**Problem Statement or Requirement:**

A requirement from the Hospital, Management asked us to create a predictive model which will predict the Chronic Kidney Disease (CKD) based on the several parameters. The Client has provided the dataset of the same.

1. **Identify your problem statement**

* Stage - 1 🡪 Machine Learning
* Stage – 2 🡪 Classification

1. **Tell basic info about the dataset (Total number of rows, columns)**

The dataset has 399 rows and 25 columns

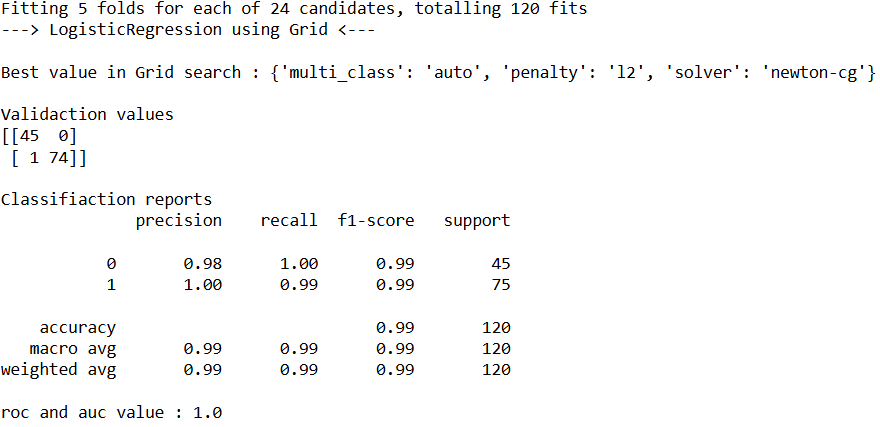
1. **Mention the pre-processing method if you’re doing any (like converting string to number – nominal data)**

In this data set we are using Nominal Pre-Processing method

Totally we have 12 categorical columns in this data set.

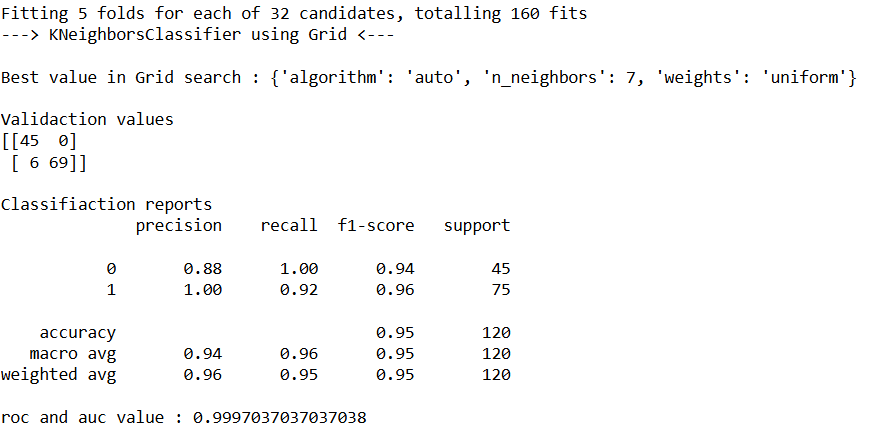
Using StandardScaler method for input test and train and convert the user input into StandardScaler method to get the correct answer.

1. **Develop a good model with good evaluation metric. You can use any machine learning algorithm; you can create many models. Finally, you have to come up with final model.**

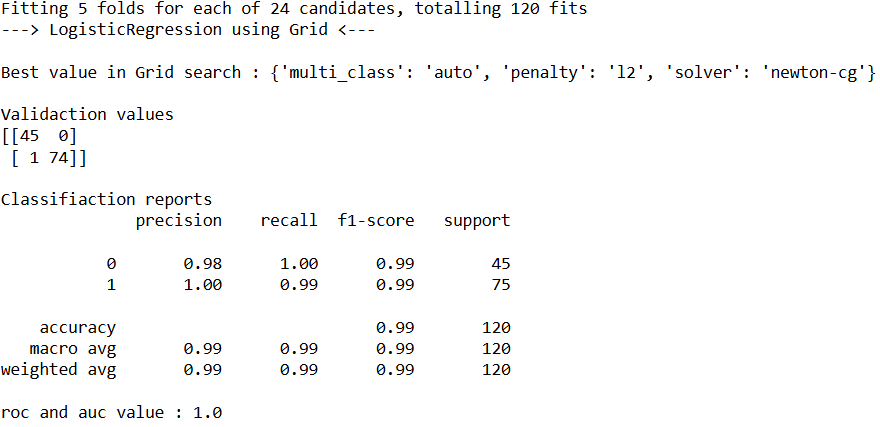


1. **All the research values of each algorithm should be documented. (You can make tabulation or screenshot of the results.)**

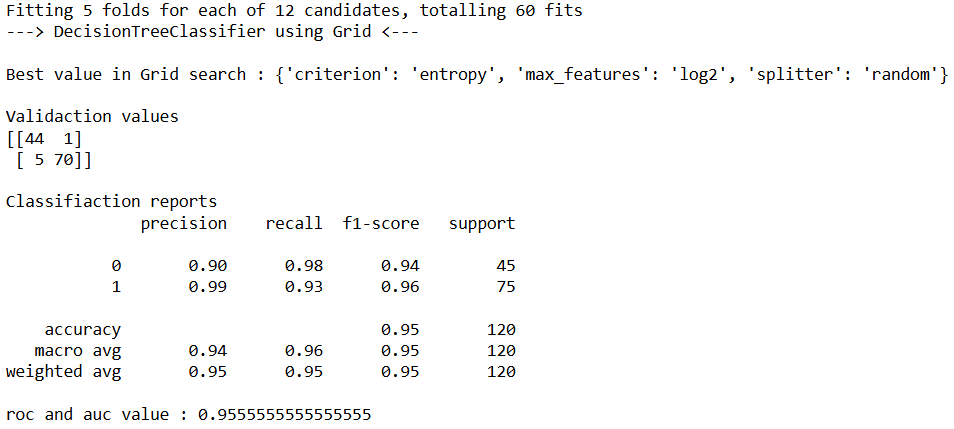
**KNN:**

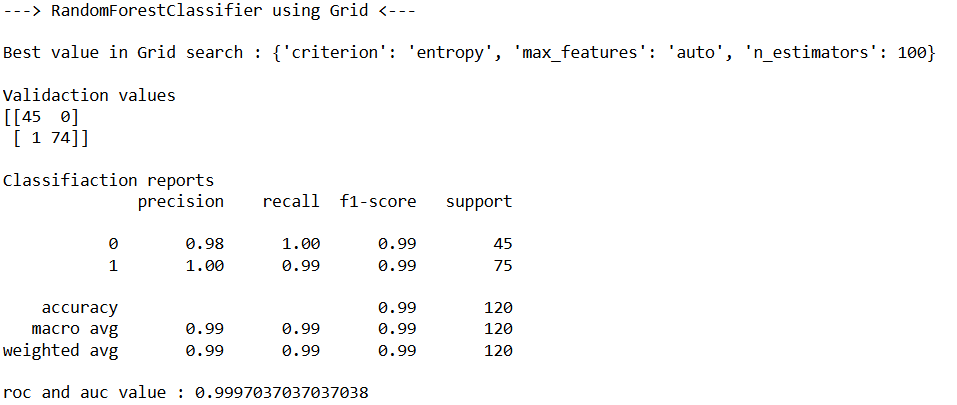


**LogisticRegression:**

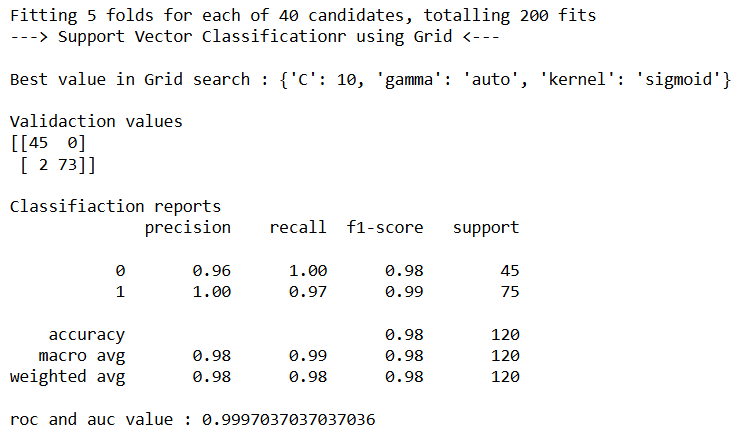


**DecisionTreeClassifier:**

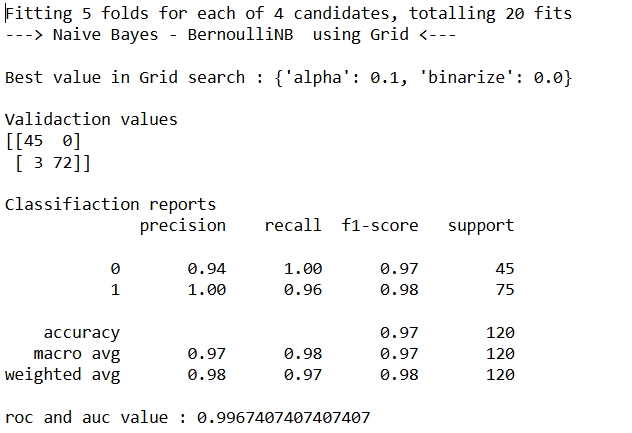


**RandomForestClassifier:  
**

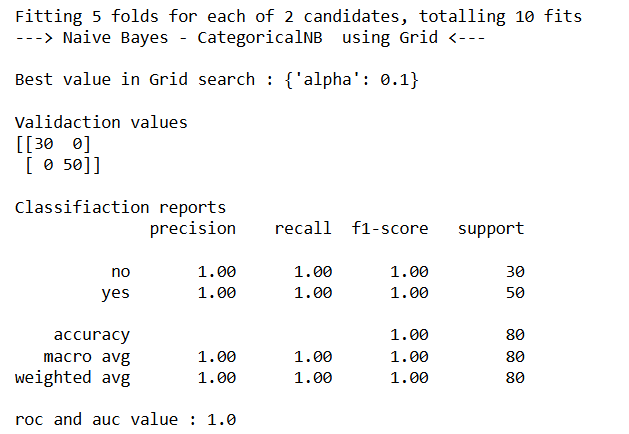
**Support Vector Classification:**

****

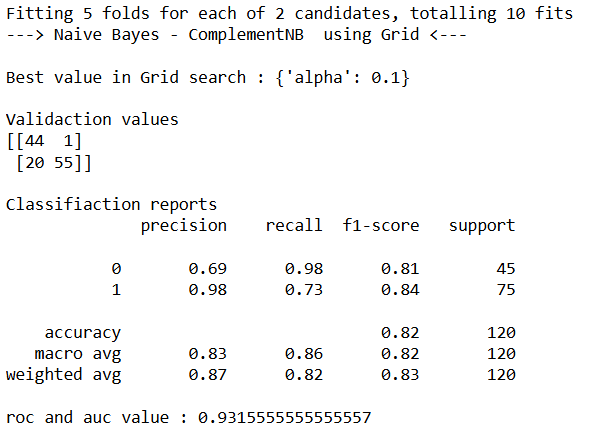
**Naive Bayes and 5 types**

**1.BernoulliNB**  
****

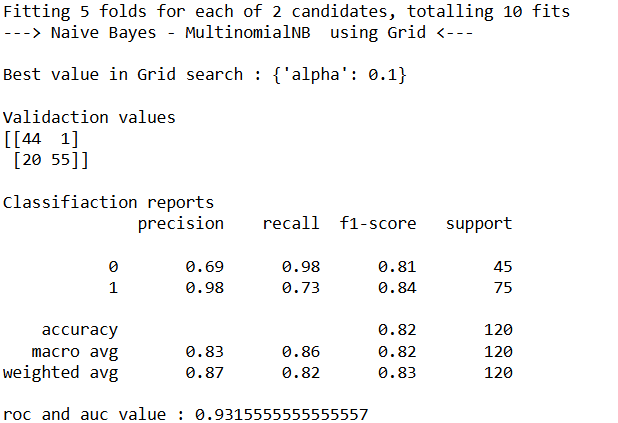
**2. CategoricalNB**



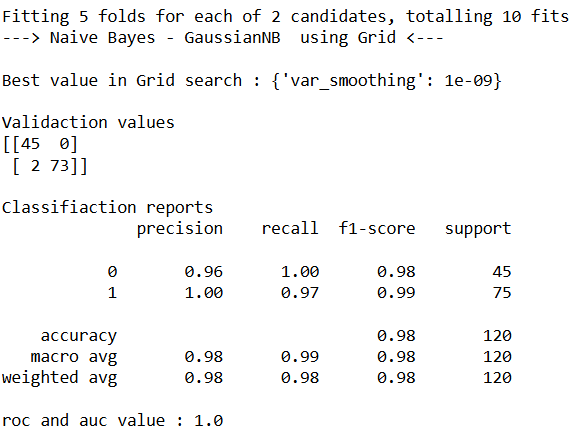
**3.ComplementNB**

****

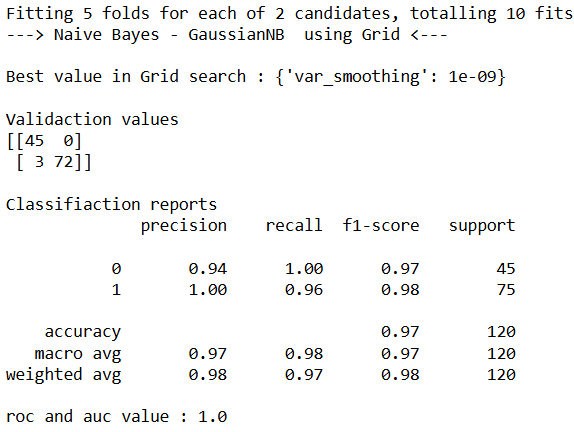
**4. MultinomialNB**

****

**5. GaussianNB 🡪 With out Standardization**

****

**With Standardization**

****

1. **Mention your final model, justify why u have chosen the same.**

**Best model is LogisticRegression**

Confusion matrix score:

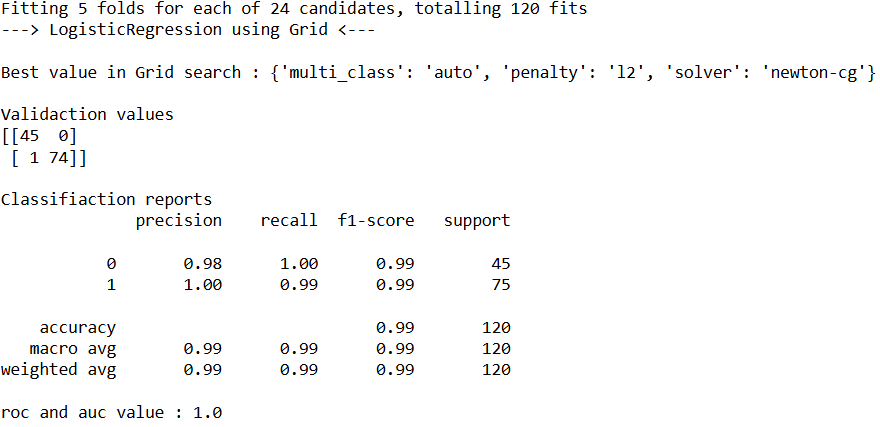
45 correctly predicted as Classification No – 0 and error1 was 0

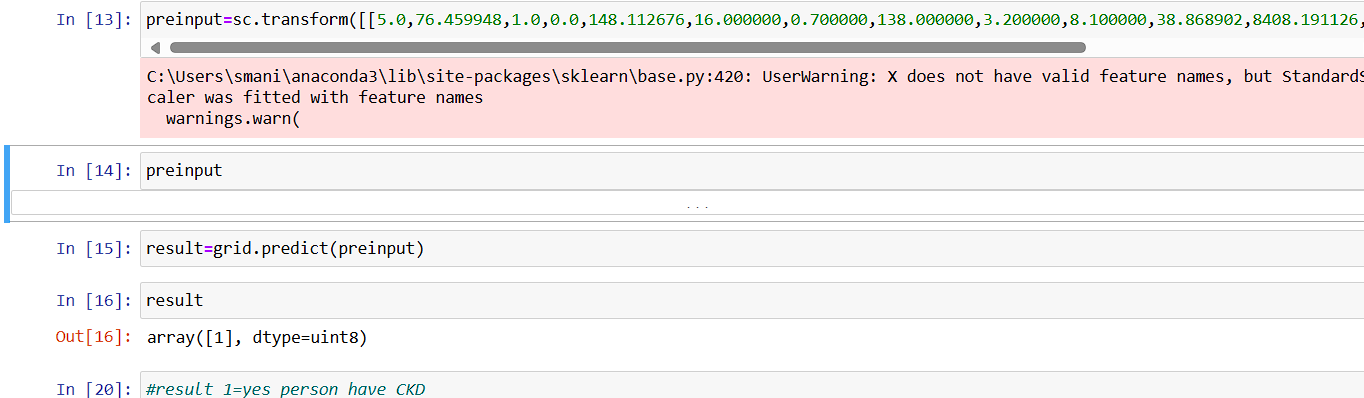
74 correctly predicted as Classification Yes – 1 and error2 was 1

Accuracy and f1\_score was 0.99 all other scores also High score.

Roc and Auc this algorithm 100% cover the full curve 1.0 score.

Result also correct from the user input.





**Notes:**

In this I’m used pd.get\_dummies(df, drop\_first=True), so **lose few category column**, so i **can’t fully reverse** it.

If we not drop the **few category column** then we use .idxmax(axis=1) to revise the Number to categorical data.