(224) and TN-(219) is much greater than FP-(8) and FN-(20) which is a really good sign for our prediction purpose. (Lowest FP count out of all the models).

A precision score of 0.916 indicates that the classification model achieved a precision rate of 91.6% on the test dataset. A recall score of 0.965 indicates that the classification model achieved an 96.5% recall rate on the test dataset.

v) Fifth Model – K – neighbor Classifier

Code:



K – Neighbor Classifier:

Test accuracy = 83.65% , Train accuracy = 86.55%

Accuracy is pretty good for both the train and test sets. It indicates that the model correctly classified approximately 83.65% of the instances in the test set and approximately 86.55% of the instances in the train set.

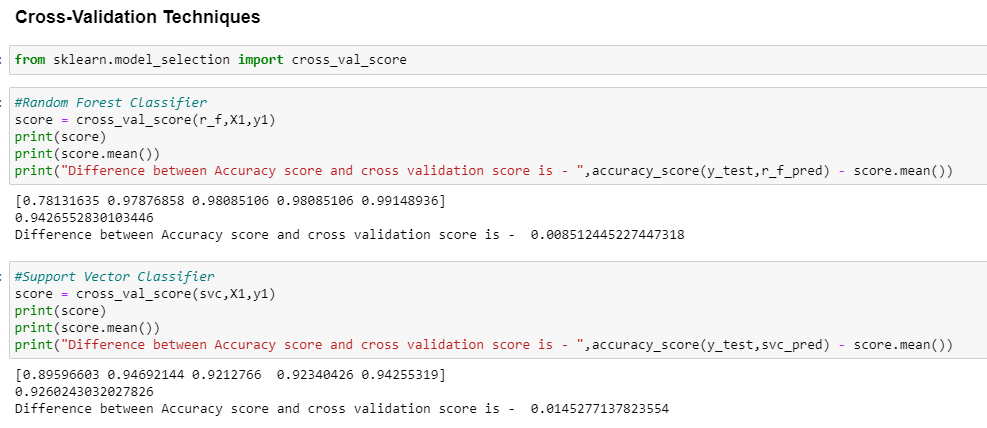
As we can see, the test accuracy is not much lower than the train accuracy. Therefore, there is absolutely no signs of overfitting in this model.

From the confusion matrix above we see that the number of TP-(170) and TN-(224) is much greater than FP-(3) and FN-(74) which is a really good sign for our prediction purpose.

A precision score of 0.752 indicates that the classification model achieved a precision rate of 75.2% on the test dataset. A recall score of 0.987 indicates that the classification model achieved an 98.7% recall rate on the test dataset.

**2) MODEL SELECTION**

Code:



Cross validation is a technique used to evaluate the performance of a model on unseen data. As seen from the above code, I am going to use cross- validation for our model selection between the top two models (Random Forest Classifier and Support Vector Classifier) as these two models have the highest accuracy, precision and recall score out of all the models and also the number of TP and TN values is also high, also the number of FP and FN is also the lowest.

CONCLUSION:

From the model training and the cross val scores (of the top two models) done in the previous section, I am going to choose Support Vector Classifier (SVC) as my final model for the prediction of Attrition rate of a company.

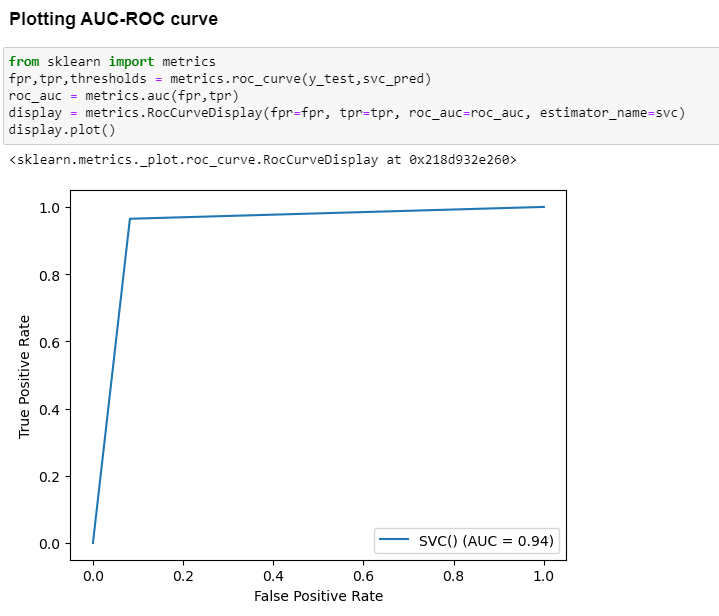
The reason I choose SVC as my final model, is because of it's really good performance with the classification of values. The accuracy score of the model is the high compared to all of the classification models tested which is (94.02%) and from the classification report, we see that the precision, recall and f1 scores are well balanced and have a really good scores individually too. The metric recall or TPR(measure of how accurately our model is able to identify the relevent data) is important for our data, as the HR needs an accurate data for the attrition rate of the company and needs to take relevent steps to retain their employees. From the confusion matrix, we see that the ratio of TP and TN is really good as compared to other models, and the number of FP is more as compared to Random Forest Classifier (which has the same accuracy score as SVC).

Finally, from the cross val scores of the top 2 models (i.e Random Forest Classifier and SVC) SVC has equal accuracy scores across the five folds whereas RFC does not have equal scores.

Therefore, SVC is my best and final Model.

Plotting AUC-ROC curve for Final Model

Code:



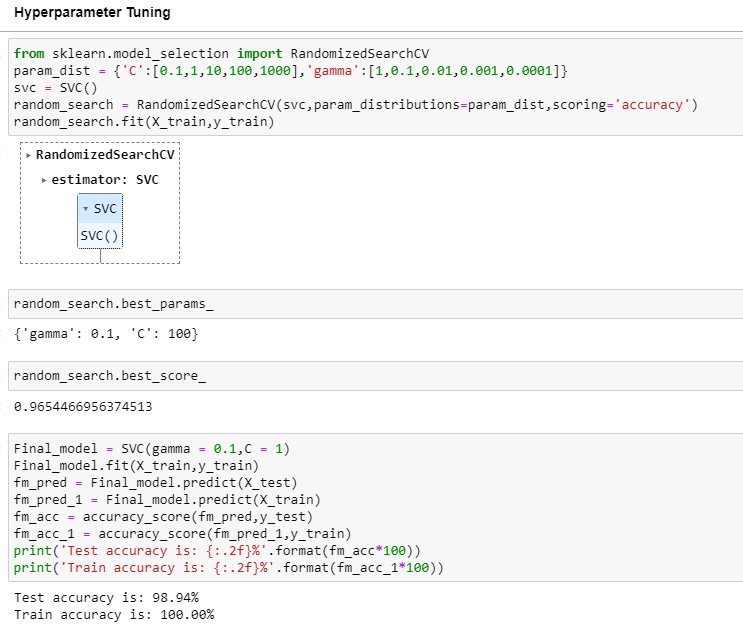
The AUC (Area under curve) and ROC (Reciever operating Characteristics) curve are common evaluation measures used in binary classification tasks. They provide insight into the performance and discrminating power of a classification model.

The Higher AUC score indicates a better performing model with a great ability to discriminate between classes. The ROC curve, when closer to the upper left corner, demonstrates a higher true positive rate (TPR) versus false positive rate (FPR). This indicates that the model has a better balance between correctly identifying positive instances and minimizing false positive errors.

In this case, An AUC score of 0.94 implies that, the model correctly ranks a randomly selected positive instance higher than a randomly selected negative instance about 94% of the time.

**3) HYPERPARAMETER TUNING**

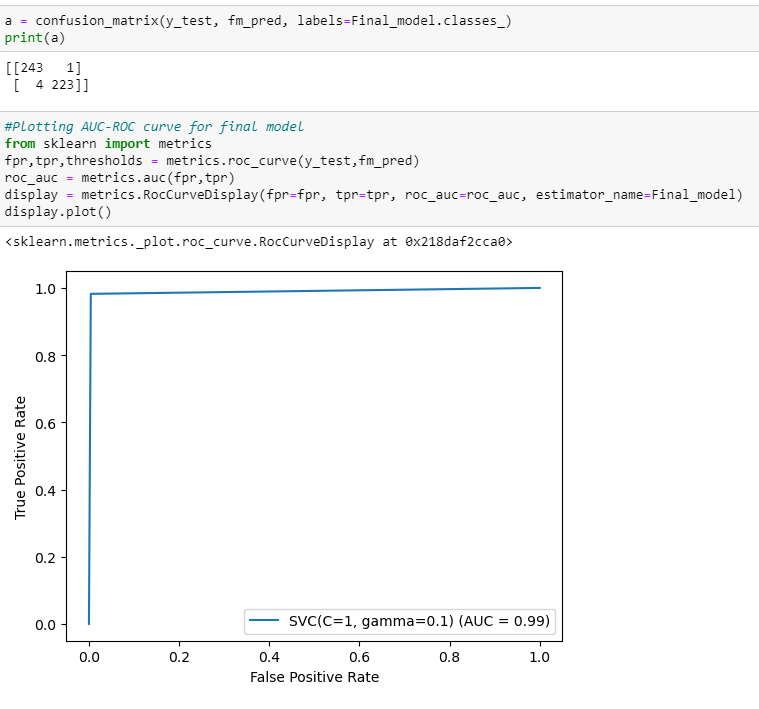
Code:



Hyper Parameter Tuning is basically referred to as tweaking the parameters of the model. As you can see from the above set of code we have performed hyper parameter Tuning for our final model i.e Support Vector Classifier (SVC) to enhance the accuracy of the model.

We see that the train accuracy has been increased 4.93% after performing Hyper-Parameter Tuning. We used Randomized Search Hyperparameter optimization techniques to increase our model accuracy.

PLOTTING AUC-ROC CURVE FOR FINAL TUNED MODEL



As you can see the AUC score has been increased from 0.94 to 0.99 implying that, the model correctly ranks a randomly selected positive instance higher than a randomly selected negative instance about 99% of the time. Suggesting the model is really good for our prediction process.

**CONCLUDING REMARKS**

So, these are the various steps needed to be taken for the prediction of Attrition rate in a company. By utilising this above model, an HR can take significant decisions and measures for reducing the attrition rate in a company.

**THE END**