**Name Of the Project**

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

**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.

In this project we are going to discuss about attrition of an employees in company. We have to predict that; employee will leave current company or not. In this project Attrition is the dependent variable which is depend on the various independent features. Attrition of employees means that, the employee who leaves the company for any reason.

Attrition of employee from the organization has many reasons, like resignation, termination, death of an employee, retirement, job mismatch or issues in the workplace. Attrition is a w waste of time and money. For attrition process an organization has paid lots of money for advertisement and other things. After employee attrition, it takes too much time to hire new employee.

**HR –**

HR (Human Resource) is very important team in any organization. There is lots of responsibilities on HR team. From hiring of employees to the retention of an employee all are have to maintained by HR team like employee training, promotion opportunities for an employee, bonus and workplace environment.

HR plays very important role in employee attrition. When an employee resigns from his job, it’s the responsibility of an HR to find out the reason why the employee resigns. It is the duty of HR to discuss with an employee if he/she has any issues or for what reason he/she leave the company. HR should be tried to give a solution for his/her problem.

HR should motivate the employees for their working and conduct the motivational activities at the workplace and try to conduct extracurricular activities, cultural programs for their overall development and some change from their routine. HR should launch incentive schemes for an employee and try to held some quiz or other competitions and make sure to keep prizes for competitions so that employee should feel attached to the organization and try to work hard for an organization.

**DATA ANALYSIS –**

We have a data for HR Employee Attrition with 1470 rows and 35 columns. In given dataset 34 columns are independent columns and 1 column is dependent column. Attrition column is a dependent column which is depends on other 34 columns. Following are the given columns with their description;

***Age - This column represents the age of an employee***

***Attrition - This column shows that either employee attrited or not.***

***Business Travel - This column represents the traveling history of an employee.***

***Daily Rate - This column represents the travel rate.***

***Department - This column shows from which department the employee is.***

***Distance from home - It shows the distance from home to office.***

***Education - It represents the Education of an employee.***

***Education Field - It shows that from which field the employee complete his/her education.***

***Employee Count - Number of employees.***

***Employee Number - Specific number which provides to each employee.***

***Enviornment Satisfaction - Enviornment in office is satisfactory or not.***

***Gender - It shows the employee is male or female.***

***Hourly Rate - Hourly rate of working.***

***Job Involvement - It represents how much an employee is dedicated to their work.***

***Job Level - This column shows the job level of an employee.***

***Job Role - It represents the Designation of an employee.***

***Job Satisfaction - It shows that, employee is satisfied or not with his job.***

***Marital Status - It represents is employee married, single or divorcee.***

***Monthly Income - It shows the monthly income of an employee.***

***Num Companies Worked - This column represents for many companies the employee worked.***

***Overtime - Is employee does overtime or not.***

***Percent Salary Hike - How much hike the employee get in salary.***

***Performance Rating - It shows the rating of an employee based on his/her performance.***

***Total Working Years - It shows from how many years the employee worked.***

***Work Life Balance - This column represents the balance between work and life.***

***Years At Company - This shows years of working of an employee in company.***

***Years With Curr Manager - This shows the years of working with current manager.***

After describing the column we have to check for the null values in dataset.

|  |
| --- |
| hr.isnull().sum() |

Age 0

Attrition 0

BusinessTravel 0

DailyRate 0

Department 0

DistanceFromHome 0

Education 0

EducationField 0

EmployeeCount 0

EmployeeNumber 0

EnvironmentSatisfaction 0

Gender 0

HourlyRate 0

JobInvolvement 0

JobLevel 0

JobRole 0

JobSatisfaction 0

MaritalStatus 0

MonthlyIncome 0

MonthlyRate 0

NumCompaniesWorked 0

Over18 0

OverTime 0

PercentSalaryHike 0

PerformanceRating 0

RelationshipSatisfaction 0

StandardHours 0

StockOptionLevel 0

TotalWorkingYears 0

TrainingTimesLastYear 0

WorkLifeBalance 0

YearsAtCompany 0

YearsInCurrentRole 0

YearsSinceLastPromotion 0

YearsWithCurrManager 0

We can see that there is no null value in given dataset.

After checking the null values some columns are dropped from the dataset because they have not much effect on dataset. Dropped columns are;

Employee Count

Job Level

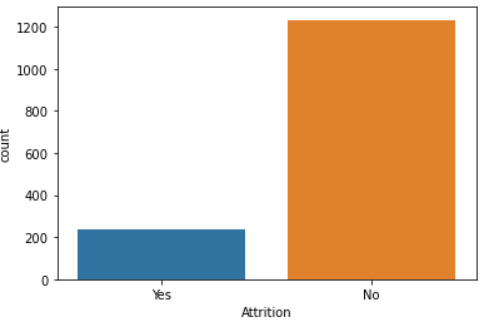
Over 18

Standard Hours

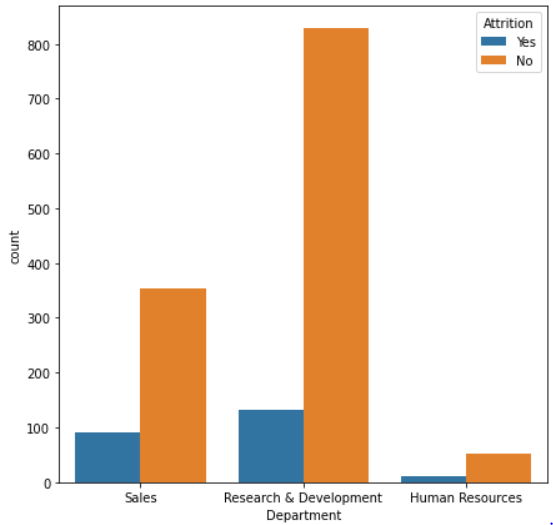
**Exploratory Data Analysis:**

**VISUALIZATION -**

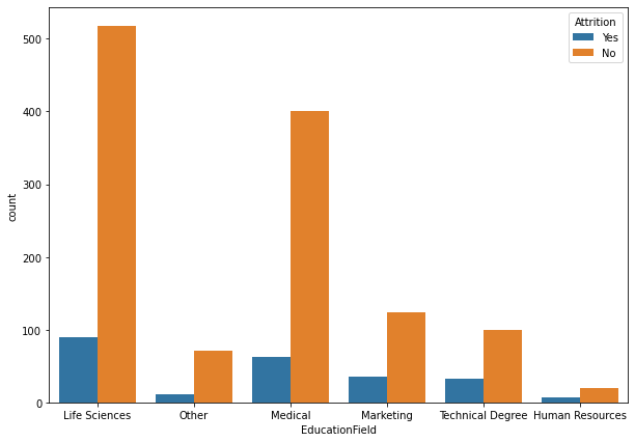
Visualization is the diagrammatical representation of data. Data can be visually representing in map or graphical form. Visualization of data is the easiest thing to understand the data. For given dataset we plot the graph for independent columns with the dependent column (Attrition).

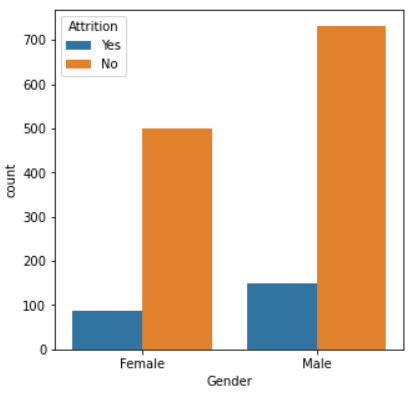


Above graph represents the attrition. From the above graph we can see that number of employees who want Attrition is less than the number of employees who don’t want Attrition.

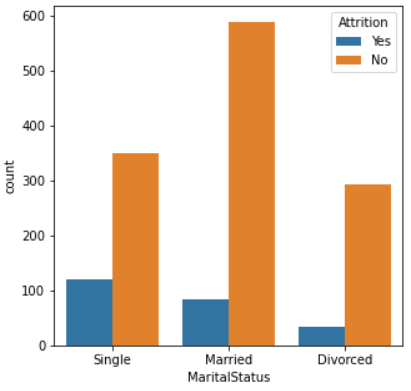


Above graph shows the department wise Attrition of an employees. Employees from Research and Development Department has the highest in number who wants Attrition.

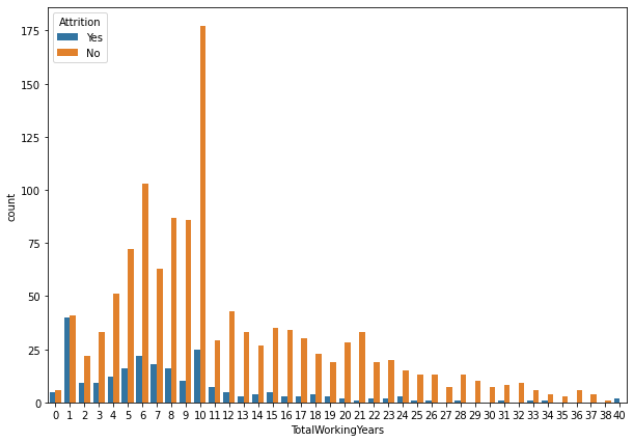
  
Above graph is based on the Education Field. Number of employees is more from Life Science education field who wants Attrition.



Above graph shows the Gender wise Attrition. Number of males is more than the females who wants Attrition.



From above graph we can see that, number of employees who wants Attrition more are single.



Above graph represents the Total working years with Attrition. Less number of experience Employees wants the Attrition.

We represent the data in Graphical format.

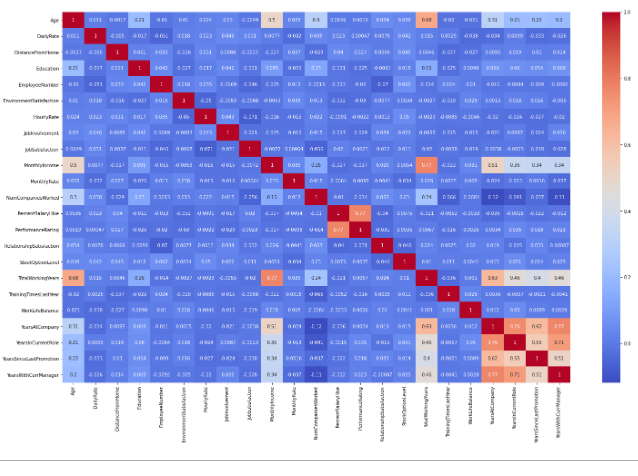
**CORRELATION -**

Correlation is a linear relation between two variables. It shows that how variables are related to each other. Correlation can be Negative or Positive. Positive correlation means two variables are directly proportional to each other and in Negative correlation two variables are indirectly proportional to each other, means if one variable increases other decreases.

We shows the correlation in diagrammatic form using heatmap ;

|  |
| --- |
| plt.figure(figsize=(25,15))  sns.heatmap(hr.corr(),annot=True, cmap='coolwarm') |

First, we have to write the code for plotting the heatmap for correlation.



##### *As we seen that from above plot maximum columns are highly correlated with each other. Age and Monthly Income are highly correlated with Total Working Years. Years at Company, Years in Current Role, Years since last Promotion and Years with Curr Manager are correlated with each other.*

**Pre-Processing Pipeline:**

**SKEWNESS –**

Skewness is an asymmetric curve in set of data. Skewness is the data cleaning process. Skewness can be right or positive skewness, left or negative skewness and zero or normal skewness.

When the distribution shifted towards left side then it’s a positive skewness and when the distribution shifted towards right side then it’s called negative skewness. Normal distribution has bell shape curved. We also find the skewness in given dataset by using following code;

|  |
| --- |
| hr.skew().sort\_values(ascending = False) |

We can see that Skewness present in following columns;

YearsSinceLastPromotion - 1.984290

PerformanceRating - 1.921883

YearsAtCompany - 1.764529

MonthlyIncome - 1.369817

TotalWorkingYears - 1.117172

NumCompaniesWorked - 1.026471

StockOptionLevel - 0.968980

DistanceFromHome - 0.958118

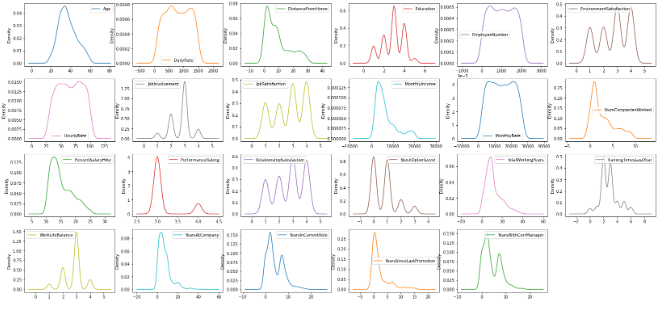
YearsInCurrentRole - 0.917363

YearsWithCurrManager - 0.833451

PercentSalaryHike - 0.821128

Now we plotting the graph for Skewness;

|  |
| --- |
| hr.plot(kind = 'density',subplots = True, layout = (6,6), figsize=(28,20), sharex=False)  plt.show() |



We plot the skewness in different columns.

After finding the Skewness now we convert objective data into numerical form by using Label Encoder by using following code;

|  |
| --- |
| from sklearn.preprocessing import LabelEncoder  lec = LabelEncoder()    for i in hr.columns:  if hr[i].dtypes == object:  hr[i] = lec.fit\_transform(hr[i].values.reshape(-1,1)) |

Now we have to check that is objective columns are converted into numerical form or not.

|  |
| --- |
| hr.info() |

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1470 entries, 0 to 1469

Data columns (total 31 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Age 1470 non-null int64

1 Attrition 1470 non-null int32

2 BusinessTravel 1470 non-null int32

3 DailyRate 1470 non-null int64

4 Department 1470 non-null int32

5 DistanceFromHome 1470 non-null int64

6 Education 1470 non-null int64

7 EducationField 1470 non-null int32

8 EmployeeNumber 1470 non-null int64

9 EnvironmentSatisfaction 1470 non-null int64

10 Gender 1470 non-null int32

11 HourlyRate 1470 non-null int64

12 JobInvolvement 1470 non-null int64

13 JobRole 1470 non-null int32

14 JobSatisfaction 1470 non-null int64

15 MaritalStatus 1470 non-null int32

16 MonthlyIncome 1470 non-null int64

17 MonthlyRate 1470 non-null int64

18 NumCompaniesWorked 1470 non-null int64

19 OverTime 1470 non-null int32

20 PercentSalaryHike 1470 non-null int64

21 PerformanceRating 1470 non-null int64

22 RelationshipSatisfaction 1470 non-null int64

23 StockOptionLevel 1470 non-null int64

24 TotalWorkingYears 1470 non-null int64

25 TrainingTimesLastYear 1470 non-null int64

26 WorkLifeBalance 1470 non-null int64

27 YearsAtCompany 1470 non-null int64

28 YearsInCurrentRole 1470 non-null int64

29 YearsSinceLastPromotion 1470 non-null int64

30 YearsWithCurrManager 1470 non-null int64

dtypes: int32(8), int64(23)

memory usage: 310.2 KB

Now we don’t have any categorical column.

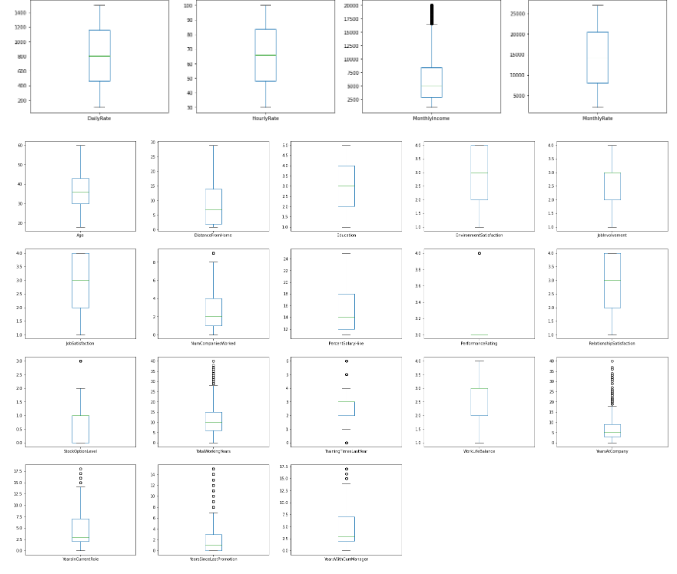
After converting the objective data into numerical format, we have to find out the outliers in dataset.

**OUTLIERS** -

An outlier is something which is very different from the given data. Distance of an outliers from the other points are maximum or out of range. For finding an outliers we divided the columns into two groups; hr\_non\_cat\_columns and hr\_descrete\_columns.

Now we draw the plot for outliers using code.

|  |
| --- |
| hr\_non\_cat\_columns.plot(kind='box',subplots=True,layout=(5,5),figsize=(30,25))  plt.show()  hr\_descrete\_columns.plot(kind='box',subplots=True,layout=(5,5),figsize=(30,25))  plt.show() |



Outliers are present in following columns;

MonthlyIncome

TotalWorkingYears

YearsAtCompany

YearsSinceLastPromotion

TrainingTimesLastYear

NumCompaniesWork

PerformanceRating

YearsInCurrentRole

After checking an Outliers, we have to remove this. We removing the Outliers by using Z score Technique.

|  |
| --- |
| from scipy.stats import zscore  z = np.abs(zscore(hr))  z.shape |

After importing Z score technique dataset have 1470 rows and 31 columns.

|  |
| --- |
| hr = hr[(z<3).all(axis=1)]  hr.shape |

After removing an outlier from dataset, now, we have left with 1387 rows and 31 columns and we loss 5.64 percent data.

**VIF –**

VIF stands for the Variance Inflation Factor. It is used to detect the multicollinearity in given data. For performing VIF we need both independent and dependent columns. VIF is only applicable for numerical variables.

|  |
| --- |
| from statsmodels.stats.outliers\_influence import variance\_inflation\_factor  def calculate\_vif(dataset):  vif = pd.DataFrame()  vif['Features'] = dataset.columns  vif['VIF\_Values'] = [variance\_inflation\_factor(dataset.values,i) for i in range(dataset.shape[1])]  return(vif.sort\_values(by='VIF\_Values',ascending = False)) |

|  |
| --- |
| calculate\_vif(hr\_non\_cat\_columns) |

Features VIF\_Values

1 HourlyRate 5.868839

0 DailyRate 4.070855

3 MonthlyRate 4.045189

2 MonthlyIncome 2.668589  
  
  
We set the threshold for VIF as 10, We can see that all the values are under threshold.

**Balancing the Dataset –**

For finding the best model the target variable (independent variable) must be balanced. In this dataset target variable is imbalanced, number of employees who said yes is less than the number of employees who said no. So first we have to balanced the data means yes and no should have number.

First, we split the data in hrx and hry.

|  |
| --- |
| hrx = hr.drop('Attrition',axis=1)  hrx.shape  hry = hr['Attrition']  hry.shape |

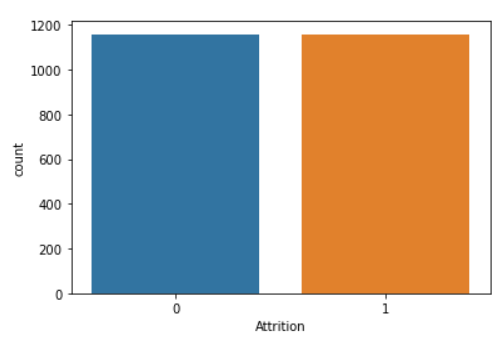
Shape of hrx = (1387,30)

Shape of hry = (1387)

Now we can balance the data by using SMOTE.

|  |
| --- |
| from imblearn.over\_sampling import SMOTE  sm = SMOTE()  x,y = sm.fit\_resample(hrx,hry) |

|  |
| --- |
| print(y.shape)  print(y.value\_counts())  print()  sns.countplot(y)  plt.show() |



From above plot we can see that data is balanced now.

Now we have to normalized the data by using power transform that is we have to remove the skewness in data.

|  |
| --- |
| from sklearn.preprocessing import power\_transform  x\_new = power\_transform(x) |

|  |
| --- |
| x[descrete\_columns].skew(),x[non\_cat\_columns].skew() |

Age -0.001910

DistanceFromHome -0.042867

Education -0.074288

EnvironmentSatisfaction -0.111747

JobInvolvement -0.060722

JobSatisfaction -0.122397

NumCompaniesWorked 0.002591

PercentSalaryHike 0.101750

PerformanceRating 0.000000

RelationshipSatisfaction -0.123212

StockOptionLevel 0.341529

TotalWorkingYears -0.020121

TrainingTimesLastYear 0.073077

WorkLifeBalance -0.048880

YearsAtCompany -0.016395

YearsInCurrentRole -0.031622

YearsSinceLastPromotion 0.263779

YearsWithCurrManager -0.030095

DailyRate -0.151564

HourlyRate -0.077303

MonthlyIncome 0.052030

MonthlyRate -0.178599

We set the threshold for skewness at 0.75, now we can see that all the values are within the threshold and the skewness has removed.

**BUILDING MACHINE LEARNING MODEL**

Basically, there are two types of models which we used for machine learning programs;

* Classification Model
* Regression Model

Hr dataset is a Classification dataset so we built a classification model. For building a machine learning model first we have to import some important library and then find a Best Random state for Best Accuracy score.

|  |
| --- |
| from sklearn.linear\_model import LogisticRegression  from sklearn.metrics import accuracy\_score  from sklearn.metrics import confusion\_matrix,classification\_report  from sklearn.model\_selection import train\_test\_split  from sklearn.model\_selection import cross\_val\_score |

|  |
| --- |
| maxAccu = 0  maxRs = 0  for i in range(1,100):  x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=.30,random\_state=i)  lr.fit(x\_train,y\_train)  predhr = lr.predict(x\_test)  acc = accuracy\_score(y\_test,predhr)  if acc>maxAccu:  maxAccu=acc  maxRs = i  print('Best Accuracy is',maxAccu,'on Random state',maxRs) |

So, we got a best Accuracy score – 0.8647

On Random state – 75

Now we have to split the data into training and testing format; x\_train,x\_test,y\_train,y\_test.

|  |
| --- |
| x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=.30,random\_state=75) |

We send 30 percent data for testing and 70 percent data for training with random state 75,

* x\_train.shape - (1621,30)
* x\_test,shape - (695,30)
* y\_train - (1621)
* y\_test - (695)

We split the data into x and y. x\_train has 1621 rows and 30 columns, x\_test has 695 rows and 30 columns, y\_train has 1621 rows and y\_test has 695 rows.

Now we test the different machine learning model.

|  |
| --- |
| def Model(model):  model.fit(x\_train,y\_train)  pred = model.predict(x\_test)  print('Accuracy',accuracy\_score(y\_test,pred))  print(classification\_report(y\_test,pred))  scr1 = cross\_val\_score(model,x,y,cv=5)  print('Cross validation score :',scr1.mean()) |

* Logistic Regression

|  |
| --- |
| from sklearn.linear\_model import LogisticRegression  lr = LogisticRegression()  Model(lr) |

Accuracy score for Logistic regression is – 0.8647

Cross Validation Score is – 0.7988

* Decision Tree Classifier

|  |
| --- |
| from sklearn.tree import DecisionTreeClassifier  dt = DecisionTreeClassifier()  Model(dt) |

Best Accuracy score for Decision tree classifier – 0.8057

Cross Validation score – 0.6459

* Random Forest Classifier

|  |
| --- |
| from sklearn.ensemble import RandomForestClassifier  rf = RandomForestClassifier()  Model(rf) |

Best Accuracy Score for Random Forest – 0.9064

Cross Validation score – 0.8260

* Service Vector Classifier

|  |
| --- |
| from sklearn.svm import SVC  svc = SVC()  Model(svc) |

Accuracy Score for SVC – 0.9021

Cross Validation score – 0.8601

We find the Accuracy score and Cross Validation score for different classification models. Random Forest classifier model has a highest Accuracy score than other models. We choose Random Forest classifier as our final model.

**Hyper Parameter Tuning –**

Hyper parameter tuning is used to improving the Accuracy of machine learning model. In hyper parameter tuning we used grid search cv method for finding the best parameters for model. Every machine learning model has different Hyperparameters.

|  |
| --- |
| RandomForestClassifier() |

RandomForestClassifier()

Now we import Grid Search CV technique from sklearn model. Grid search cv is used to find the best parameters for model. Grid search cv choses the best parameters from which we included in our parameter grid.

|  |
| --- |
| from sklearn.model\_selection import GridSearchCV  parameters = {'criterion' : ["gini", "entropy"],  'max\_depth' : [2,4,6,8,10],  'max\_features' : ["auto", "sqrt", "log2"],  'class\_weight' : ["balanced", "balanced\_subsample"]} |

|  |
| --- |
| GSC = GridSearchCV(RandomForestClassifier(),parameters,cv=5,scoring='accuracy')  GSC.fit(x\_train,y\_train)  GSC.best\_params\_ |

'class\_weight': 'balanced',

'criterion': 'entropy',

'max\_depth': 10,

'max\_features': 'log2'

We got the best parameters for Random Forest classifier. Now we find the best Accuracy score for the Random Forest model.

|  |
| --- |
| GSC\_pred = GSC.best\_estimator\_.predict(x\_test)  accuracy\_score(y\_test,GSC\_pred) |

Accuracy score after Hyper parameter tuning is – 0.9107

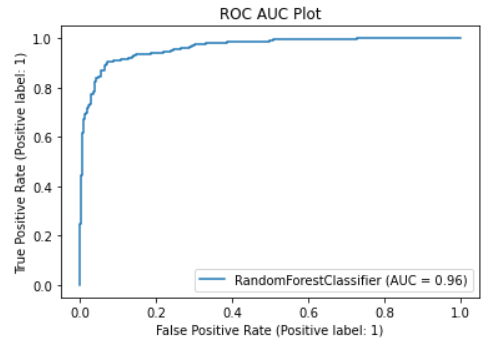
After using hyper parameter tuning Accuracy score increased by one percent. So, we choose the Random Forest model with Hyper parameter tuning.

**CONCLUDING REMARK –**

Final step for this is to plot ROC AUC plot as a conclusion.

ROC stands for the Receiver Operating Characteristic curve and AUC stands for Area Under Curve. ROC is a curve showing the performance of a model in a graphical format. It plots the graph between True positive rate and False positive rate. ROC curve is only used for binary classification problems.

|  |
| --- |
| from sklearn.metrics import plot\_roc\_curve  plot\_roc\_curve(GSC.best\_estimator\_,x\_test,y\_test)  plt. title('ROC AUC Plot')  plt.show() |



This is the ROC AUC curve for hr dataset. Plot is drawn between True positive rate and False positive rate (True positive rate vs False positive rate).