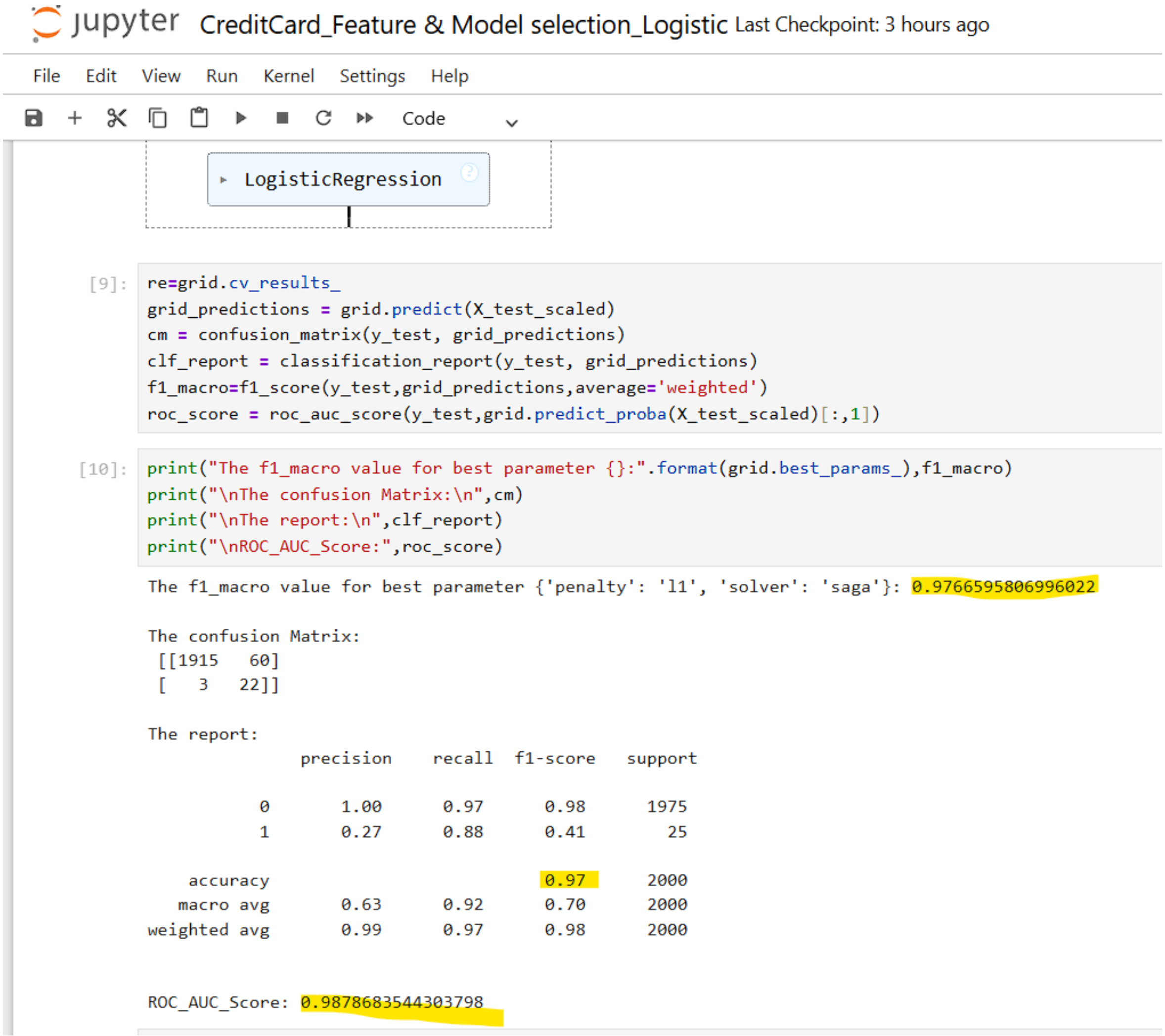
Credit Card Fraud Detection

Credit card fraud detection’s end goal is to classify whether the transaction is fraud or legit. Based on the requirement and dataset which is comes under supervised learning - classification. I have listed out the model accuracy of various classification algorithms.

Logistic Regression:

Model Accuracy is 97 percentage and roc score is 0.98.


jupyter
CreditCard_Feature & Model selection_Logistic
Last Checkpoint: 3 hours ago
File
Edit
View
Run
Kernel
Settings
Help
Code
LogisticRegression
[9]:
[10]:

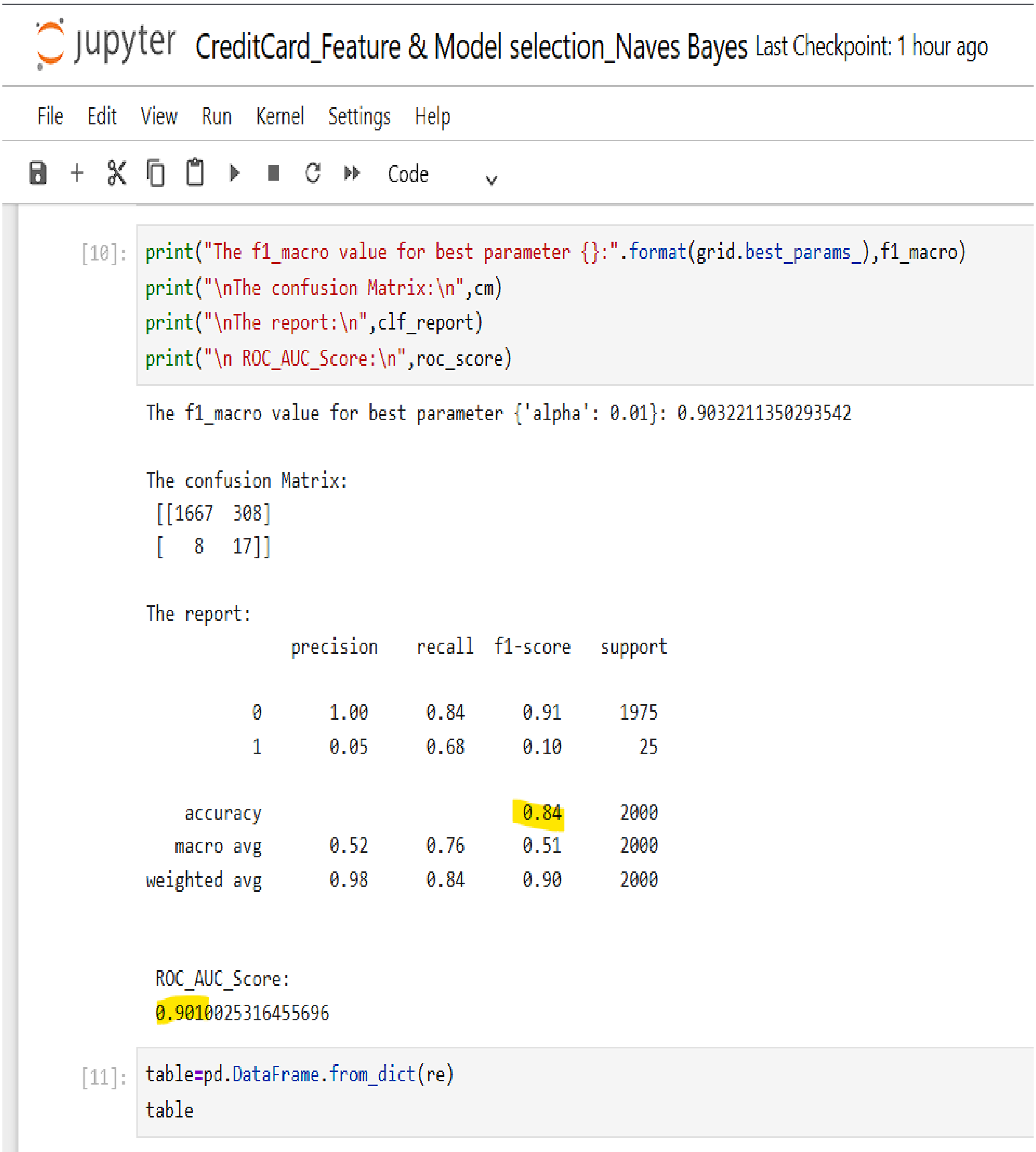
```
re=grid.cv_results_
grid_predictions = grid.predict(X_test_scaled)
cm = confusion_matrix(y_test, grid_predictions)
clf_report = classification_report(y_test, grid_predictions)
f1_macro=f1_score(y_test,grid_predictions,average='weighted')
roc_score = roc_auc_score(y_test,grid.predict_proba(X_test_scaled)[:,1])
```

```
print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
print("\nThe confusion Matrix:\n",cm)
print("\nThe report:\n",clf_report)
print("\nROC_AUC_Score:",roc_score)
```

```
The f1_macro value for best parameter {'penalty': 'l1', 'solver': 'saga'}: 0.9766595806996022

The confusion Matrix:
 [[1915   60]
 [   3   22]]

The report:
               precision    recall  f1-score   support

           0       1.00      0.97      0.98      1975
           1       0.27      0.88      0.41        25

    accuracy                           0.97      2000
   macro avg       0.63      0.92      0.70      2000
weighted avg       0.99      0.97      0.98      2000


ROC_AUC_Score: 0.9878683544303798
```

Naves Bayes:

Model Accuracy is 84 percentage and roc score is 0.90.


jupyter
CreditCard_Feature & Model selection_Naves Bayes Last Checkpoint: 1 hour ago
File
Edit
View
Run
Kernel
Settings
Help
Code

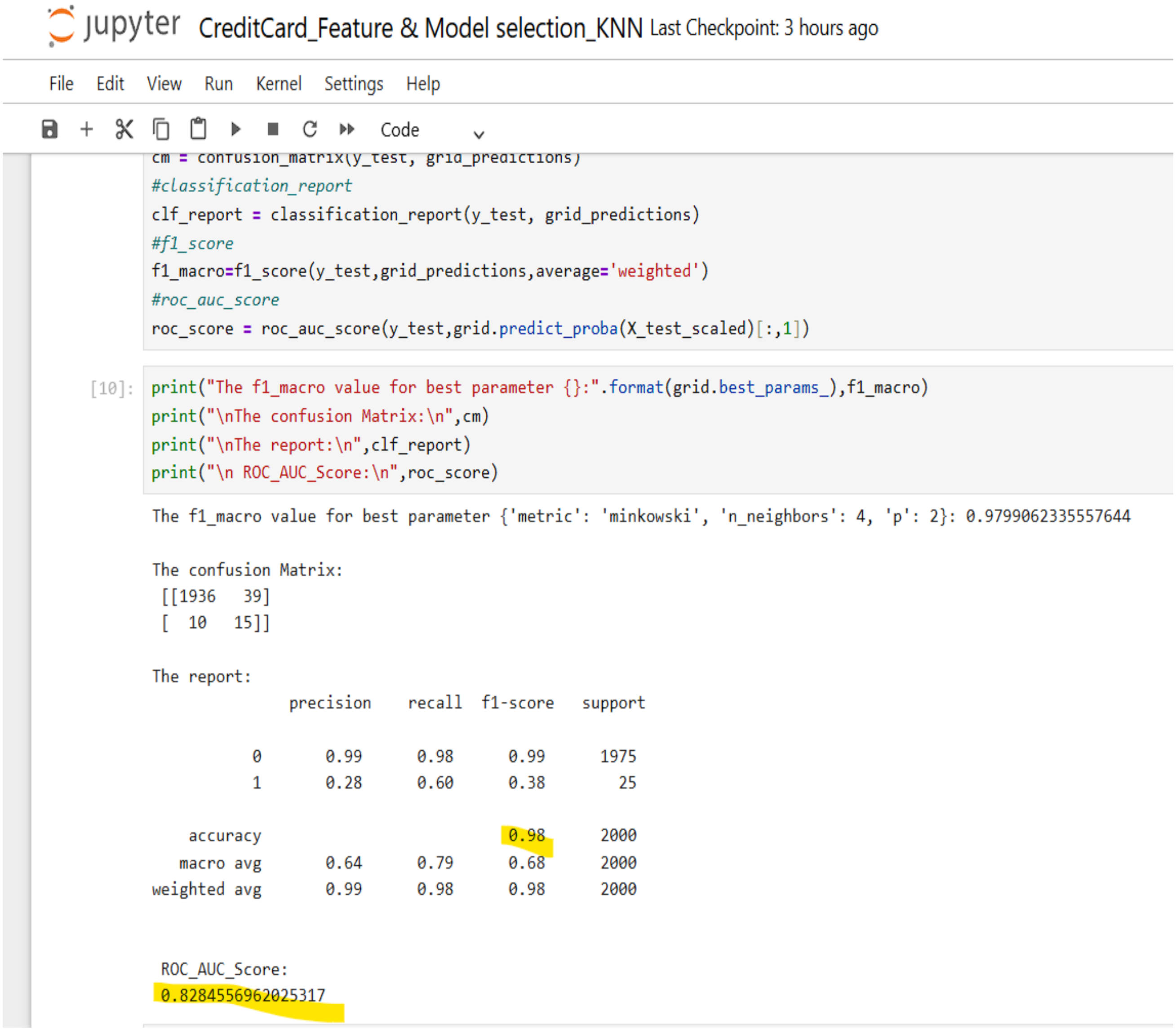
```
[10]: print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
      print("\nThe confusion Matrix:\n",cm)
      print("\nThe report:\n",clf_report)
      print("\n ROC_AUC_Score:\n",roc_score)
```

```
The f1_macro value for best parameter {'alpha': 0.01}: 0.9032211350293542

The confusion Matrix:
 [[1667  308]
 [   8   17]]

The report:
               precision    recall  f1-score   support

           0       1.00      0.84      0.91      1975
           1       0.05      0.68      0.10        25

    accuracy                           0.84      2000
   macro avg       0.52      0.76      0.51      2000
weighted avg       0.98      0.84      0.90      2000


 ROC_AUC_Score:
 0.9010025316455696
```

```
[11]: table=pd.DataFrame.from_dict(re)
      table
```

KNN:

Model Accuracy is 98 percentage and roc score is 0.82.


jupyter
CreditCard_Feature & Model selection_KNN Last Checkpoint: 3 hours ago
File
Edit
View
Run
Kernel
Settings
Help
Code
[10]:

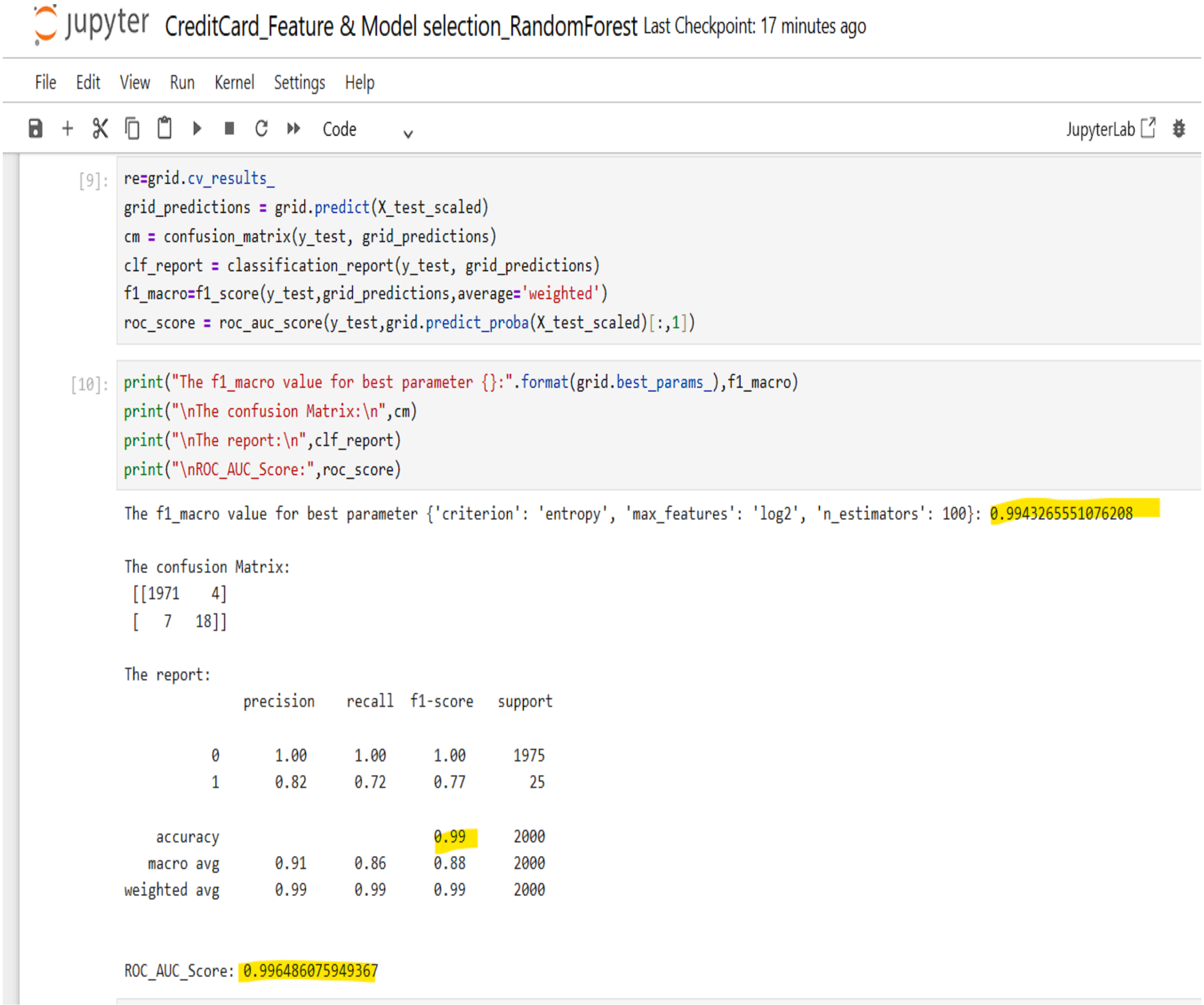
```
cm = confusion_matrix(y_test, grid_predictions)
#classification_report
clf_report = classification_report(y_test, grid_predictions)
#f1_score
f1_macro=f1_score(y_test,grid_predictions,average='weighted')
#roc_auc_score
roc_score = roc_auc_score(y_test,grid.predict_proba(X_test_scaled)[:,1])
```

```
print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
print("\nThe confusion Matrix:\n",cm)
print("\nThe report:\n",clf_report)
print("\n ROC_AUC_Score:\n",roc_score)
```

```
The f1_macro value for best parameter {'metric': 'minkowski', 'n_neighbors': 4, 'p': 2}: 0.9799062335557644

The confusion Matrix:
 [[1936   39]
 [  10   15]]

The report:
               precision    recall  f1-score   support

           0       0.99      0.98      0.99      1975
           1       0.28      0.60      0.38        25

    accuracy                           0.98      2000
   macro avg       0.64      0.79      0.68      2000
weighted avg       0.99      0.98      0.98      2000


 ROC_AUC_Score:
 0.8284556962025317
```

Random Forest:

Model Accuracy is 99 percentage and roc score is 0.99.


jupyter
CreditCard_Feature & Model selection_RandomForest Last Checkpoint: 17 minutes ago
File
Edit
View
Run
Kernel
Settings
Help
Code
JupyterLab

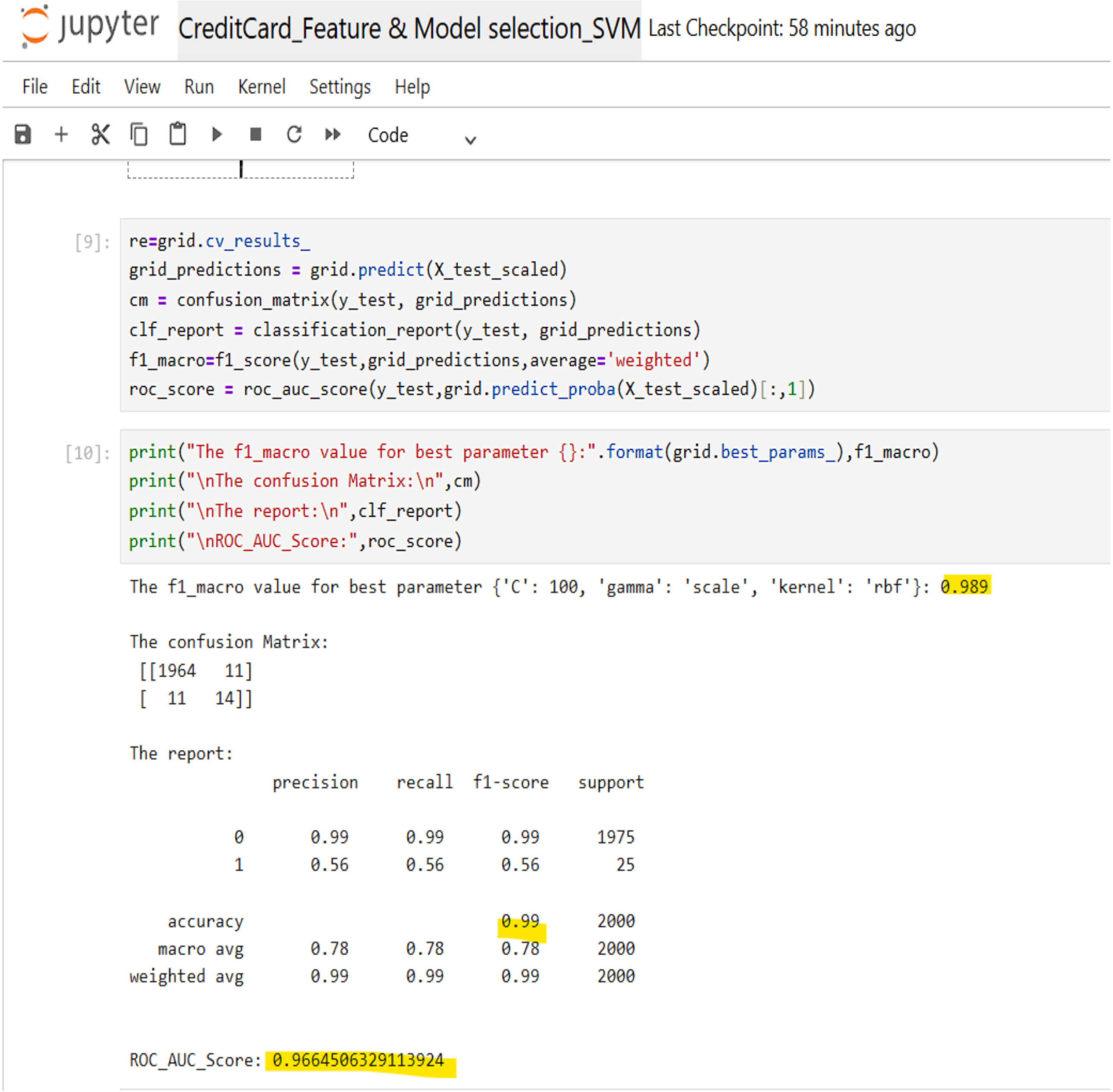
```
[9]: re=grid.cv_results_
     grid_predictions = grid.predict(X_test_scaled)
     cm = confusion_matrix(y_test, grid_predictions)
     clf_report = classification_report(y_test, grid_predictions)
     f1_macro=f1_score(y_test,grid_predictions,average='weighted')
     roc_score = roc_auc_score(y_test,grid.predict_proba(X_test_scaled)[:,1])
```

```
[10]: print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
      print("\nThe confusion Matrix:\n",cm)
      print("\nThe report:\n",clf_report)
      print("\nROC_AUC_Score:",roc_score)
```

```
The f1_macro value for best parameter {'criterion': 'entropy', 'max_features': 'log2', 'n_estimators': 100}: 0.9943265551076208

The confusion Matrix:
 [[1971    4]
 [   7   18]]

The report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      1975
           1       0.82      0.72      0.77        25

    accuracy                           0.99      2000
   macro avg       0.91      0.86      0.88      2000
weighted avg       0.99      0.99      0.99      2000


ROC_AUC_Score: 0.996486075949367
```

SVM:

Model Accuracy is 97 percentage and roc score is 0.98.



jupyter
CreditCard_Feature & Model selection_SVM Last Checkpoint: 58 minutes ago
File Edit View Run Kernel Settings Help
Code

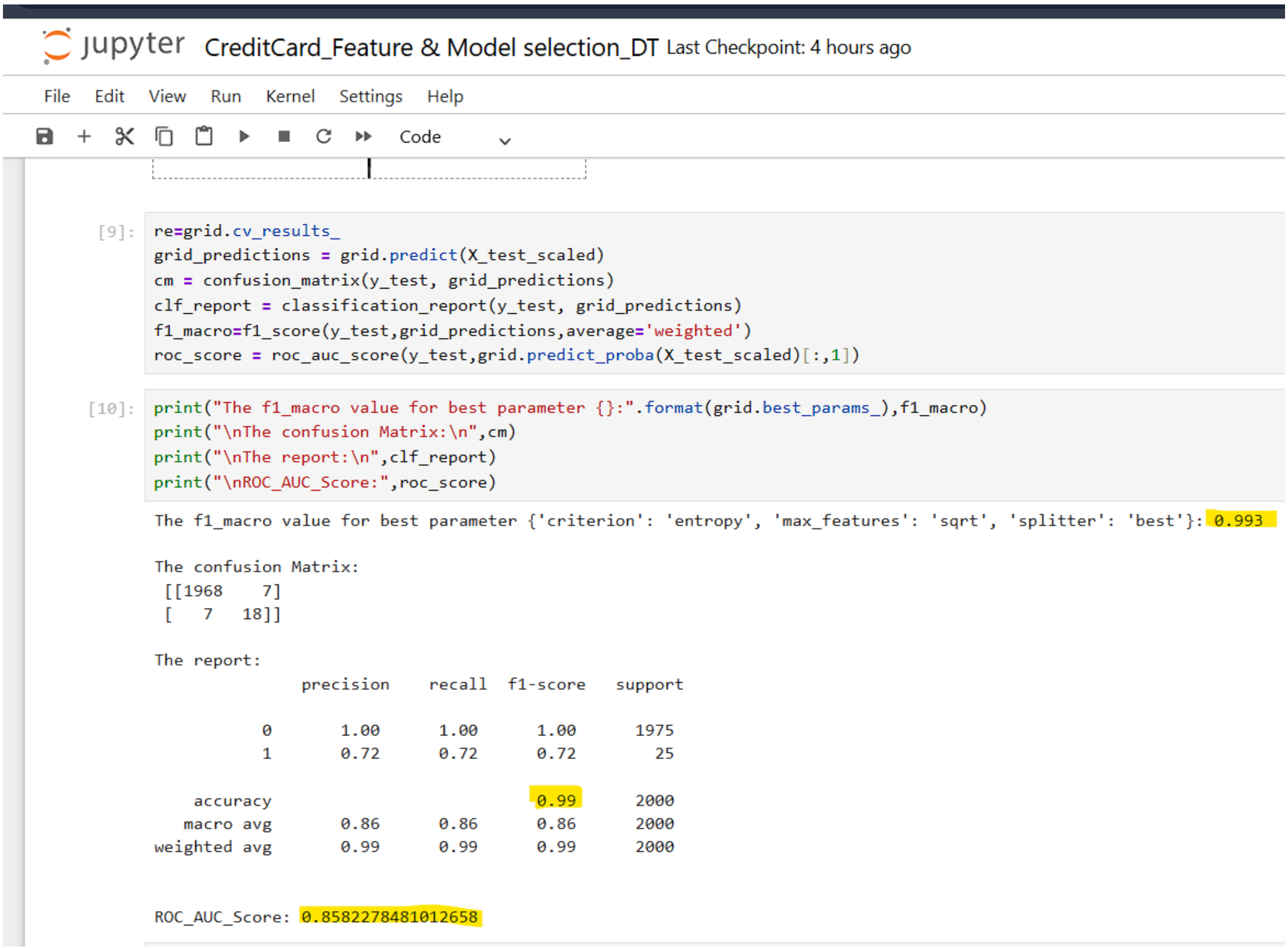
```
[9]: re=grid.cv_results_
     grid_predictions = grid.predict(X_test_scaled)
     cm = confusion_matrix(y_test, grid_predictions)
     clf_report = classification_report(y_test, grid_predictions)
     f1_macro=f1_score(y_test,grid_predictions,average='weighted')
     roc_score = roc_auc_score(y_test,grid.predict_proba(X_test_scaled)[:,1])
```

```
[10]: print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
      print("\nThe confusion Matrix:\n",cm)
      print("\nThe report:\n",clf_report)
      print("\nROC_AUC_Score:",roc_score)
```

```
The f1_macro value for best parameter {'C': 100, 'gamma': 'scale', 'kernel': 'rbf'}: 0.989

The confusion Matrix:
 [[1964   11]
 [  11   14]]

The report:
               precision    recall  f1-score   support

           0       0.99      0.99      0.99      1975
           1       0.56      0.56      0.56        25

    accuracy                           0.99      2000
   macro avg       0.78      0.78      0.78      2000
weighted avg       0.99      0.99      0.99      2000


ROC_AUC_Score: 0.9664506329113924
```

Decision Tree:

Model Accuracy is 99 percentage and roc score is 0.85.



jupyter
CreditCard_Feature & Model selection_DT Last Checkpoint: 4 hours ago
File
Edit
View
Run
Kernel
Settings
Help
Code

```
[9]: re=grid.cv_results_
     grid_predictions = grid.predict(X_test_scaled)
     cm = confusion_matrix(y_test, grid_predictions)
     clf_report = classification_report(y_test, grid_predictions)
     f1_macro=f1_score(y_test,grid_predictions,average='weighted')
     roc_score = roc_auc_score(y_test,grid.predict_proba(X_test_scaled)[:,1])
```

```
[10]: print("The f1_macro value for best parameter {}:".format(grid.best_params_),f1_macro)
      print("\nThe confusion Matrix:\n",cm)
      print("\nThe report:\n",clf_report)
      print("\nROC_AUC_Score:",roc_score)
```

```
The f1_macro value for best parameter {'criterion': 'entropy', 'max_features': 'sqrt', 'splitter': 'best'}: 0.993

The confusion Matrix:
 [[1968    7]
 [   7   18]]

The report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      1975
           1       0.72      0.72      0.72        25

    accuracy                           0.99      2000
   macro avg       0.86      0.86      0.86      2000
weighted avg       0.99      0.99      0.99      2000


ROC_AUC_Score: 0.8582278481012658
```

Conclusion:

Based on Accuracy and ROC_AUC_Score, Random forest is consider as best model. Because it has 99 percenatge of accuracy and 0.99 of roc_auc_score compartively higher than other models.