Final Val Method 1/04/2024

1. The Hospital Management Asked to create a Predictive Model, Will Predict the Chronic Kidney Disease. Based On the Several Parameters.

2. The Basic Information is,

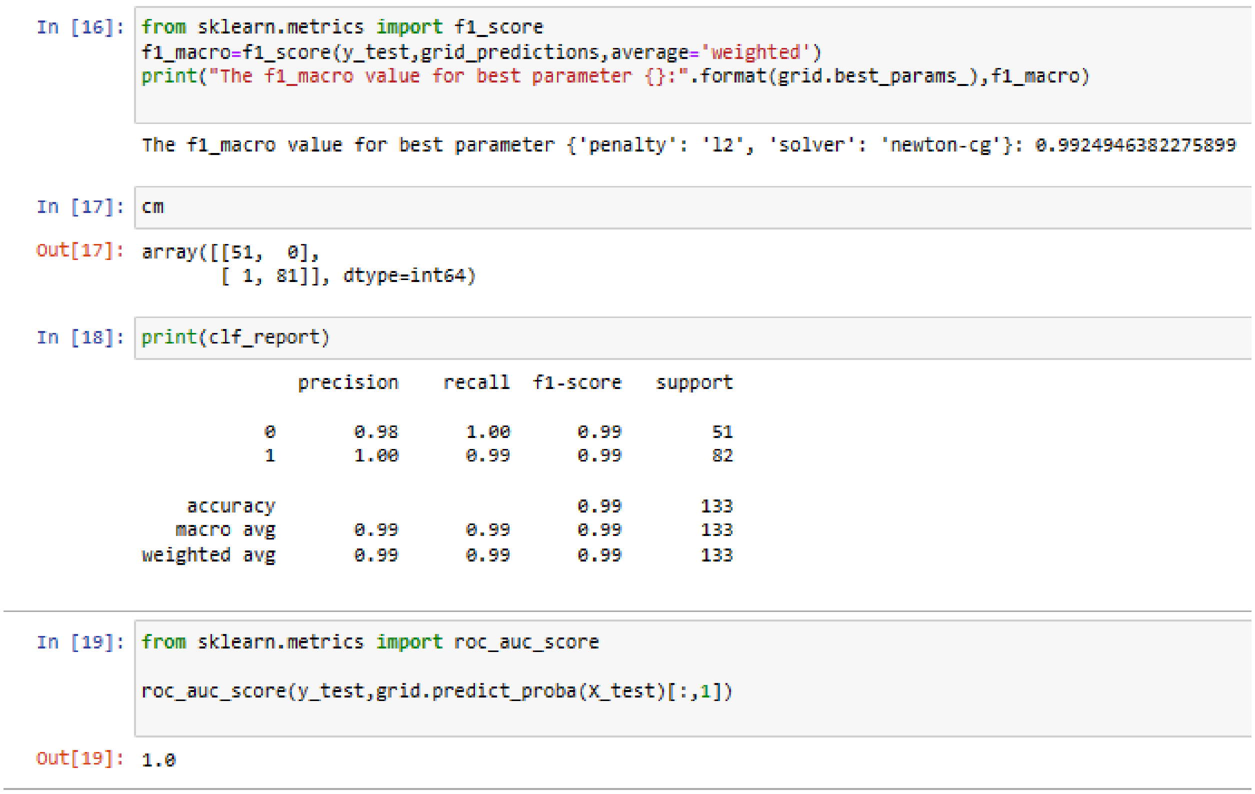
Input →Dataset, Output →Predict Chronic Kidney Disease

Total No of Rows= 399

Total No of Columns= 25

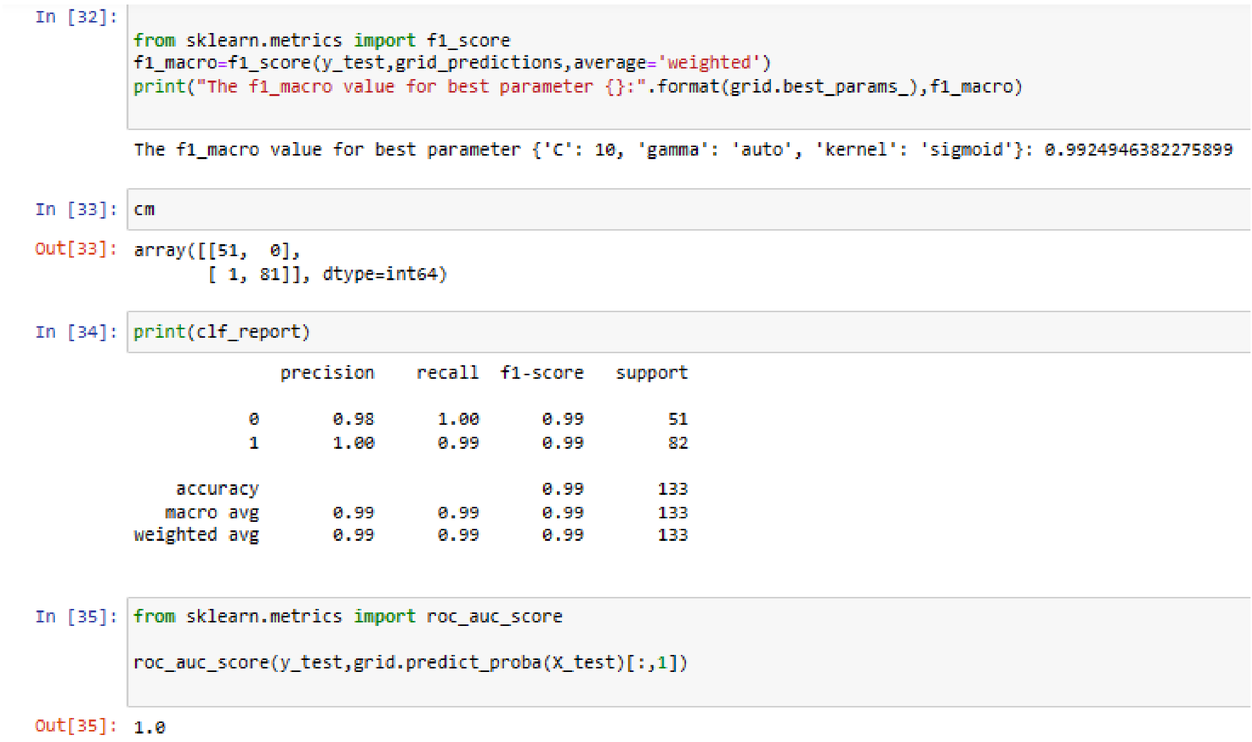
3. Here the Preprocessing Method is, to handle Categorical column using, Converting String to Number (Nominal Data→ One Hot Encoder

1.LOGISTIC GRID CLASSIFICATION ASSIGNMENT



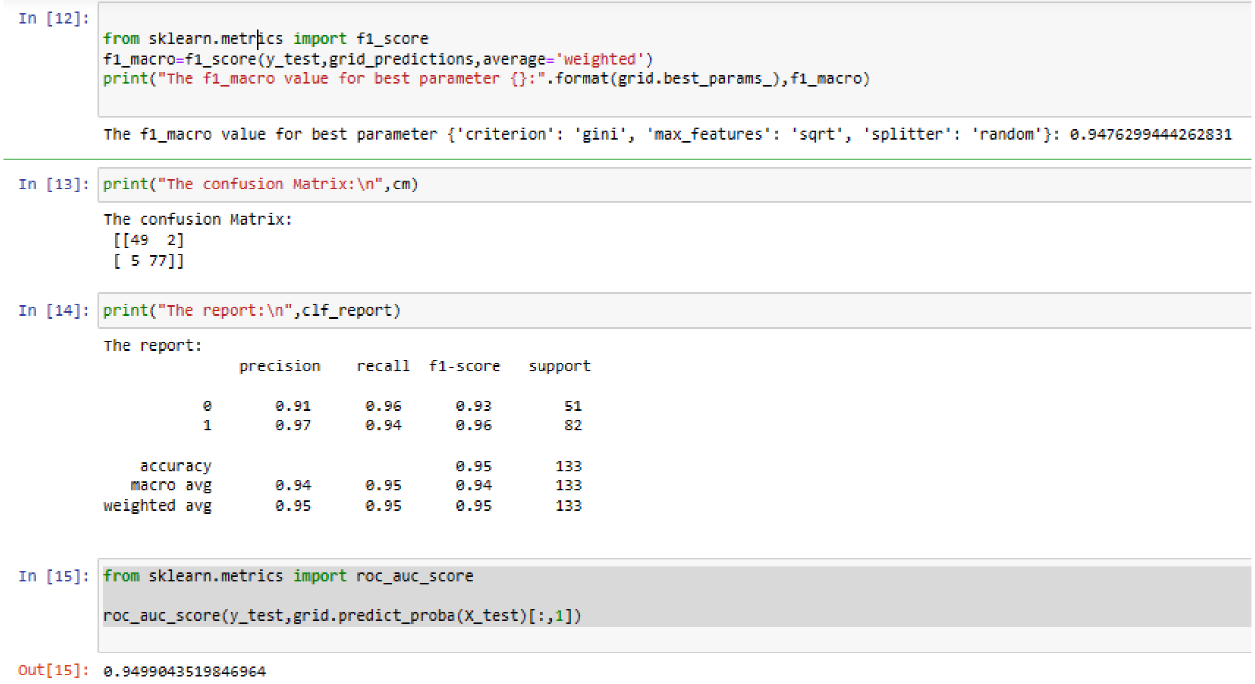
```
In [16]: from sklearn.metrics import f1_score
         f1_macro=f1_score(y_test,grid_predictions,average='weighted')
         print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)

         The f1_macro value for best parameter {'penalty': 'l2', 'solver': 'newton-cg'}: 0.9924946382275899

In [17]: cm

Out[17]: array([[51,  0],
                [ 1, 81]], dtype=int64)

In [18]: print(clf_report)

                       precision    recall  f1-score   support

                    0       0.98      1.00      0.99        51
                    1       1.00      0.99      0.99        82

             accuracy                           0.99       133
            macro avg       0.99      0.99      0.99       133
         weighted avg       0.99      0.99      0.99       133

In [19]: from sklearn.metrics import roc_auc_score

         roc_auc_score(y_test,grid.predict_proba(X_test)[:,1])

Out[19]: 1.0
```

2.SVM GRID CLASSIFICATION ASSIGNMENT



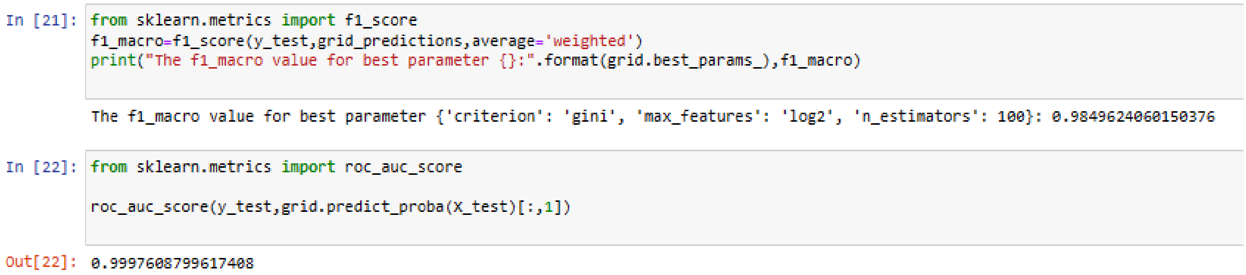
```
In [32]:  from sklearn.metrics import f1_score
          f1_macro=f1_score(y_test,grid_predictions,average='weighted')
          print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)

          The f1_macro value for best parameter {'C': 10, 'gamma': 'auto', 'kernel': 'sigmoid'}: 0.9924946382275899

In [33]:  cm

Out[33]:  array([[51,  0],
                 [ 1, 81]], dtype=int64)

In [34]:  print(clf_report)

                        precision    recall  f1-score   support

                     0       0.98      1.00      0.99        51
                     1       1.00      0.99      0.99        82

              accuracy                           0.99       133
             macro avg       0.99      0.99      0.99       133
          weighted avg       0.99      0.99      0.99       133


In [35]:  from sklearn.metrics import roc_auc_score

          roc_auc_score(y_test,grid.predict_proba(X_test)[:,1])

Out[35]:  1.0
```

3.DC GRID CLASSIFICATION ASSIGNMENT

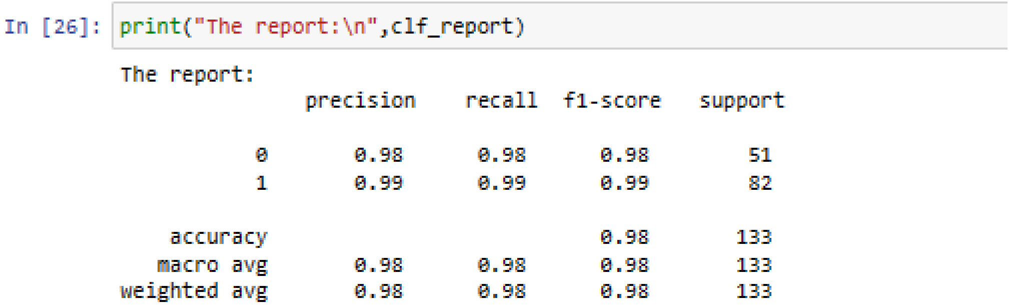


```
In [12]:  from sklearn.metrics import f1_score
          f1_macro=f1_score(y_test,grid_predictions,average='weighted')
          print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)

          The f1_macro value for best parameter {'criterion': 'gini', 'max_features': 'sqrt', 'splitter': 'random'}: 0.9476299444262831

In [13]:  print("The confusion Matrix:\n",cm)

          The confusion Matrix:
           [[49  2]
           [ 5 77]]

In [14]:  print("The report:\n",clf_report)

          The report:
                        precision    recall  f1-score   support

                     0       0.91      0.96      0.93        51
                     1       0.97      0.94      0.96        82

              accuracy                           0.95       133
             macro avg       0.94      0.95      0.94       133
          weighted avg       0.95      0.95      0.95       133


In [15]:  from sklearn.metrics import roc_auc_score

          roc_auc_score(y_test,grid.predict_proba(X_test)[:,1])

Out[15]:  0.9499043519846964
```

4.RF GRID CLASSIFICATION ASSIGNMENT



```
In [21]: from sklearn.metrics import f1_score
         f1_macro=f1_score(y_test,grid_predictions,average='weighted')
         print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)

         The f1_macro value for best parameter {'criterion': 'gini', 'max_features': 'log2', 'n_estimators': 100}: 0.9849624060150376

In [22]: from sklearn.metrics import roc_auc_score

         roc_auc_score(y_test,grid.predict_proba(X_test)[:,1])

Out[22]: 0.9997608799617408
```



```
In [26]: print("The report:\n",clf_report)

The report:
               precision    recall  f1-score   support

           0       0.98      0.98      0.98        51
           1       0.99      0.99      0.99        82

    accuracy                           0.98       133
   macro avg       0.98      0.98      0.98       133
weighted avg       0.98      0.98      0.98       133
```

The Result of RF GRID CLASSIFICATION Algorithm is good accuracy value 0.98 compared to all Algorithm.