**HR Analytics Dataset**

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.

**Human Resource 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:**

Attrition in human resources refers to the gradual loss of employees over time. 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.

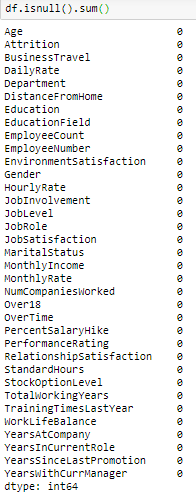
The objective of this project is to analyze and predict the employee will be leaving the company or not using HR Analytics and applying machine learning algorithms. It also aims at factors impacting the attrition of employees in the organization.

Limitation of this project is that the observations and predictions made are based only on the provided dataset.

* **Size of the dataset:** 1470 rows \* 35 columns
* **Some of the input variables(features) are:**
  + Age
  + Gender
  + Department
  + Education Field
  + Job Level
  + Job Role
  + Job satisfaction
  + Monthly Income
  + Overtime
  + Performance Rating
  + Total years of working
* **Output variable:** Attrition

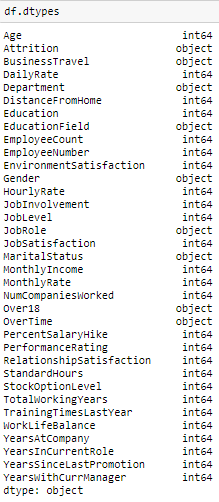
**Data Cleaning and Pre-processing:**

**C**hecking for null values:



From the above output it can be seen that there are no missing values in the dataset.

Checking the datatypes of variables:

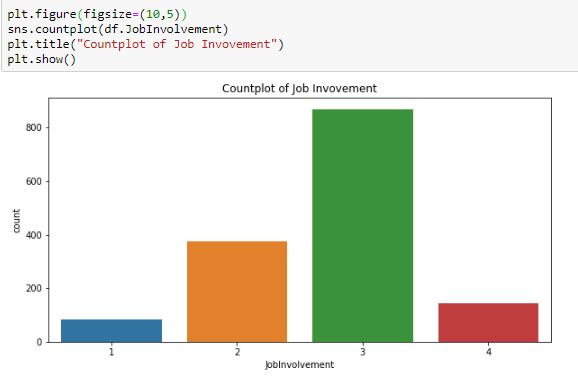


From the above output it can be seen that all the variables are either object or of integer datatype.

**Exploratory Data Analysis:**

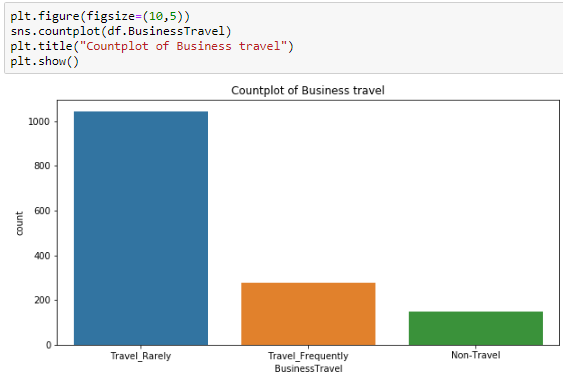
Univariate analysis:

1. Countplot of Job involvement:



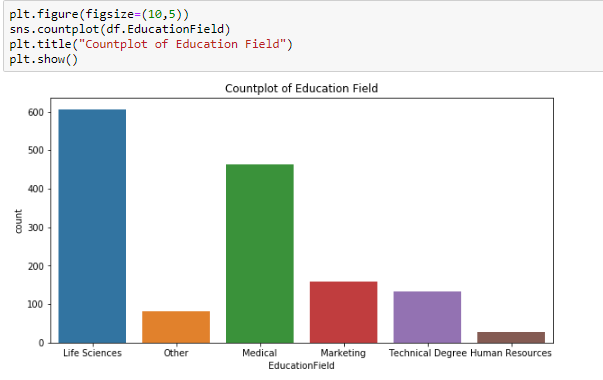
The above barplot shows that most of the employees have high job involvement.

1. Countplot of Business Travel:



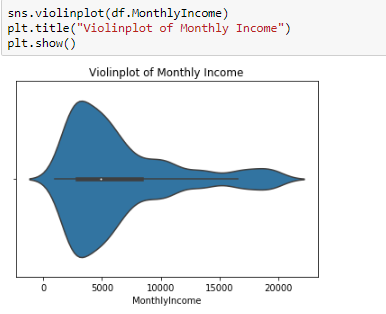
From the above barplot, it can be seen that most of the employees travel rarely.

1. Countplot of Education Field:



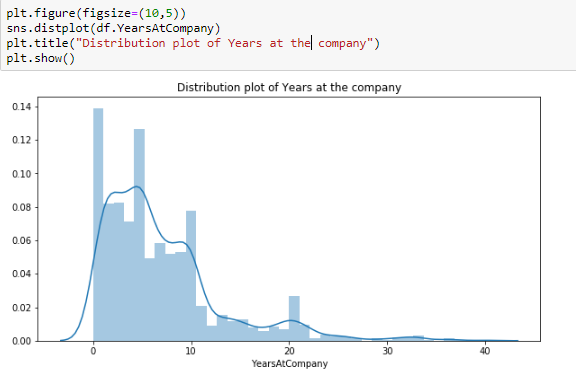
From the above barplot, it can be observed that Life sciences and Medical are the prominent education fields of the employees.

1. Violin plot of monthly income:



From the above violin plot, it can be observed that most of the employees have monthly income within 7500-8000

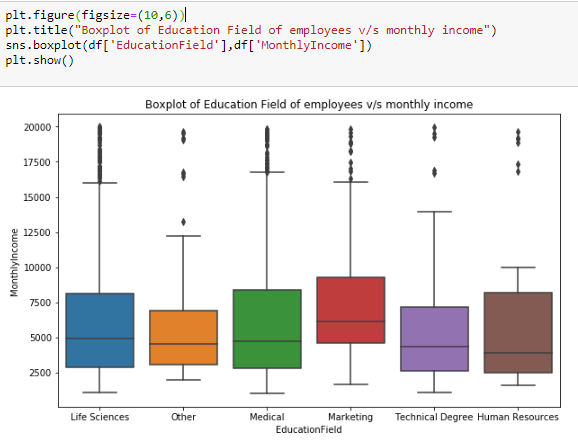
1. Distribution plot of Years at the company:



From the above distribution plot it can be observed that most of the employees have spent utmost 10 years in the company.

Bivariate Analysis:

1. Boxplot of Education Field of employees v/s monthly income:



From the above barplot it can be observed that employees from Marketing field have higher monthly income as compared to other employees with other educational background.

1. Barplot of Over Time v/s monthly income:



From the above barplot it can be observed that there no much difference in monthly income of employees who are working overtime and who aren’t working overtime.

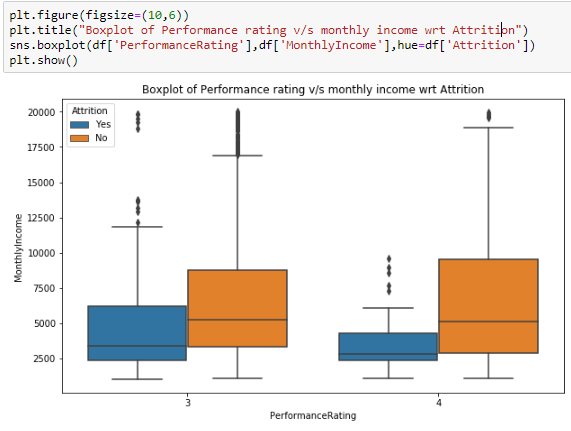
1. Barplot of Department v/s monthly income:



From the above barplot it can be observed that Sales wing is having the highest income as compared to other two wings.(HR and R&D)

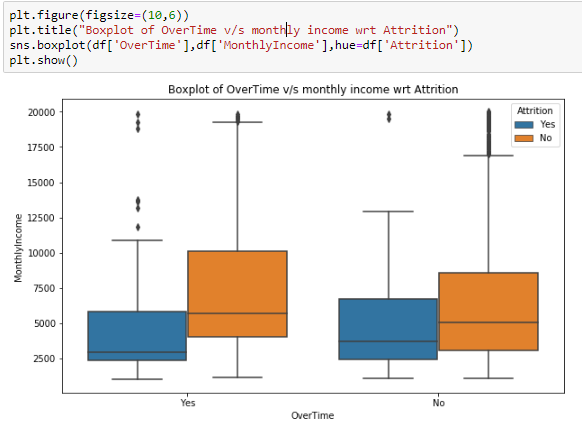
Multivariate analysis:

1. Boxplot of Performance rating v/s monthly income with respect to Attrition:



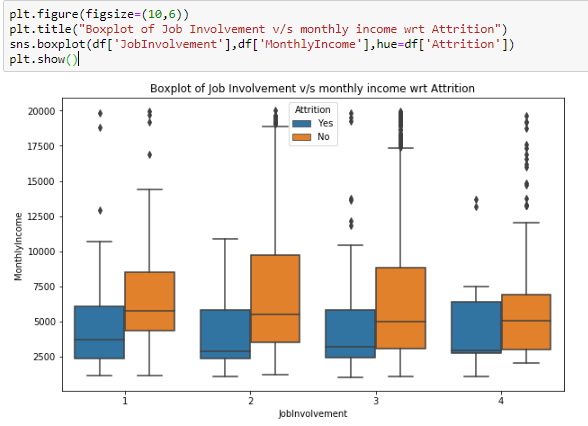
From the above boxplot, it can be observed that employees with higher performance rating are leaving the company.

1. Boxplot of Overtime v/s monthly income with respect to Attrition:



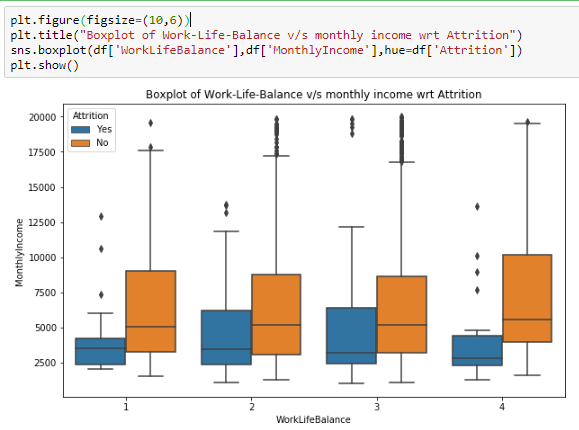
From the above boxplot, it can be observed that employees who don’t do overtime are still with the company.

1. Boxplot of Job Involvement v/s monthly income with respect to Attrition:



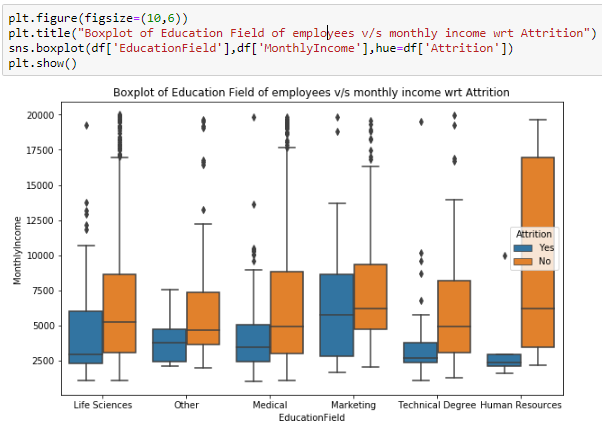
From the above boxplot it can be observed that employees who show medium involvement in job have high attrition rates as compared to others.

1. Boxplot of Work-Life-Balance v/s monthly income with respect to Attrition:



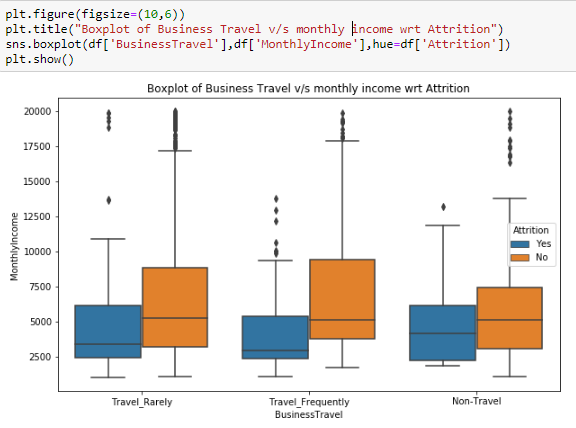
From the above boxplot, it can be observed that employees with better work life balance are leaving the company more.

1. Boxplot of Education Field of employees’ v/s monthly income with respect to Attrition:



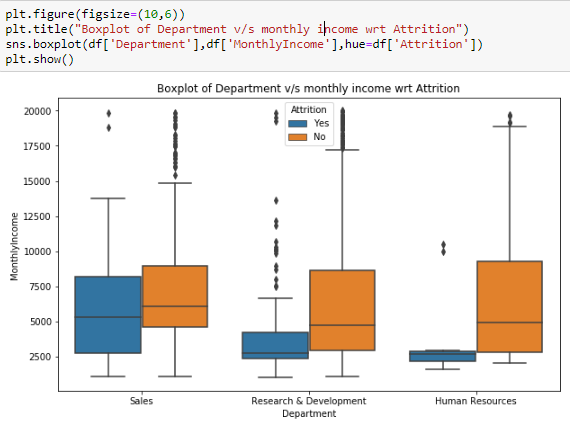
From the above boxplot, it can be observed that Marketing field has the highest attrition and HR has the lowest attrition. Also HR has the highest pay as compared to other fields.

1. Boxplot of Business Travel v/s monthly income with respect to Attrition:



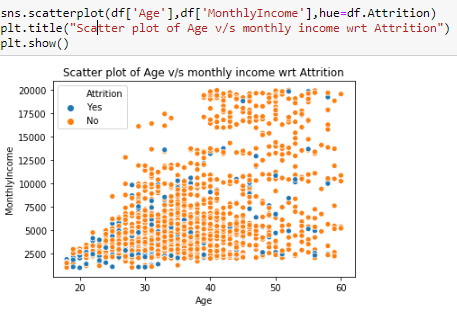
From the above boxplot, it can be observed that employees who don’t travel have more attrition rate as compared to employees to travel frequently/rarely for business purpose.

1. Boxplot of Department v/s monthly income with respect Attrition



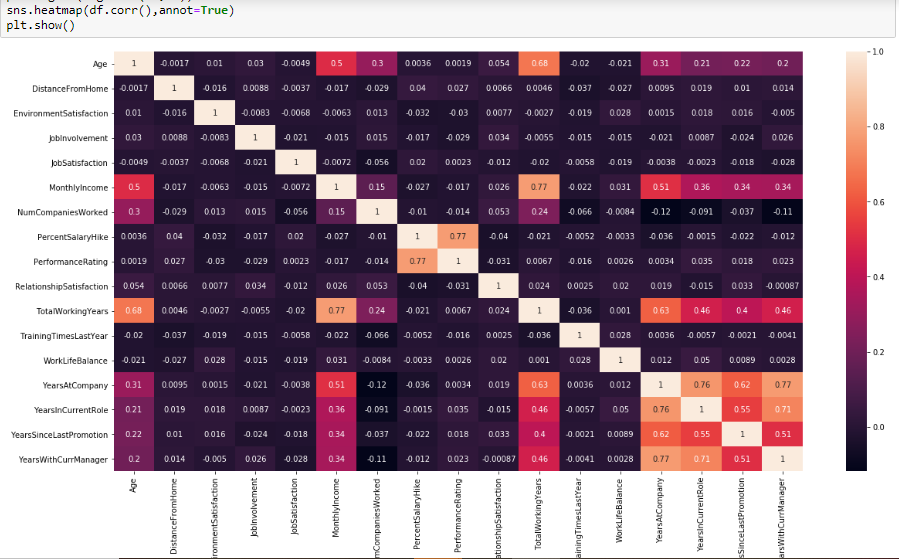
From the above boxplot, it can be seen that sales is having the highest attrition as compared to other fields.

1. Scatter plot of Age v/s monthly income with respect to Attrition:



From the above scatter plot it can be observed that attrition is slight more among the employees of age below 35 and monthly income below 10,000.

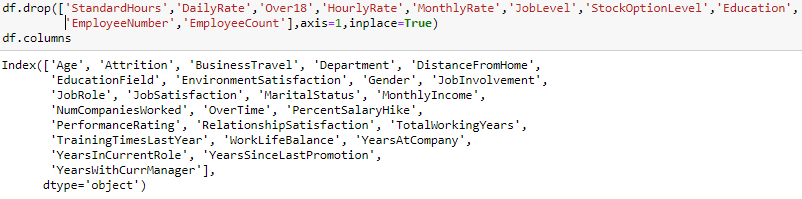
1. Heatmap showing correlation between different variables:



From the above heatmap following observations can be made:

1. Total Working years and monthly are very positively correlated.
2. There is good positive correlation between total working years and age.
3. Total working years and years at company are also positively correlated to each other.
4. Years with current manager and years spent in company are very positively correlated.
5. Number of companies worked and years at company are negatively correlated.

**Dropping columns that won’t impact the output variable:**



**Converting categorical columns into numerical columns:**

**Two methods are used here to perform the above task:**

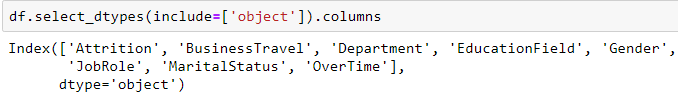
1. **Label Encoding:**

Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

1. **One Hot encoding:**

One-hot encoding is essentially the representation of categorical variables as binary vectors. These categorical values are first mapped to integer values. Each integer value is then represented as a binary vector that is all 0s (except the index of the integer which is marked as 1).

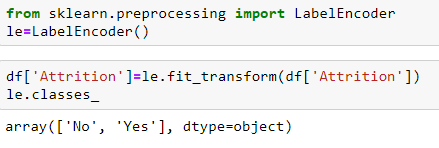
For this project, after dropping non-required columns we are left with the following columns which are of object type:



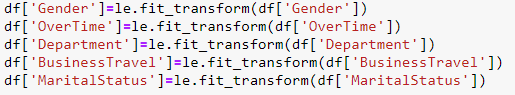
Now all these columns which are categorical needs to be converted into numerical columns using Label Encoding or One hot encoding techniques.

**Using Label Encoding:**

* For Attrition (Output variable):

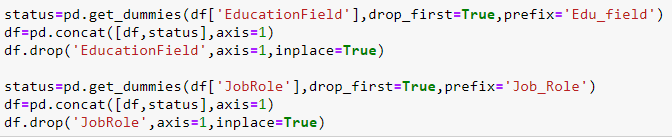


* Similarly, for Gender, overtime, Department, Business Travel, Marital status:

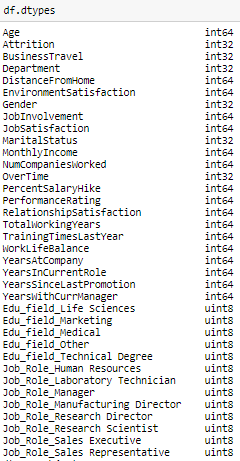


**Using One Hot encoding:**

For Educational field and Job Role columns:



Now we shall check the datatypes of all the variables. All the variables should be of numeric type.



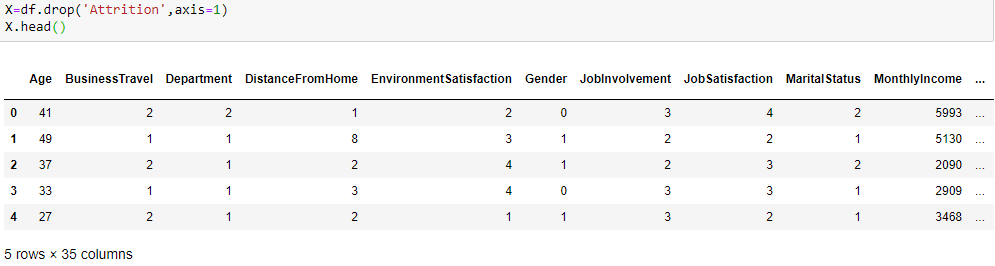
From the above output it can be confirmed that both the input variables as well as the output variables are numerical data.

**Outlier detection and Treatment:**

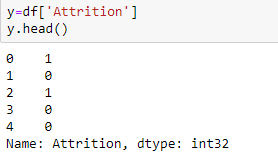
Since in this dataset there are no continuous variables, outlier treatment won’t be required. So now we can proceed with model building.

**Model Building:**

The first step in model building is defining the input variables and the output variable. For this purpose, we first drop the output variable and make all the variables as input variables and name it as a Dataframe ‘X’:



Further we will assign the output variable(Attrition) to a Dataframe and name it as ‘y’:



Now our input and output variables are separated, next task is to split the dataset into train and test data. For this we use Train\_test\_split function which is present in the scikit learn library.It splits the data arrays into two subsets: for training data and for testing data.



Here the test size will be 30% of the dataset and the remaining 70% will be train dataset size. First we fit the model to the training dataset and then we predict the model accuracy and other metrics on the test dataset.

Some metrics which is used for classification to evaluate the model performance are:

* Accuracy score
* Confusion Matrix
* Classification Report
* AUC and ROC

**Accuracy score:**

**Accuracy** is the fraction of predictions our model got right. Formally, accuracy has the following definition:

Accuracy =

Accuracy =

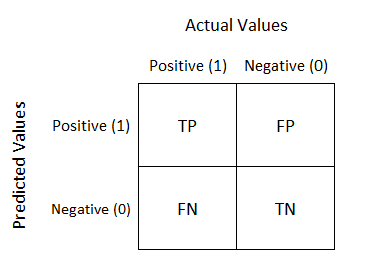
Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

A **true positive** is an outcome where the model correctly predicts the positive class. Similarly, a **true negative** is an outcome where the model correctly predicts the negative class.

A **false positive** is an outcome where the model incorrectly predicts the positive class. And a **false negative** is an outcome where the model incorrectly predicts the negative class.

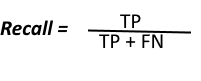
**Confusion Matrix:**

It is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.



It is extremely useful for measuring Recall, Precision, Specificity, Accuracy and most importantly AUC-ROC Curve

**Recall:**



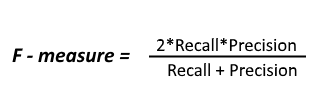
Out of all the positive classes, how much are predicted correctly. It should be high as possible.

**Precision:**



Out of all the positive classes predicted correctly, how many are actually positive.

**F – score:**



It is difficult to compare two models with low precision and high recall or vice versa. So to make them comparable, we use F-Score. F-score helps to measure Recall and Precision at the same time. It uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more.

**Classification Report:**

Classification report gives the summary of all the above metrics together. It consists of Recall, precision, f1score and support.

**AUC and ROC curve:**

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR (True Positive Rate)against FPR (False Positive Rate) at various threshold values and essentially separates the ‘signal’ from the ‘noise’. The Area Under the Curve (AUC)is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

**Higher the AUC, better the performance of the model at distinguishing between the positive and the negative classes.**

So first let’s see what is meant by True Positive Rate (TPR) and False Positive Rate (FPR).

**True Positive Rate (TPR):**

TPR is nothing but Recall. It’s also referred to as sensitivity. So it is out of all the positive classes, how much are predicted correctly.

**TPR** =

**False Positive Rate (FPR):**

FPR tells us what proportion of the negative class got incorrectly classified by the classifier.

**FPR** =

A lower FPR is desirable since negative class has to be classified correctly.

**Specificity/True Negative Rate:**

Specificity tells us what proportion of the negative class got correctly classified.

**TNR** =

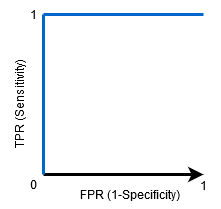
**False Negative Rate:**

False Negative Rate (FNR) tells us what proportion of the positive class got incorrectly classified by the classifier.

**FNR** =

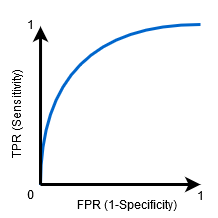
**Case 1:**

When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives, and all Positives as Negatives.



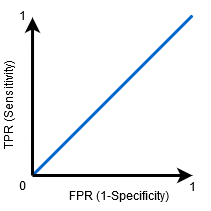
**Case 2:**

When 0.5<AUC<1, there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values. This is so because the classifier is able to detect more numbers of True positives and True negatives than False negatives and False positives.



**Case 3:**

When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points. Meaning either the classifier is predicting random class or constant class for all the data points.



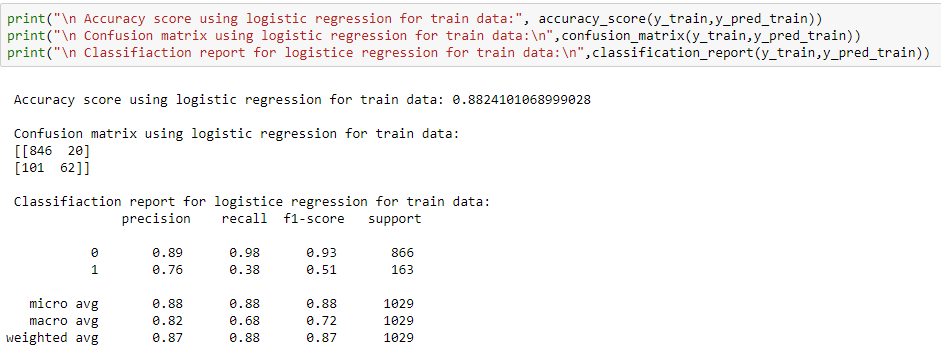
**GridserachCV:**

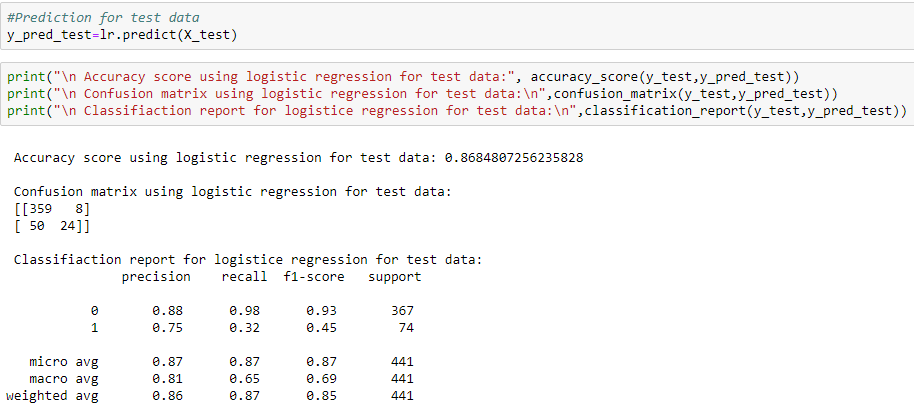
GridSearchCV is a library function that is a member of sklearn's model\_selection package. It helps to loop through predefined hyperparameters and fit the model on training set. So, in the end, best parameters from the listed hyperparameters can be selected to build the model.

So here we apply GridsearchCV to find the best parameters and then build the model with those best parameters.

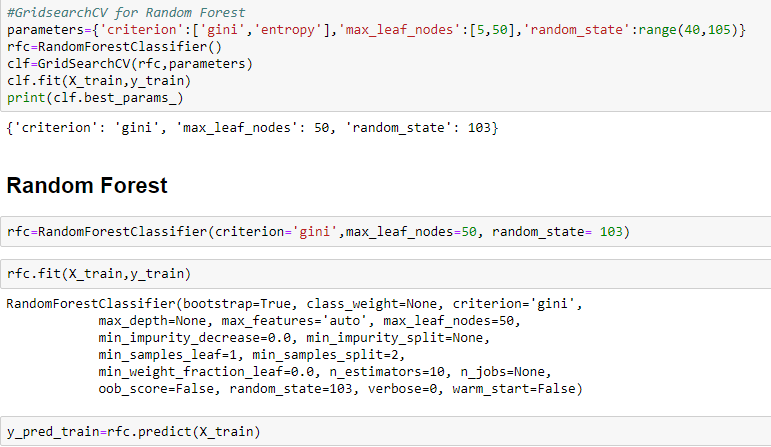
**Logistic Regression**:

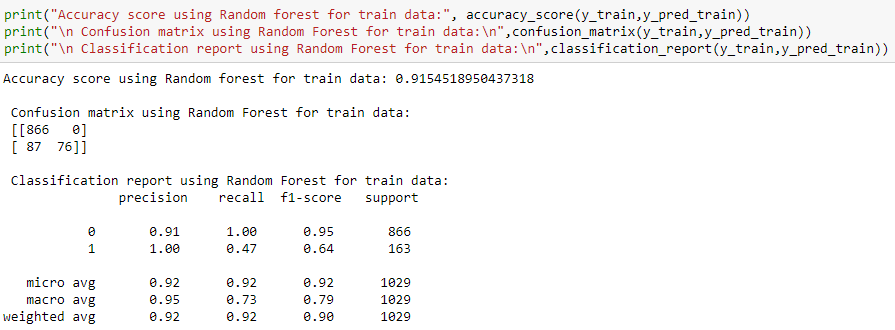


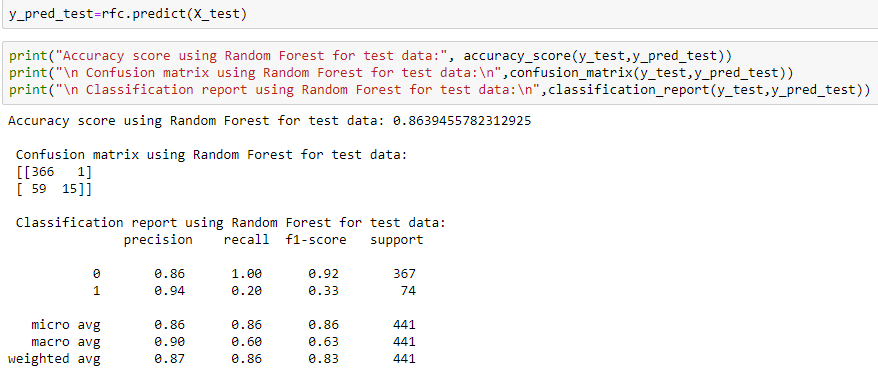




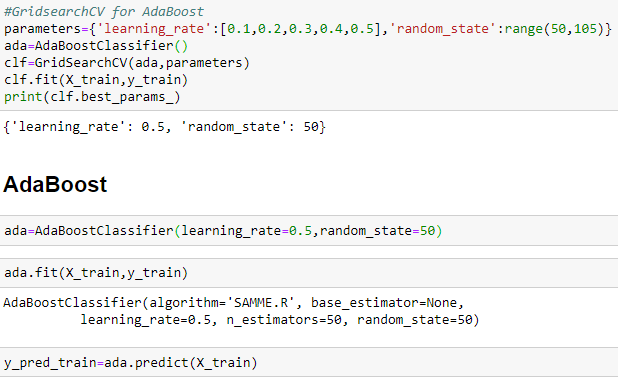
**Random Forest:**

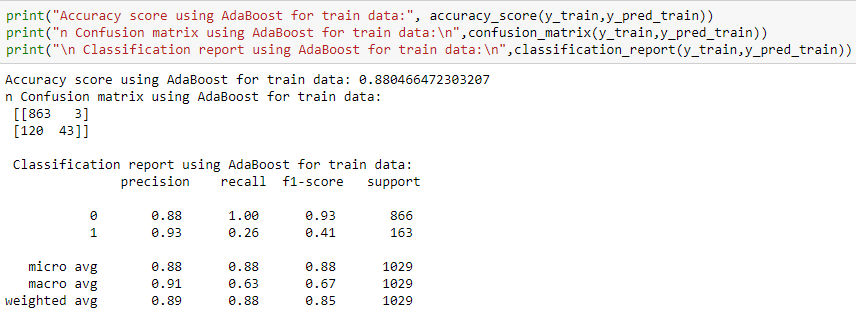


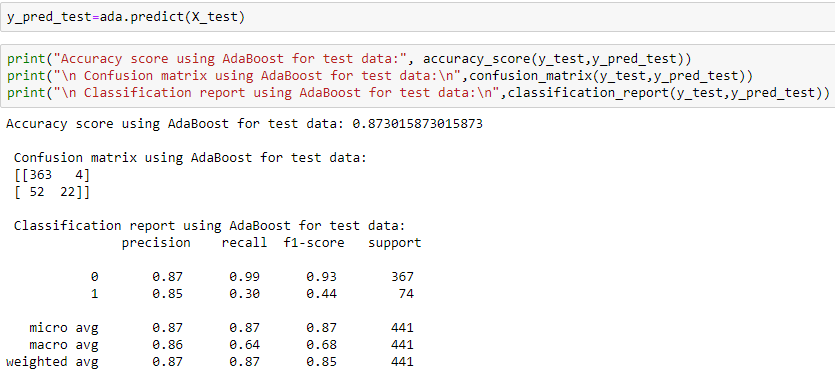




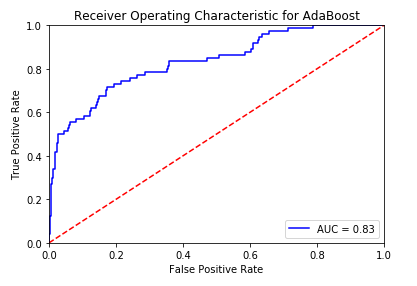
**AdaBoost Algorithm:**

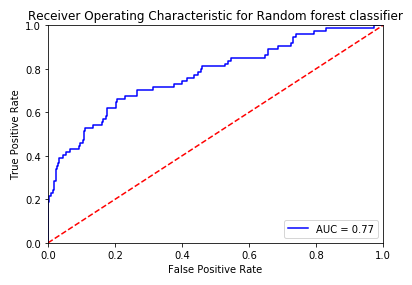


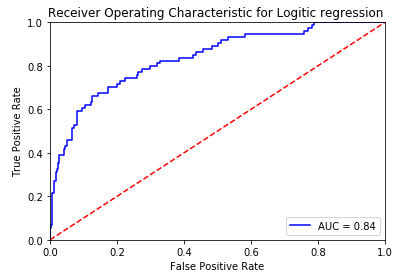




**AUC – ROC curve for above models:**

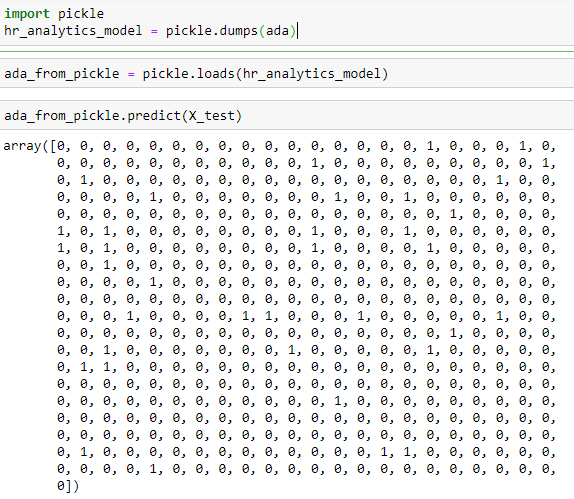






By comparing the evaluation results of the above models it can be observed that AdaBoost is giving better results under all the criteria (Accuracy, confusion matrix, recall, precision, f1-score and AUC-ROC curve).

**Serialization:**



**Findings from the project:**

1) Most of the employees travel rarely.

2) From the given data, it can be found that most of the employees are from life sciences and medical field.

3) Monthly income of most employees is within 10,000 units.

4) Very few employees are working in the company from past 20- 40 years. Most of them have been with the company with utmost 10 years.

5) Employees having higher performance rating are leaving the company(This interpretation may be only limited to this data).

6) Attrition is more among employees who work overtime.

7) Marketing field has the highest attrition while HR field is having the lowest attrition.

8) Employee who donot travel are more prone of leaving the company.

9) Attrition is more among employees of age less than 35 and who are having a monthly income less than 10,000 units.

**Mushroom Dataset**

A mushroom or toadstool is the fleshy, spore-bearing fruiting body of a fungus, typically produced above ground, on soil, or on its food source. Many species of mushrooms seemingly appear overnight, growing or expanding rapidly.

A mushroom may be edible, poisonous, or unpalatable.

Raw brown mushrooms are 92% water, 4% carbohydrates, 2% protein and less than 1% fat. In a 100 gram (3.5 ounce) amount, raw mushrooms provide 22 calories and are a rich source (20% or more of the Daily Value, DV) of B vitamins, such as riboflavin, niacin and pantothenic acid, selenium (37% DV) and copper (25% DV), and a moderate source (10-19% DV) of phosphorus, zinc and potassium (table). They have minimal or no vitamin C and sodium content. The vitamin D content of a mushroom depends on postharvest handling, in particular the unintended exposure to sunlight.

Some mushrooms are used or studied as possible treatments for diseases, particularly their extracts. Mushrooms can be used for dyeing wool and other natural fibers. They can also be used as fire starters. Mushrooms and other fungi play a role in the development of new biological remediation techniques.

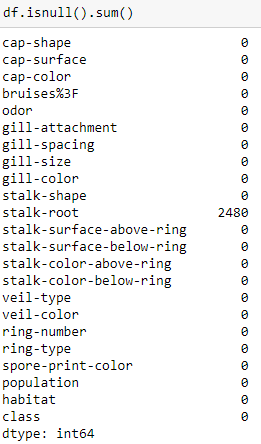
The objective of this project is to analyze the given dataset and predict whether given mushroom is edible or poisonous.

Limitation of this project is that the observations and predictions made are based only on the provided dataset.

* **Size of the dataset:** 8124 rows \* 23 columns
* **Input variables(features) are:**
  + cap-shape
  + cap-surface
  + cap-color
  + bruises%3F
  + odor
  + gill-attachment
  + gill-spacing
  + gill-size
  + gill-color
  + stalk-shape
  + stalk-root
  + stalk-surface-above-ring
  + stalk-surface-below-ring
  + stalk-color-above-ring
  + stalk-color-below-ring
  + veil-type
  + veil-color
  + ring-number
  + ring-type
  + spore-print-color
  + population
  + habitat
* **Output variable:** Class

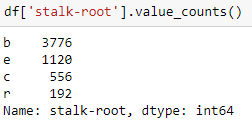
**Data Cleaning and Pre-processing:**

Checking for null values:



Only stalk-root has 2480 missing data.

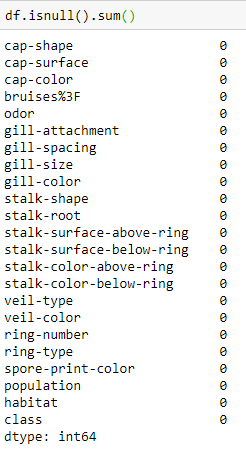
Let’s check the categories in stalk-root:



To balance the data, lets fill the missing values in stalk-root by value ‘r’ by using fillna method.



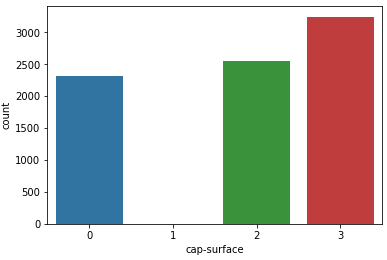
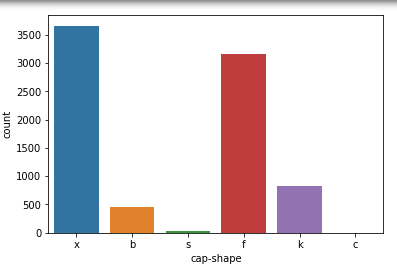
Confirming that NaN has been filled:

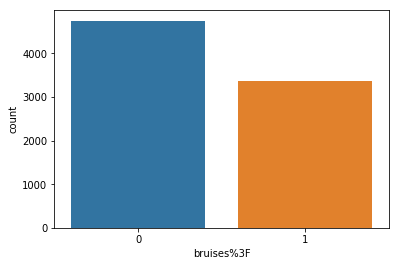
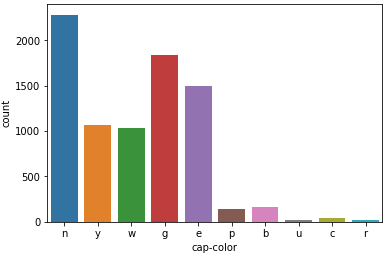


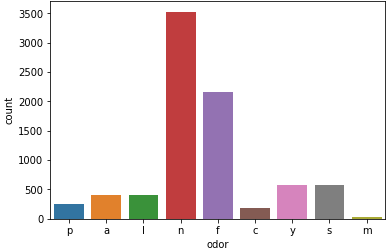
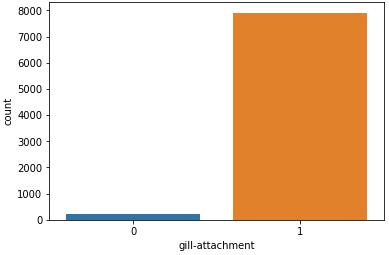
**Exploratory Data Analysis:**

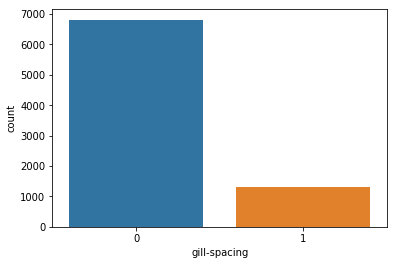
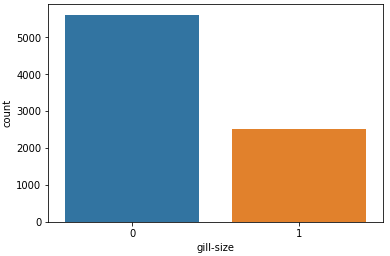
Univariate analysis:

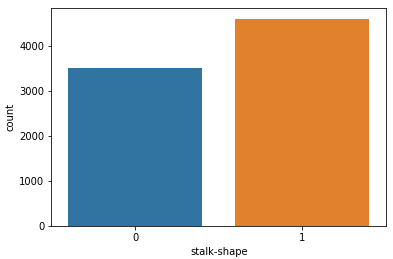
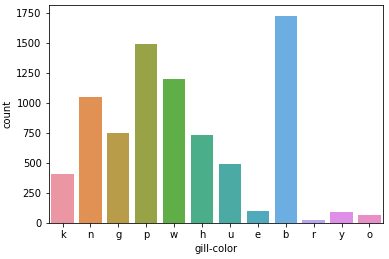


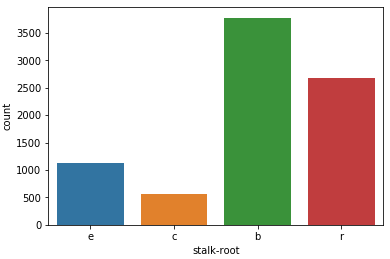
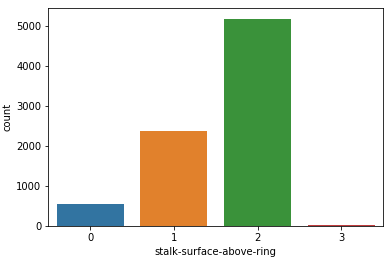


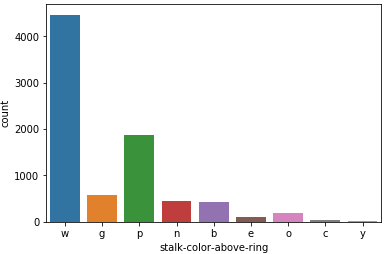
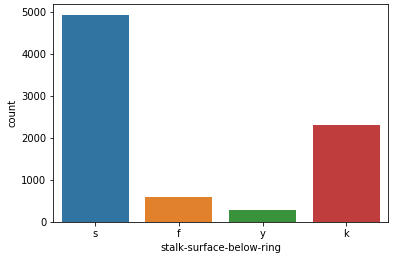


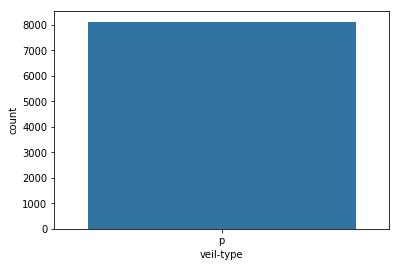
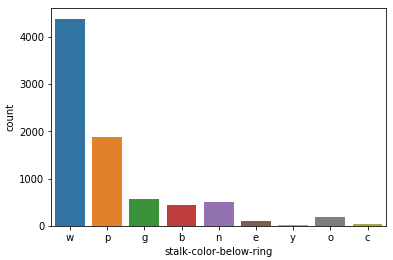


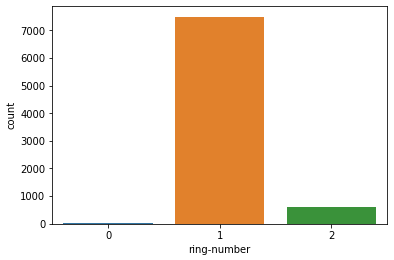
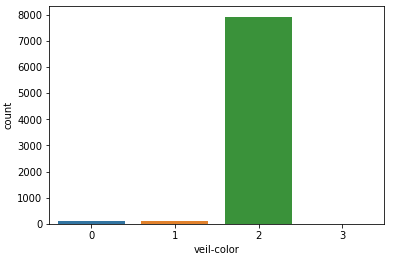


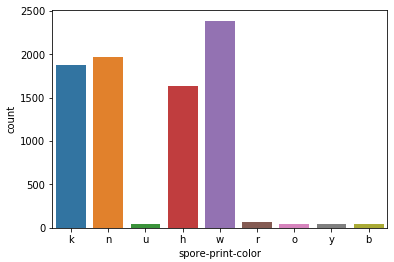
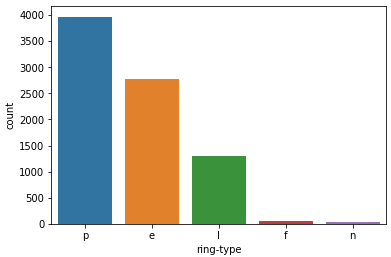


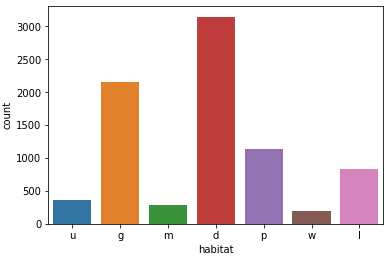
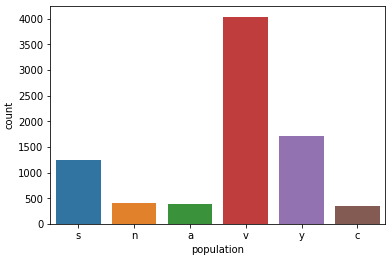


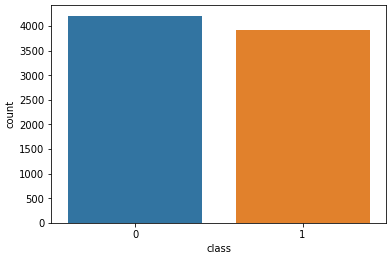












From the above countplots we can observe that veil-type has only one value throughout the dataset. So that column can be dropped as it doesn’t have any impact on the output variable.



**Converting categorical columns into numerical columns:**

**Two methods are used here to perform the above task:**

1. **Label Encoding:**

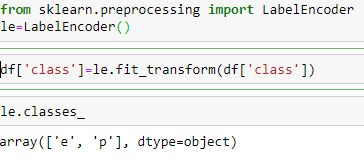
Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

1. **One Hot encoding:**

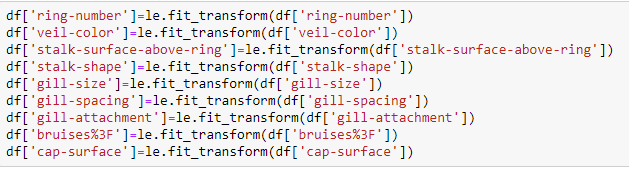
One-hot encoding is essentially the representation of categorical variables as binary vectors. These categorical values are first mapped to integer values. Each integer value is then represented as a binary vector that is all 0s (except the index of the integer which is marked as 1).

**Using Label Encoding:**

* For Class (Output variable):

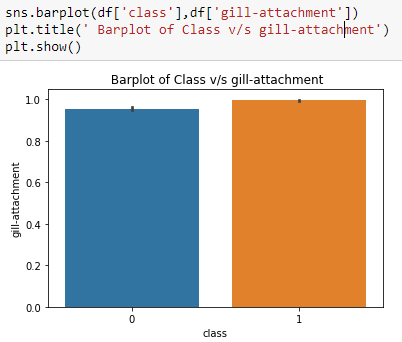


For other columns:



Bivariate and Multivariate Analysis:

1. Barplot of Class v/s gill-attachment:



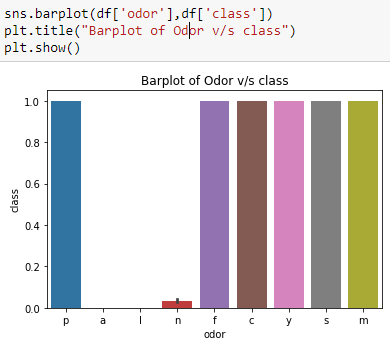
From the above barplot, we can see that class 1 mushrooms have more gill-attachement.

1. Barplot of gill-size v/s class:



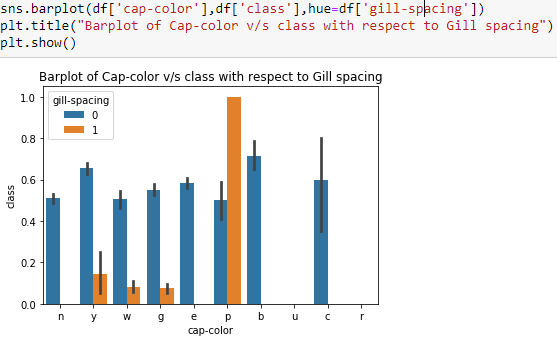
From the above barplot, it can be seen that gill-size of category 1, is more edible.

1. Barplot of Odor v/s class:



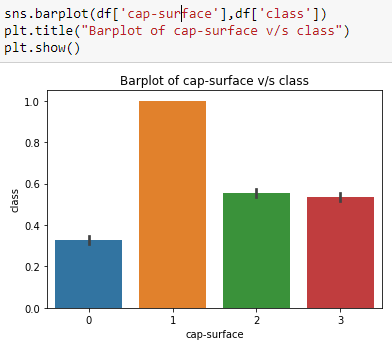
From the above barplot, it can be seen that apart from odor classes a,l,n all others are edible.

1. Barplot of Cap-color v/s class with respect to Gill spacing:



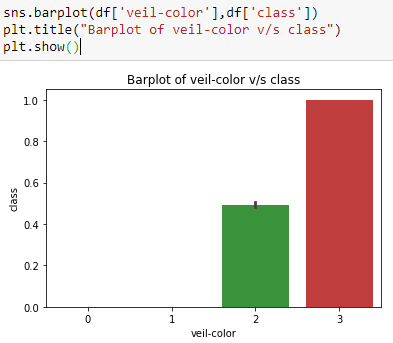
From the above graph, we can observe that cap-color with class 'p' and gill spacing of '1' are most edible ones.

1. Barplot of cap-surface v/s class:



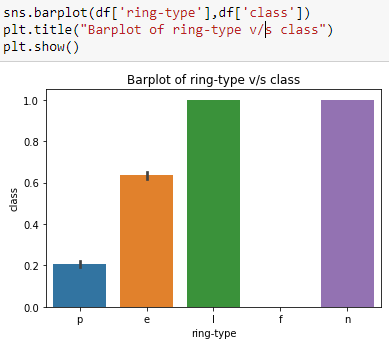
From the above graph, we can see that cap-surface with class 1 are the most edible as compared to other classes

1. Barplot of veil-color v/s class:



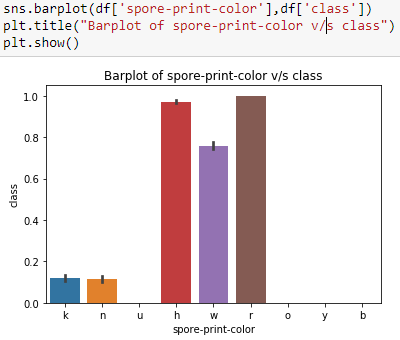
From the above barplot, it can be seen that veil-color with class 3 are the most edible mushrooms.

1. Barplot of ring-type v/s class:



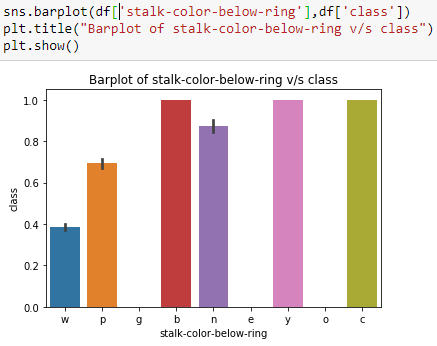
From the above barplot, it can be seen that ring-type with classes 'n' and 'l' are the edible ones and 'f' is poisonous one.

1. Barplot of spore-print-color v/s class:



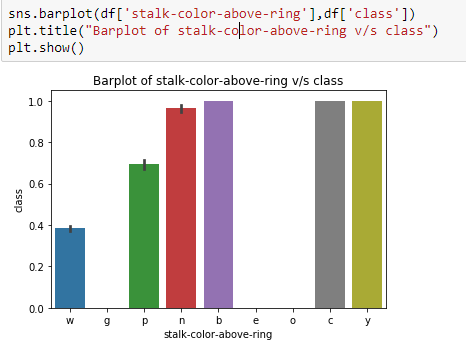
From the above barplot, we can see that spore-print-color with classes 'h','r' are more edible. Also class 'w' may be edible.

1. Barplot of stalk-color-below-ring v/s class:



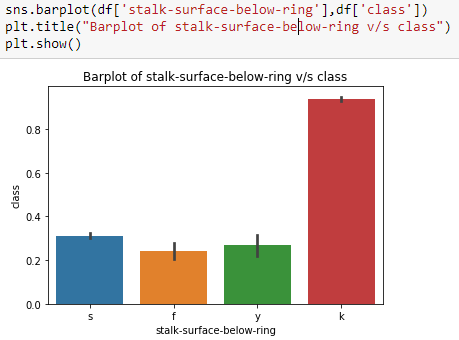
From the above barplot it can be observed that stalk-color-below-ring class b, y and c are edible.

1. Barplot of stalk-color-above-ring v/s class:



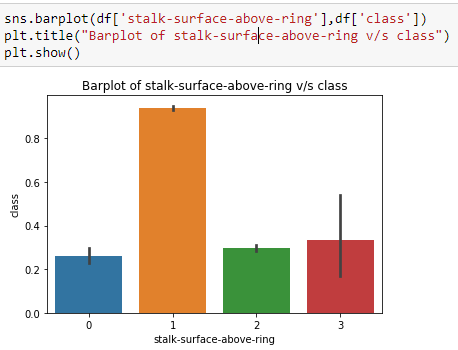
From the above barplot it can be observed that stalk-color-above-ring class b, y and c are edible.

1. Barplot of stalk-surface-below-ring v/s class:



From the above barplot we can observe that Stalk-surface-below-ring of class ‘k’ is most edible.

1. Barplot of stalk-surface-above-ring v/s class:



From the above barplot we can observe that Stalk-surface-above-ring of class ‘1’ is most edible.

**One hot encoding for remaining columns:**



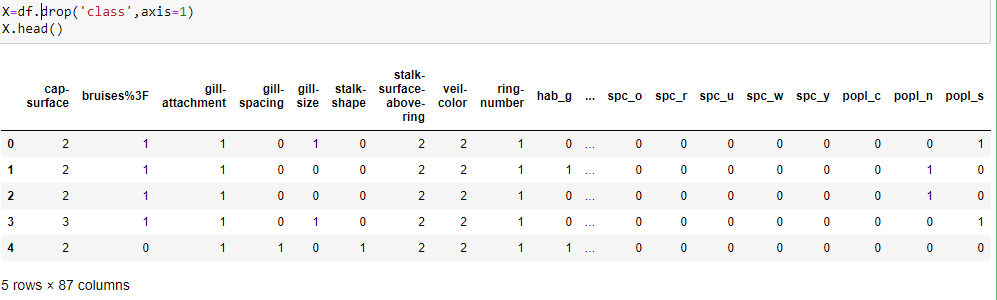
Now all the input variables as well as the output variable are converted into numerical data.

**Outlier detection and Treatment:**

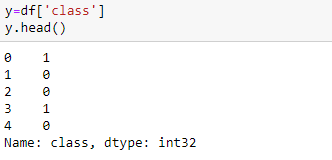
Since in this dataset there are no continuous variables, outlier treatment won’t be required. So now we can proceed with model building

**Model Building:**

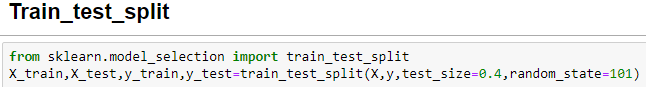
The first step in model building is defining the input variables and the output variable. For this purpose, we first drop the output variable and make all the variables as input variables and name it as a Dataframe ‘X’:



Further we will assign the output variable(Class) to a Dataframe and name it as ‘y’:



Now our input and output variables are separated, next task is to split the dataset into train and test data. For this we use Train\_test\_split function which is present in the scikit learn library.It splits the data arrays into two subsets: for training data and for testing data.



Here the test size will be 40 % of the dataset and the remaining 60% will be train dataset size. First we fit the model to the training dataset and then we predict the model accuracy and other metrics on the test dataset.

Some metrics which is used for classification to evaluate the model performance are:

* Accuracy score
* Confusion Matrix
* Classification Report
* AUC and ROC

**Accuracy score:**

**Accuracy** is the fraction of predictions our model got right. Formally, accuracy has the following definition:

Accuracy =

Accuracy =

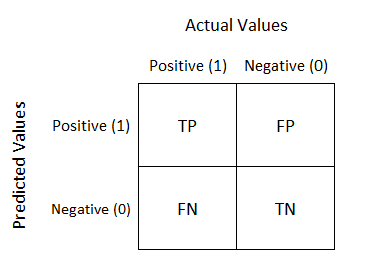
Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

A **true positive** is an outcome where the model correctly predicts the positive class. Similarly, a **true negative** is an outcome where the model correctly predicts the negative class.

A **false positive** is an outcome where the model incorrectly predicts the positive class. And a **false negative** is an outcome where the model incorrectly predicts the negative class.

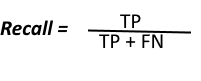
**Confusion Matrix:**

It is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.



It is extremely useful for measuring Recall, Precision, Specificity, Accuracy and most importantly AUC-ROC Curve

**Recall:**



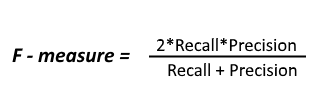
Out of all the positive classes, how much are predicted correctly. It should be high as possible.

**Precision:**



Out of all the positive classes predicted correctly, how many are actually positive.

**F – score:**



It is difficult to compare two models with low precision and high recall or vice versa. So to make them comparable, we use F-Score. F-score helps to measure Recall and Precision at the same time. It uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more.

**Classification Report:**

Classification report gives the summary of all the above metrics together. It consists of Recall, precision, f1score and support.

**AUC and ROC curve:**

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR (True Positive Rate)against FPR (False Positive Rate) at various threshold values and essentially separates the ‘signal’ from the ‘noise’. The Area Under the Curve (AUC)is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

**Higher the AUC, better the performance of the model at distinguishing between the positive and the negative classes.**

So first let’s see what is meant by True Positive Rate (TPR) and False Positive Rate (FPR).

**True Positive Rate (TPR):**

TPR is nothing but Recall. It’s also referred to as sensitivity. So it is out of all the positive classes, how much are predicted correctly.

**TPR** =

**False Positive Rate (FPR):**

FPR tells us what proportion of the negative class got incorrectly classified by the classifier.

**FPR** =

A lower FPR is desirable since negative class has to be classified correctly.

**Specificity/True Negative Rate:**

Specificity tells us what proportion of the negative class got correctly classified.

**TNR** =

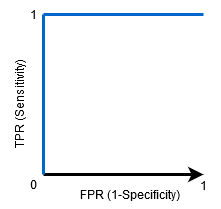
**False Negative Rate:**

False Negative Rate (FNR) tells us what proportion of the positive class got incorrectly classified by the classifier.

**FNR** =

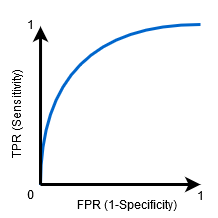
**Case 1:**

When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives, and all Positives as Negatives.



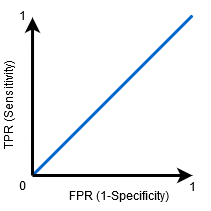
**Case 2:**

When 0.5<AUC<1, there is a high chance that the classifier will be able to distinguish the positive class values from the negative class values. This is so because the classifier is able to detect more numbers of True positives and True negatives than False negatives and False positives.



**Case 3:**

When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points. Meaning either the classifier is predicting random class or constant class for all the data points.

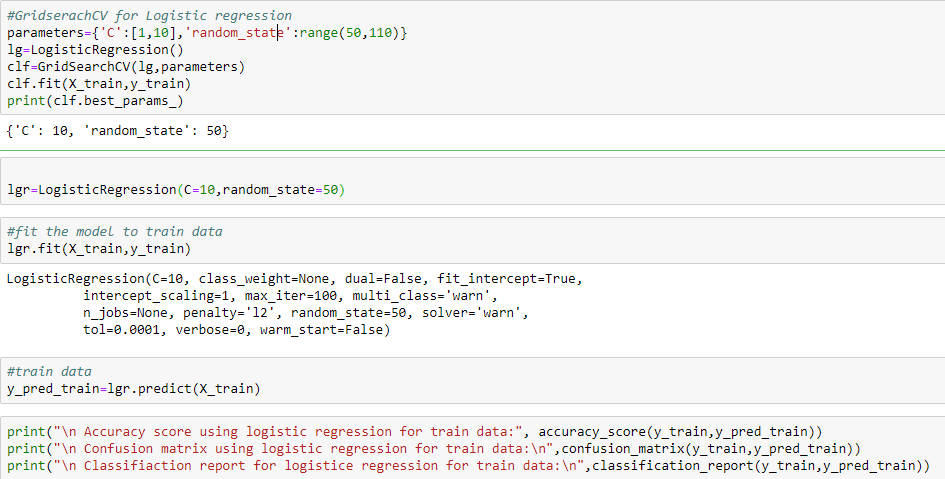


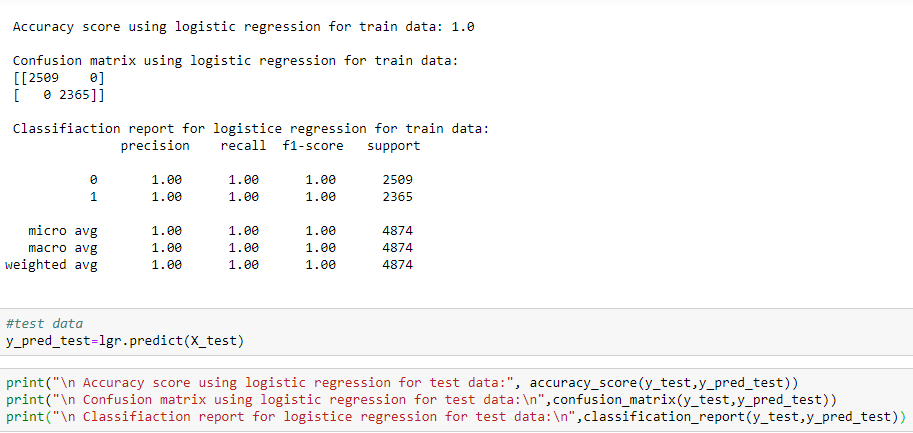
**GridserachCV:**

GridSearchCV is a library function that is a member of sklearn's model\_selection package. It helps to loop through predefined hyperparameters and fit the model on training set. So, in the end, best parameters from the listed hyperparameters can be selected to build the model.

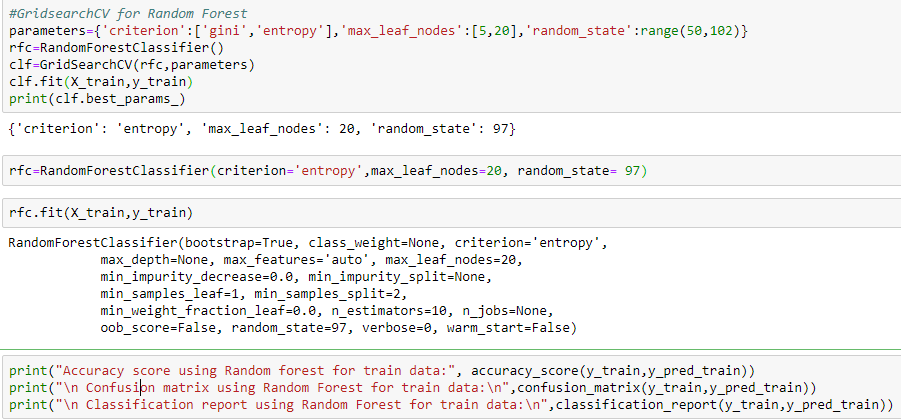
So here we apply GridsearchCV to find the best parameters and then build the model with those best parameters.

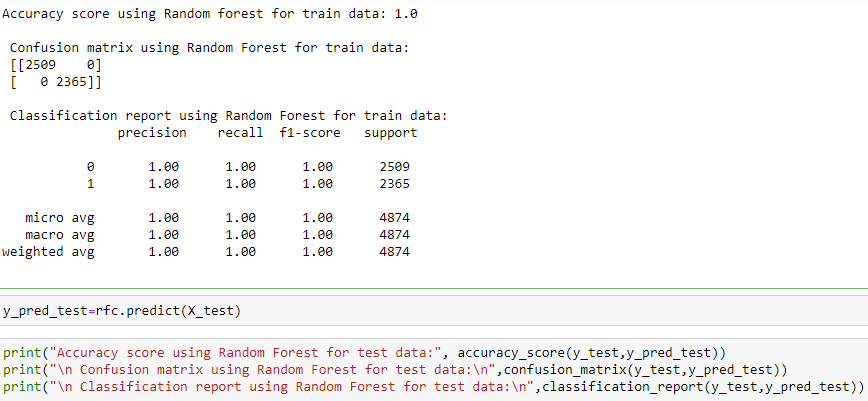
**Logistic Regression:**

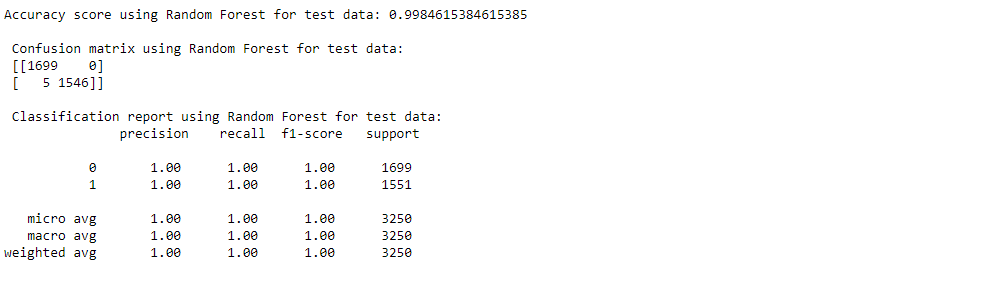




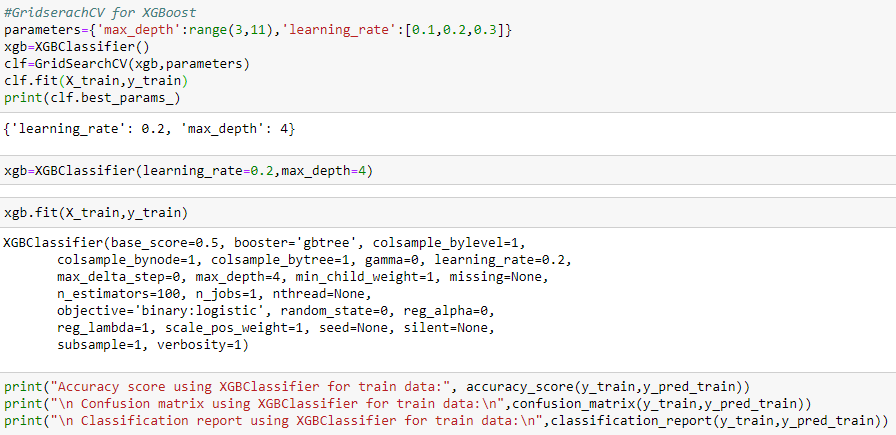
**Random Forest classifier:**

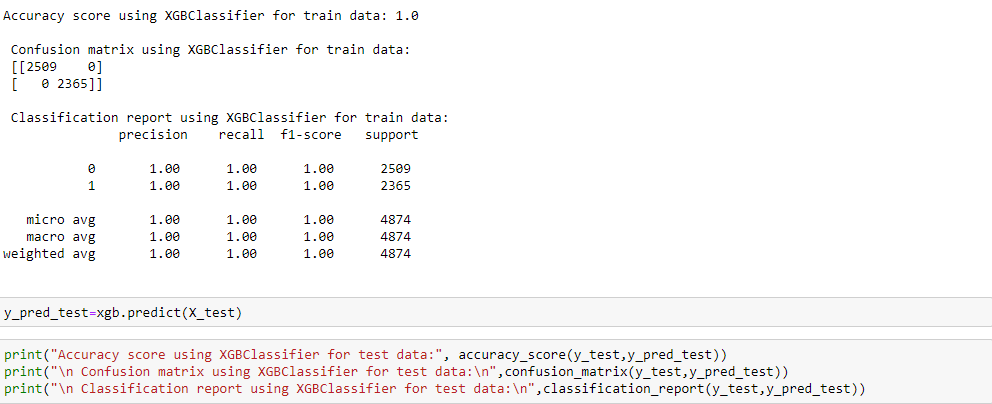


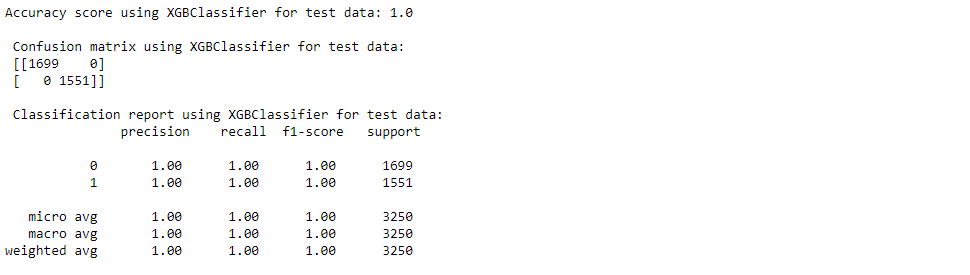




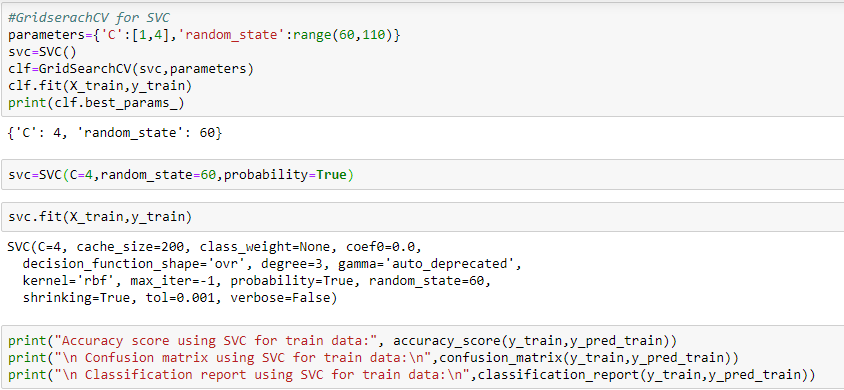
**XGBoost Classifier:**

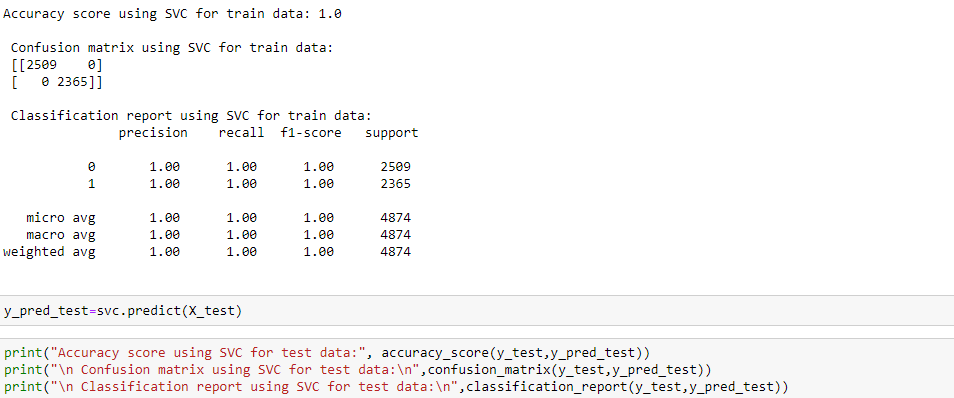


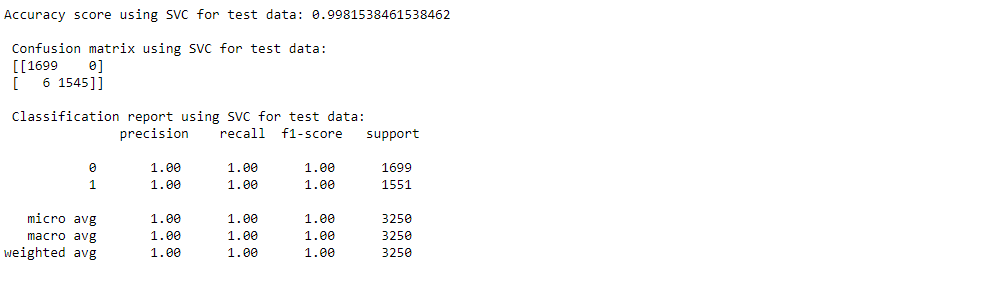




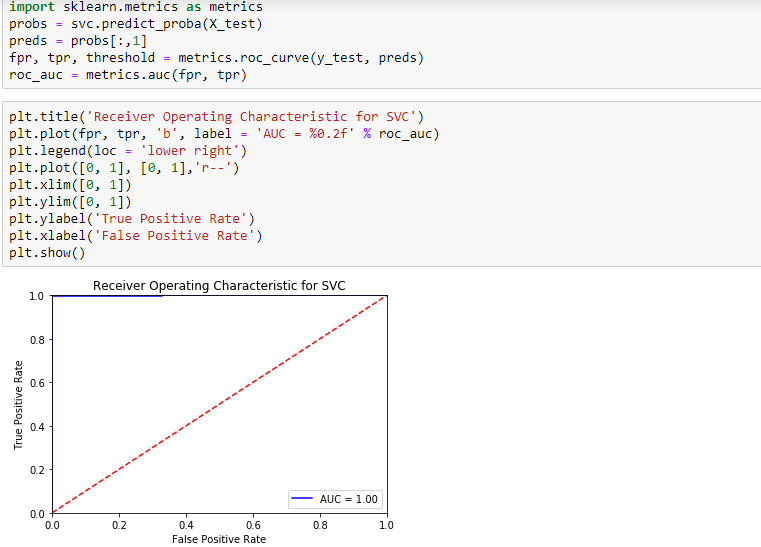
**Support Vector Classifier:**







**AUC-ROC curve for Support Vector Classifier:**



By comparing the evaluation results of the above models it can be observed that SVC is giving better results under all the criteria (Accuracy, confusion matrix, recall, precision, f1-score and AUC-ROC curve).

**Serialization:**

