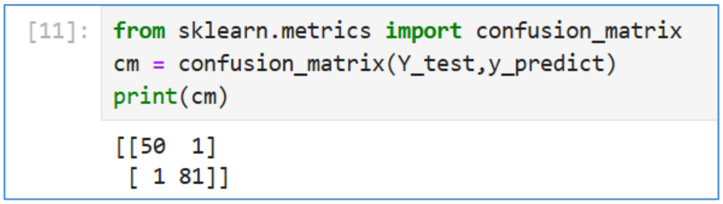
CKD Assignment – RF

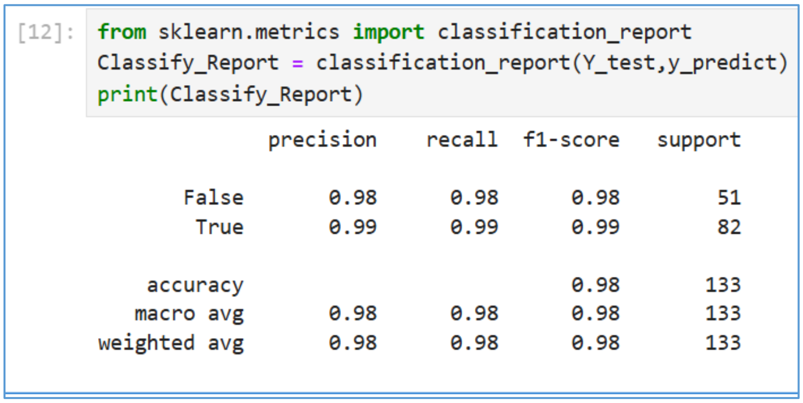
Confusion Matrix:



```
[11]: from sklearn.metrics import confusion_matrix
      cm = confusion_matrix(Y_test,y_predict)
      print(cm)

      [[50  1]
       [ 1 81]]
```

Classification Report:



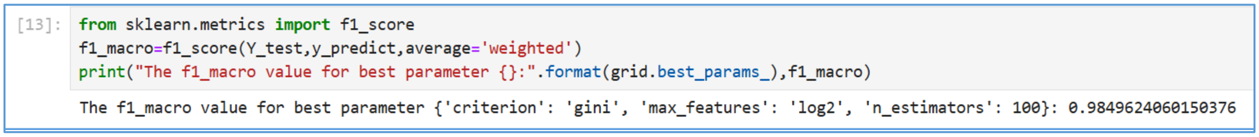
```
[12]: from sklearn.metrics import classification_report
      Classify_Report = classification_report(Y_test,y_predict)
      print(Classify_Report)

                    precision    recall  f1-score   support

             False       0.98      0.98      0.98        51
              True       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
```

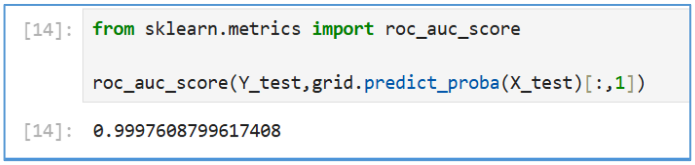
FI Macro:



```
[13]: from sklearn.metrics import f1_score
      f1_macro=f1_score(Y_test,y_predict,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
```

ROC AUC Score:

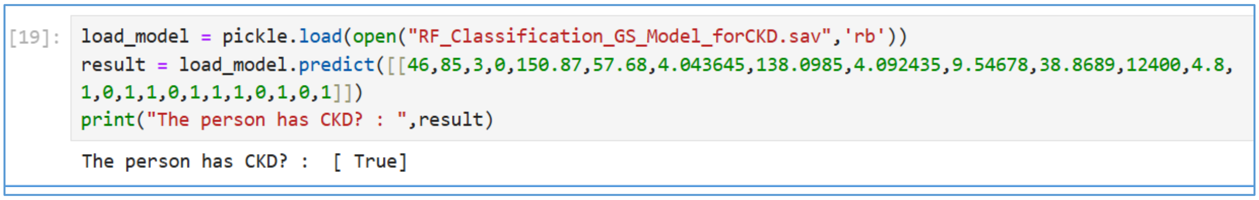


```
[14]: from sklearn.metrics import roc_auc_score

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

[14]: 0.9997608799617408
```

CKD Result:



```
[19]: load_model = pickle.load(open("RF_Classification_GS_Model_forCKD.sav",'rb'))
      result = load_model.predict([[46,85,3,0,150.87,57.68,4.043645,138.0985,4.092435,9.54678,38.8689,12400,4.8,
      1,0,1,1,0,1,1,1,0,1,0,1]])
      print("The person has CKD? : ",result)

      The person has CKD? :  [ True]
```