**HR Analytics Project – Understanding the Attrition in HR.**

**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?

**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.

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.

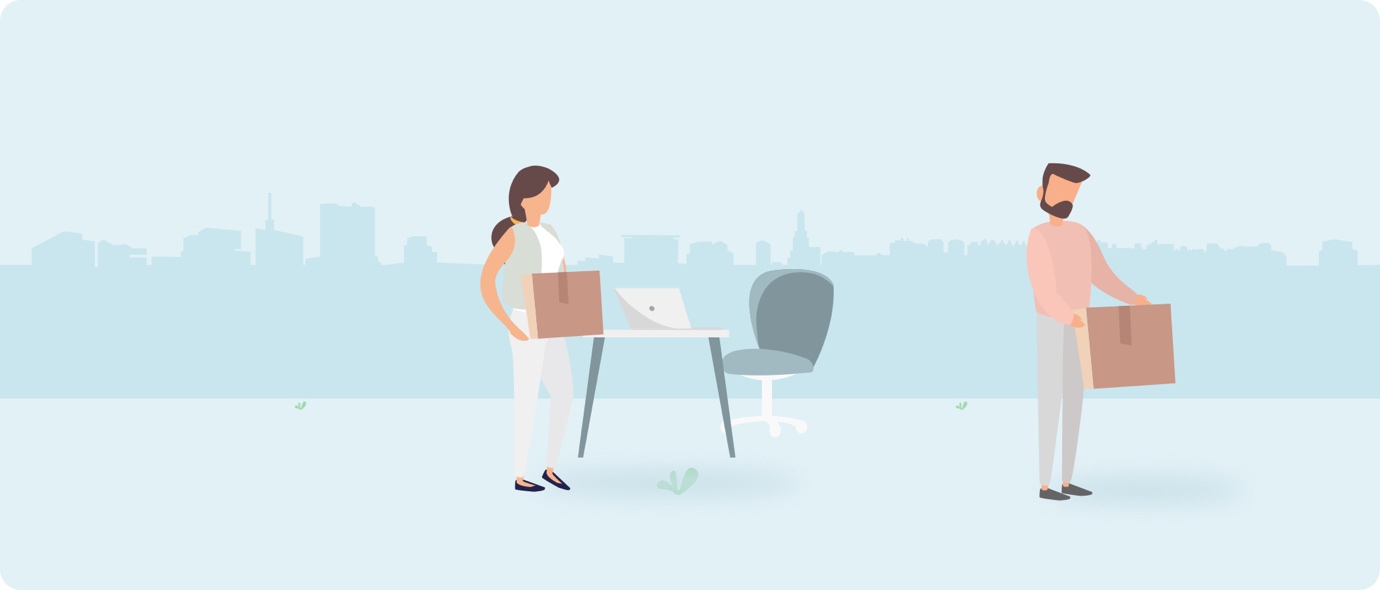
**---------WHAT WE UNDERSTAND FROM PROBLEM STATEMENT--------**

From above we can understand that company invest time and money in training of the employees to increase the effectiveness of their employees and to increase the quality of the employee and the company as well.

HR Analytics is a field in which refers to applying analytic processes to the human resource department of an organization in the hope of improving the employee performance and therefore getting a better return on investment.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 refers to loss of employees overtime. If attrition is at higher rate it will be problematic for the company as they will again have to start from scratch i.e. from Job Posting, Hire trainer, Paperwork, Hiring new employees, Training them.

So from above we can understand that it is important to reduce the Attrition Rate in the Company to reduce the Expenses and to get better returns.



**Now we will be doing some steps to find out the Employees who are going to attrite.**

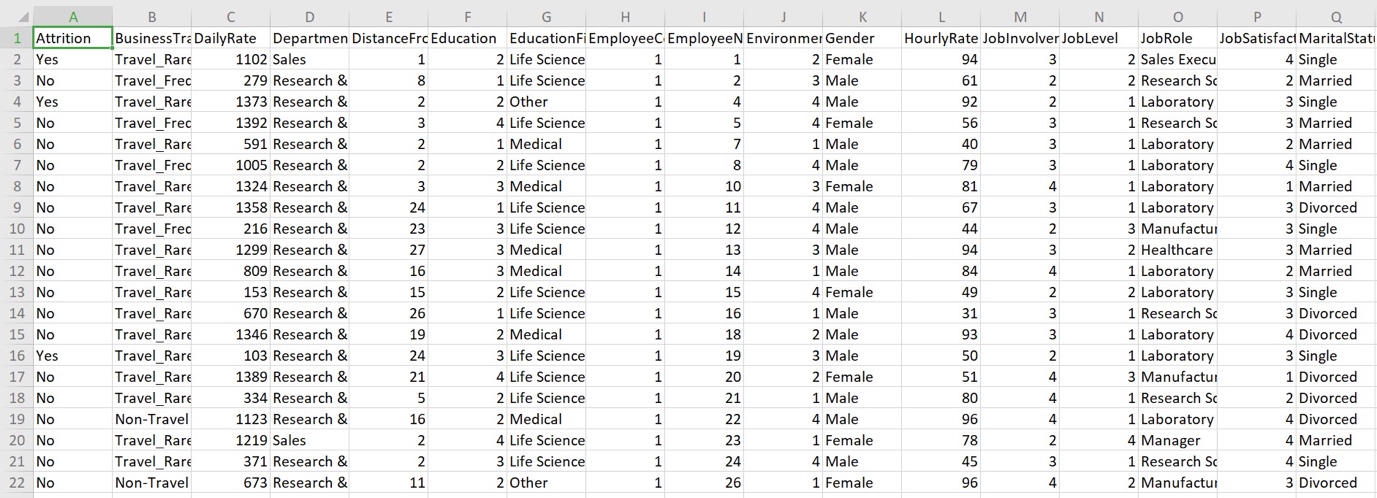
**Methodology :-**

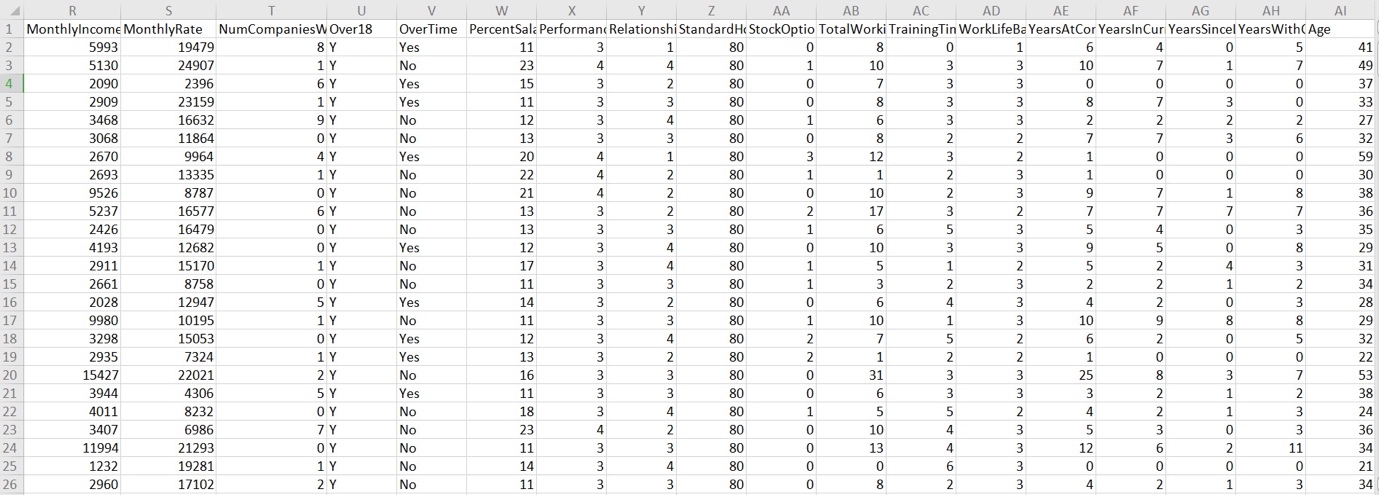
1. **DATA COLLECTION**
2. **DATA PRE-PROCESSING**
3. **DIVIDING THE DATA into TWO PARTS ‘TRANING’ AND ‘TESTING’.**
4. **BUILD UP THE MODEL USING TRAINING DATASET**
5. **DO THE ACCURACY TEST USING TESTING ACCURACY**
6. **FINDING THE CROSS-VALIDATION SCORE**
7. **COMPARING ACCURACY SCORE AND CROSS VALIDATION SCORE AND FIND OUT BEST ALGORITHM.**
8. **SAVING THE BEST MODEL FOR FURTHER USE.**

**DATA EXPLORATION: -**

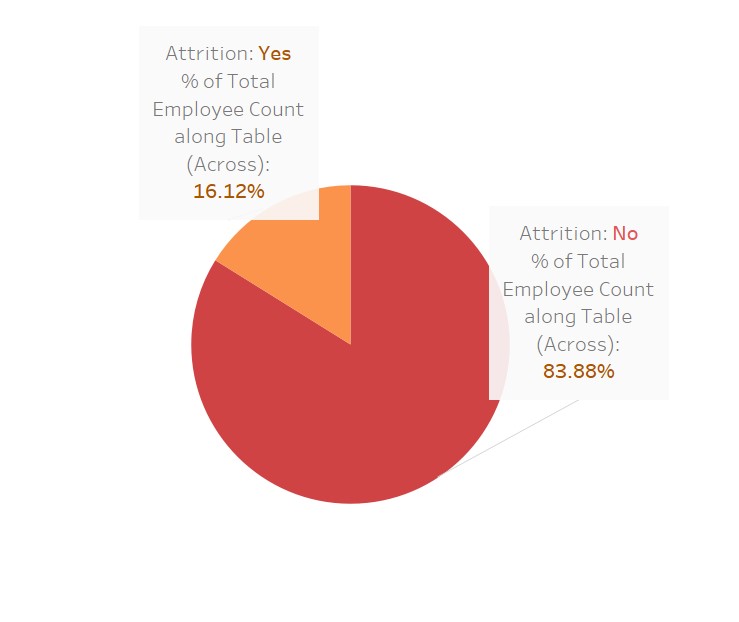
This dataset is collected from the IBM Human Resource department. The dataset contains 1470 observation and 35 Columns/Variables ‘Attrition’ is our Target Variable i.e. Either Yes or No. So here we will be using Classification Algorithms to find out the Attrition of the employee.

**Lets have a quick look at the Dataset:-**

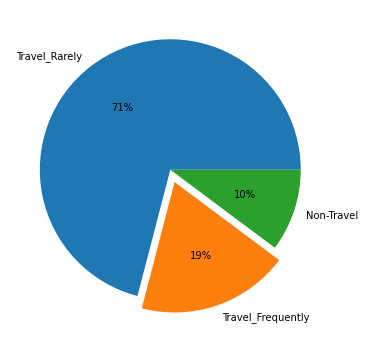




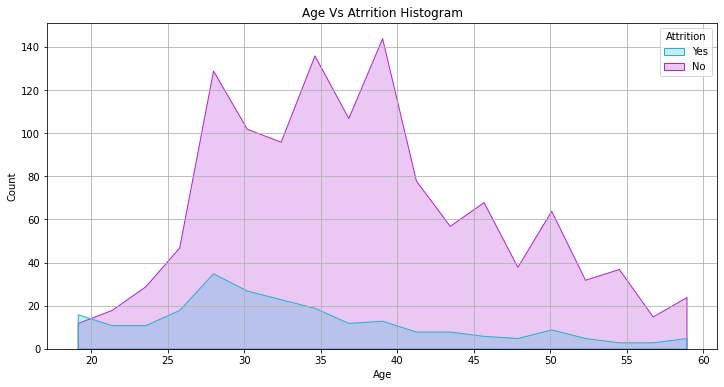
**Lets Look at the Attrition rate Diagram using pie chart:-**

****

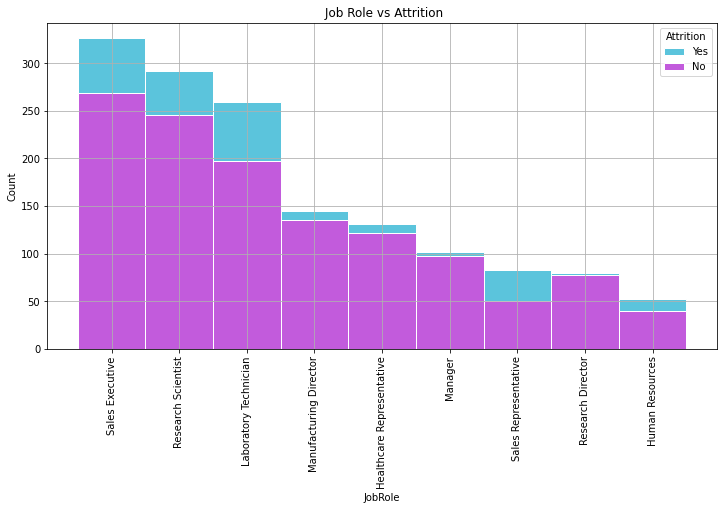
**From above we can see that in IBM HR Dataset 16.12% of the employees are Attrited.**



**From above we can see the pie chart of employee travelling and we can see that only 19% of them travel frequently.**



**From above we can see graphical representation of Age and Attrition rate of the employee who are 25-30 are attiring in more number than in any other age.**

****

**From above we can see the graphical representation of employee leaving with respect to the job role more number of employee Attrite when they are either Sales Executive or Lab Technician.**

**Data Preparation:-**

We have to see if there are any missing values in the dataset we can do that using ds.isnull.sum()

By running the above code we can see that all the column values are 0 so there are no missing values in the dataset

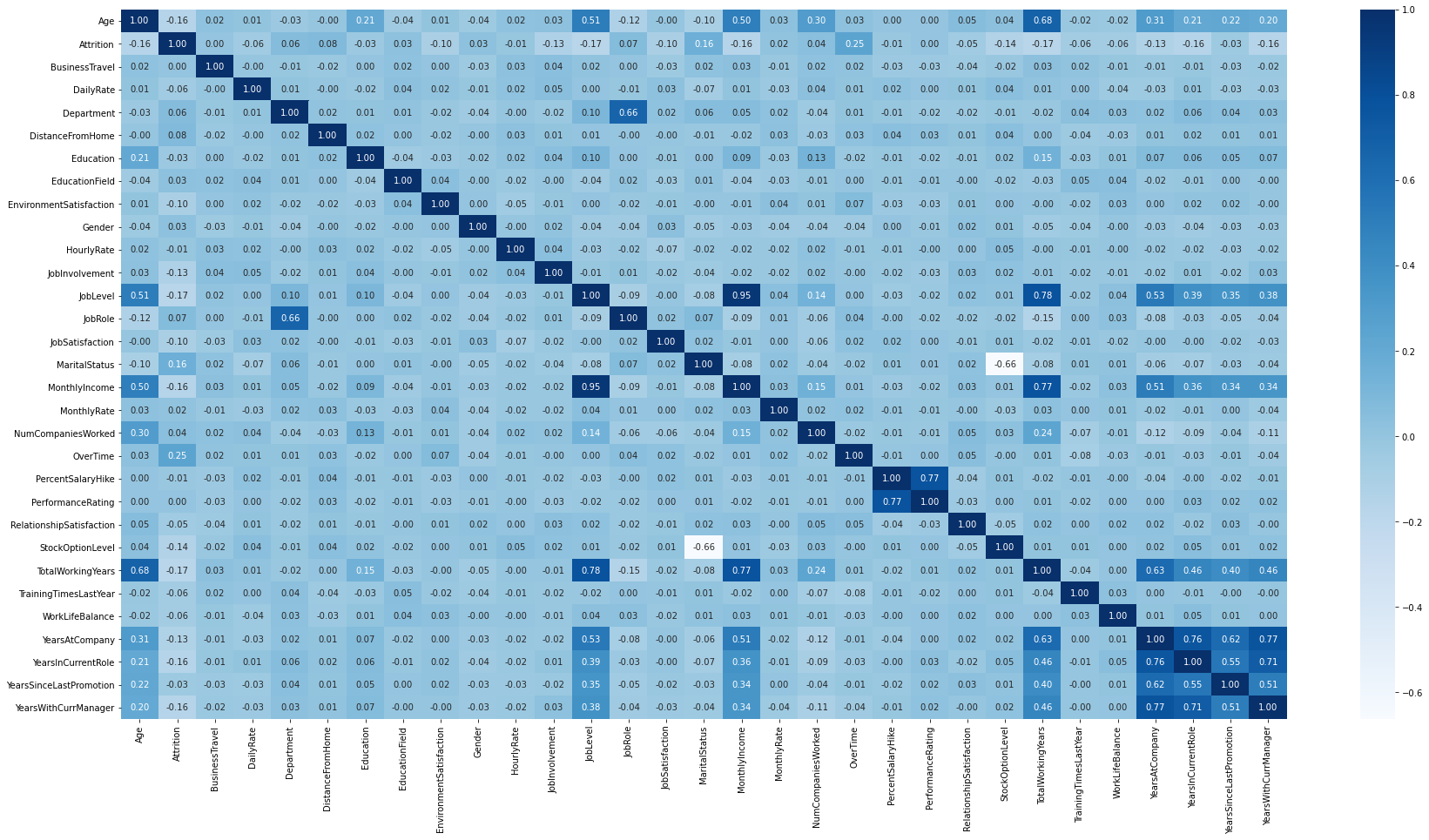
Now lets change the object column to Numeric using label Encoder as we can only pass numeric data to Algorithms as they can only configure those types.

le = LabelEncoder()

ds[ds.select\_dtypes(include=['object']).columns] = ds[ds.select\_dtypes(include=['object']).columns].apply(le.fit\_transform)

from above code all the object type variable will be converted into numeric type without loosing their impact on the dataset.

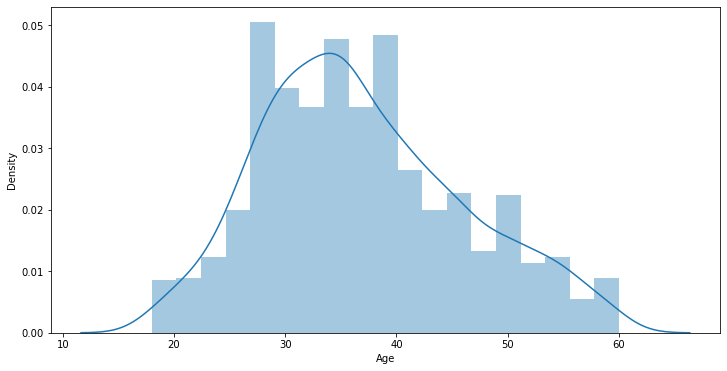
Now lets see the correlation between our data using ds.corr()

\

**From above we can see the heatmap of Correlation of the dataset and we can understand the correlation by looking at the graph using the color combination it is very easy to understand if the color is dark then the correlation is Positive and when the color is light the correlation between the column decreases and if the column is white then the column has negative correlation between them here we have negative correlation Between Marital Status and StockOptionLevel Column as there correlation is -0.66**

Then we have to see the skewness of the data using ds.skew() and if the skewness if greater or less than +/- 0.5 we will fix them using power\_transform method or boxcox or yeo-johnson method boxcox is used for positive skewness and yeo-johnson is used for negative skewness.

After fixing the skewness when all the column are normally distributed we will now split our data into training and testing set and save them in X and Y variables.



From above graph we can see the normal distributed column.

**X = Independent Variable**

**Y = Dependent / Output Variable**

X = ds.drop(‘Attrition’, axis = 1)

Y = ds[‘Attrition’]

Here we have specified to drop the dependent variable from axis = column i.e. axis = 1

As we now have Dependent variable and independent variable lets now do Standard Scaling to our Independent variable to convert all the values in same scale.

X\_new = sc.fit\_transform(X)

X\_new = pd.DataFrame(X\_new, columns = X.columns)

X\_new.head()

By using above code we can Scale all out independent column values.

Lets now split our data using train\_test\_split

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.22, random\_state = 42)

From above we have successfully separated our training and testing dataset with testing size as 22% and training size as 78% we can also do as 20 – 80 or 30 – 70 and here random state we are taking as 42 later we will find out the best random state. Using python loop.

lr = LogisticRegression()

lr.fit(X\_train, Y\_train)

lr.score(X\_train, Y\_train)

predlr = lr.predict(X\_test)

print("Accuracy Score:", accuracy\_score(Y\_test, predlr))

print("Classification Report", classification\_report(Y\_test, predlr))

print("Confusion Matrix", confusion\_matrix(Y\_test, predlr))

by writing the above code we can see the below output and we can see lots of maths by looking at it

**Accuracy Score: 0.9012345679012346**

**Classification Report precision recall f1-score support**

**0 0.91 0.98 0.95 283**

**1 0.74 0.34 0.47 41**

**accuracy 0.90 324**

**macro avg 0.82 0.66 0.71 324**

**weighted avg 0.89 0.90 0.88 324**

**Confusion Matrix [[278 5]**

**[ 27 14]]**

From above we can see that Accuracy score is 90.12% for Logistic regression and by looking at this we can say that model has performed very well but why we used other metrics? Because we want to see how the model works with different kind of data from classification report we can see f1 score from there we can see 0 is predicted 95% of time and 1 is predicted 47% of time that means f1 score of the model is 47% as Attrition is Important for us we have to look at F1 score to as it is important to us so lets see which model performs well in all the metrics. Confusion Matrix is solved as 278+14 / 278+5+27+14 it will give 90.12% same as the Accuracy score so why this by using this we can find our correct predictions and the false predictions.

Lets now find out the best random state and the best accuracy score of the dataset using python loop and the classification models

Step 1: Create a List of model

models = [LogisticRegression(), DecisionTreeClassifier(), RandomForestClassifier(), AdaBoostClassifier(), SVC(), KNeighborsClassifier(), GaussianNB()]

Step 2: Write the Logic:

maxacc = 0

maxrs = 0

n = 0

for i in range(1,1000):

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.22, random\_state = i)

for m in models:

m.fit(X\_train, Y\_train)

pred = m.predict(X\_test)

acc = accuracy\_score(Y\_test, pred)

if acc > maxacc:

maxacc = acc

maxrs = i

n = m

print("Maximum Accuracy is {} at random\_state {} for model {}". format(maxacc, maxrs, n))

from above we can see that maxacc gives the maximum accuracy for the model maxrs gives the maximum Accuracy random state and n gives the name of the model which has the highest Accuracy at random state = i

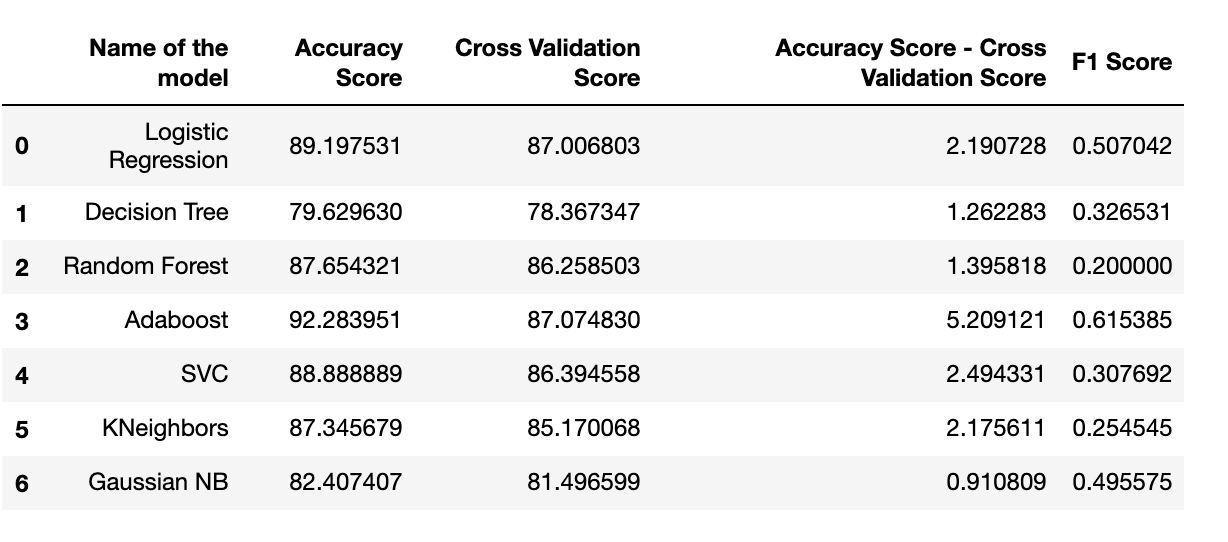
By running above code I got:

**Maximum Accuracy is 0.9228395061728395 at random\_state 281 for model AdaBoostClassifier()**

I got 92.28% accuracy score at random state 281 for AdaBoostClassifier()

Now we fill fit all the model with random state 281 for getting the better result

After finding all the model accuracy score we will see the Cross\_validation\_Score to find out if the model is Overfitting the data or Underfitting the data after that we will compare the accuracy score after cross validation and before cross validation and see which model has performed better.



**from above we can see that Decision tree, Random forest did the best but the f1 score is highest for Adaboost, Logistic Regression, Gaussian NB So we will do all the algorithms and find out the best**

**Now we will perform GridSearchCV for all the Models to improve the accuracy score**

**We got the metrics as**

**Logistic Regression**

**Logistic regression**

**Accuracy Score 0.8919753086419753**

**Classification report precision recall f1-score support**

**0 0.91 0.97 0.94 280**

**1 0.67 0.41 0.51 44**

**accuracy 0.89 324**

**macro avg 0.79 0.69 0.72 324**

**weighted avg 0.88 0.89 0.88 324**

**Confusion Matrix [[271 9]**

**[ 26 18]]**

**F1 Score 0.5070422535211268**

**Decision Tree**

**Decision Tree Classifier**

**Accuracy Score 0.8734567901234568**

**Classification report precision recall f1-score support**

**0 0.89 0.97 0.93 280**

**1 0.59 0.23 0.33 44**

**accuracy 0.87 324**

**macro avg 0.74 0.60 0.63 324**

**weighted avg 0.85 0.87 0.85 324**

**Confusion Matrix [[273 7]**

**[ 34 10]]**

**F1 Score 0.32786885245901637**

**Random Forest**

**Random Forest Classifier**

**Accuracy Score 0.8734567901234568**

**Classification report precision recall f1-score support**

**0 0.89 0.97 0.93 280**

**1 0.59 0.23 0.33 44**

**accuracy 0.87 324**

**macro avg 0.74 0.60 0.63 324**

**weighted avg 0.85 0.87 0.85 324**

**Confusion Matrix [[273 7]**

**[ 34 10]]**

**F1 Score 0.3214285714285714**

**AdaBoost**

**Adaboost Classifier**

**Accuracy Score 0.9012345679012346**

**Classification report precision recall f1-score support**

**0 0.90 1.00 0.95 280**

**1 0.93 0.30 0.45 44**

**accuracy 0.90 324**

**macro avg 0.91 0.65 0.70 324**

**weighted avg 0.90 0.90 0.88 324**

**Confusion Matrix [[279 1]**

**[ 31 13]]**

**F1 Score 0.4482758620689656**

**SVC**

**SVC**

**Accuracy Score 0.8919753086419753**

**Classification report precision recall f1-score support**

**0 0.91 0.97 0.94 280**

**1 0.70 0.36 0.48 44**

**accuracy 0.89 324**

**macro avg 0.80 0.67 0.71 324**

**weighted avg 0.88 0.89 0.88 324**

**COnfusion Matrix [[273 7]**

**[ 28 16]]**

**F1 Score 0.4776119402985074**

**KNeighbour Classifier**

**KNeighbour Classifier**

**Accuracy Score 0.8580246913580247**

**Classification report precision recall f1-score support**

**0 0.87 0.98 0.92 280**

**1 0.38 0.07 0.12 44**

**accuracy 0.86 324**

**macro avg 0.62 0.53 0.52 324**

**weighted avg 0.80 0.86 0.81 324**

**Confusion Matrix [[275 5]**

**[ 41 3]]**

**F1 Score 0.11538461538461538**

**Gaussian NB**

**Gaussian NB**

**Accuracy Score 0.8240740740740741**

**Classification report precision recall f1-score support**

**0 0.94 0.85 0.89 280**

**1 0.41 0.64 0.50 44**

**accuracy 0.82 324**

**macro avg 0.67 0.74 0.69 324**

**weighted avg 0.87 0.82 0.84 324**

**COnfusion Matrix [[239 41]**

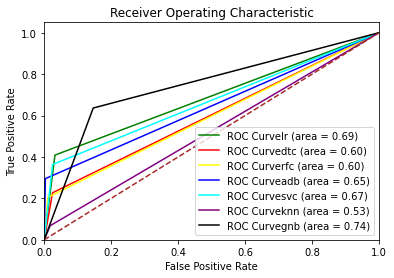
**[ 16 28]]**

**F1 Score 0.49557522123893816**

**From above we can see Logistic Regression and Gaussian NB has the highest F1 Score that means from these two model we have to find out our best model.**

**Now we will be using One More Metric AUC ROC CURVE t0 find out the area covered for each of the model and we will choose the model which has highest area under the curve as our best model.**

**Lets now see AUC ROC CURVE for all the model and select our best model for this data.**

****

**From above diagram we can see that Gaussian NB has the heighest area under the curve which is at 74% and after that there is Logistic Regression which is at 69% as we figured out that we have to choose our best model between Logistic and Gaussian NB because of the f1 score so we will choose Gaussian NB as our best model and save it as an Object file using joblib**

**joblib.dump(GSCVgnb.best\_estimator\_, 'IBMAttrition.pkl')**

**from above code we can save our best model and we can use .pkl file for further usage.**

**Conclusion:-**

As Employees are the Base of the Company Employees has to be well Trained and Well Skilled if the Posses both Employee and the Company will be benefitted.

Companies train there Employees so that the employee can develop new skill and perform well for the company company invest time and money on the employee to Upscale the company as well as the employee so they both can increase there revenue in the market and keep growing the Business.

So if the employee leaves the company or Attrition happens then the company has to again repeat the same process for every new employee that joins the company and Invest time upon them so it is necessary for the company to know that the employee will leave or stay with the company and if Attrition is High then the Company will loose many more time to train new people so it is important to Study the behaviour of the employees and as we know we can tell things by looking at the persons behaviour and body language.

It is important to know the person we will leave the company in near future as the company get some time to hire new employee and train them so that they suffer from low loss as compared to the loss they will face after the experienced employee leaves the company.

So Data Scientists are hired for predicting the Attrition of the Company so that company can know the person who can Attrite.