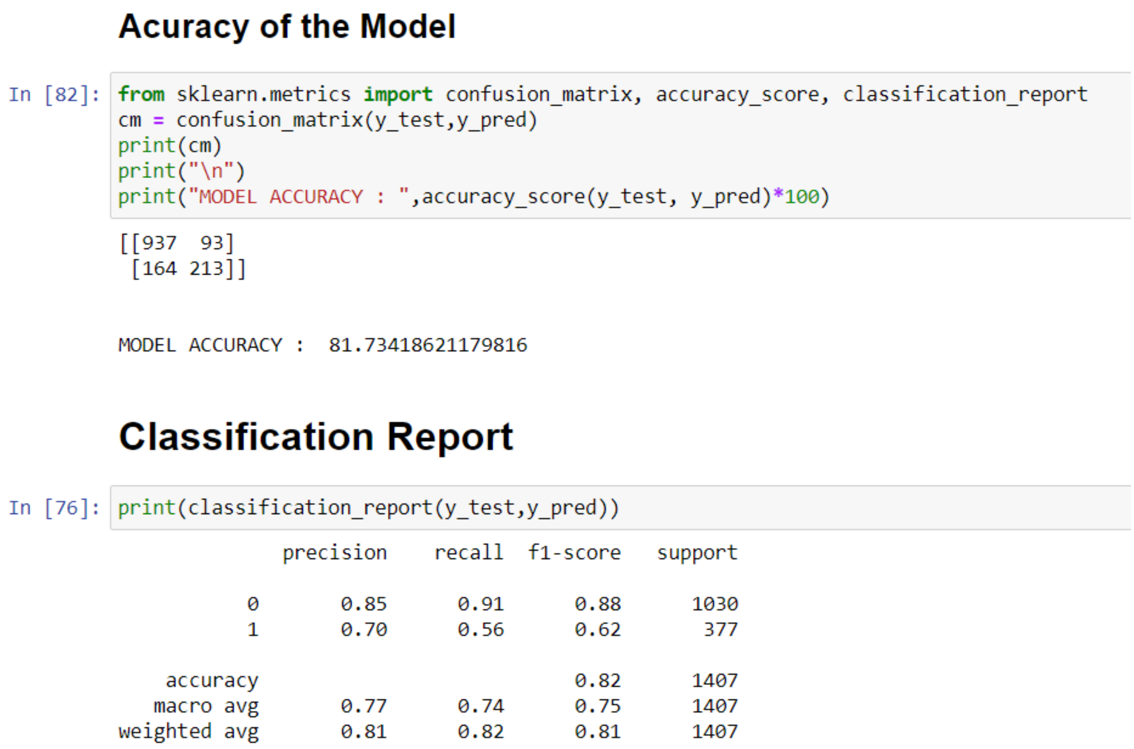
Model Accuracy:

Model Used	**Accuracy of Prediction**
Logistic Regression	81.73%
K-Nearest Neighbors	80.62%
Support Vector Machine	80.59%
Decision Tree Classification	74.30%
Naive Bayes classifier	69.83%
Random Forrest Classifiers	80.34%
Perceptron	80.26%
Kernel SVM	80.83%
AdaBoost Classifier	80.83%
Stochastic Gradient Descent Classifier	80.26%

1. Logistic Regression



Acuracy of the Model

```
In [82]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
         cm = confusion_matrix(y_test,y_pred)
         print(cm)
         print("\n")
         print("MODEL ACCURACY : ",accuracy_score(y_test, y_pred)*100)
```

```
[[937  93]
 [164 213]]


MODEL ACCURACY :  81.73418621179816
```

Classification Report

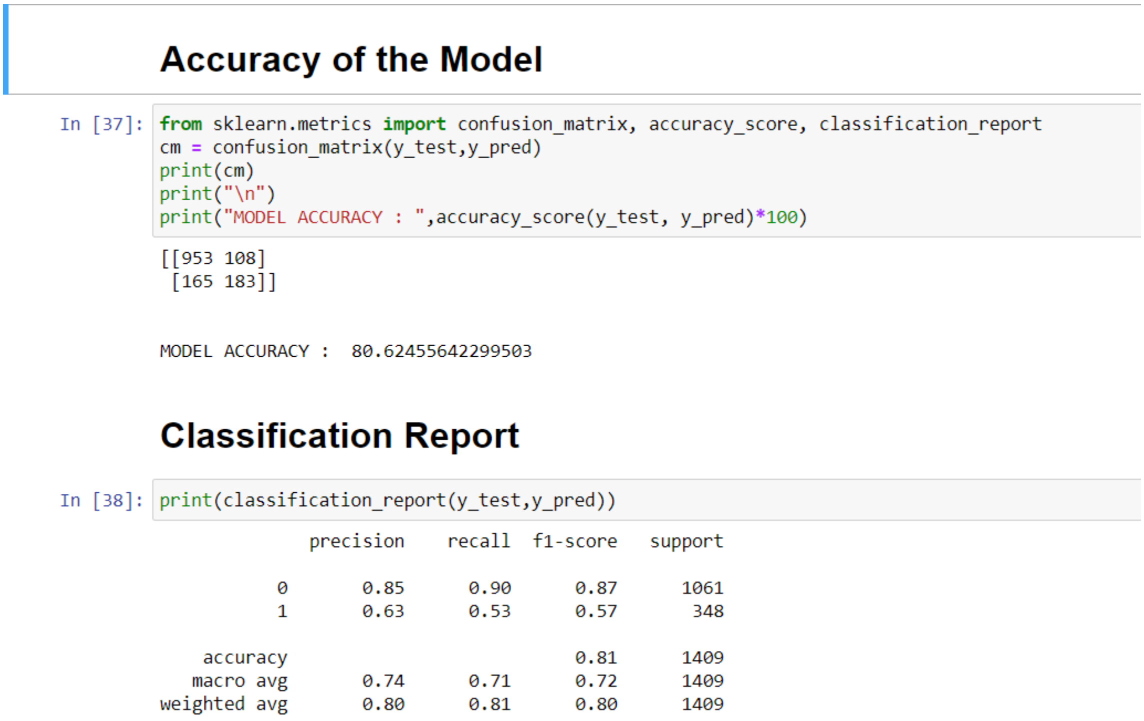
```
In [76]: print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.85      0.91      0.88      1030
           1       0.70      0.56      0.62       377

    accuracy                           0.82      1407
   macro avg       0.77      0.74      0.75      1407
weighted avg       0.81      0.82      0.81      1407
```

2. K-Nearest Neighbors



Accuracy of the Model

```
In [37]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
         cm = confusion_matrix(y_test,y_pred)
         print(cm)
         print("\n")
         print("MODEL ACCURACY : ",accuracy_score(y_test, y_pred)*100)
```

```
[[953 108]
 [165 183]]


MODEL ACCURACY :  80.62455642299503
```

Classification Report

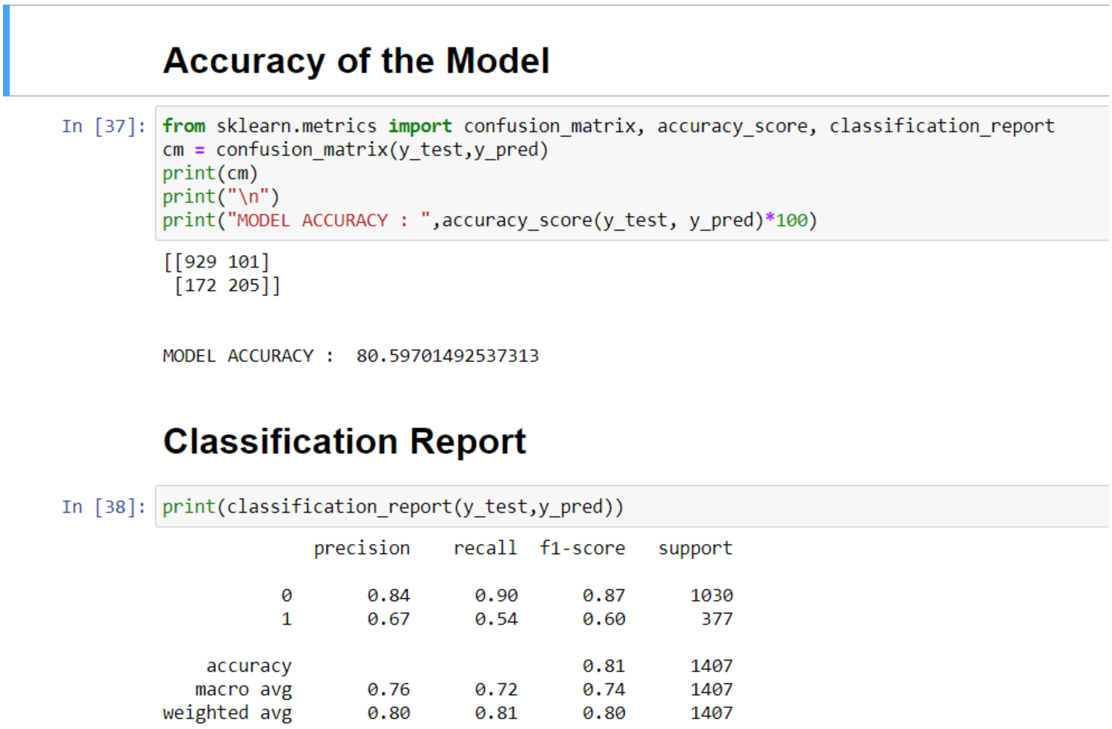
```
In [38]: print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.85      0.90      0.87      1061
           1       0.63      0.53      0.57       348

    accuracy                           0.81      1409
   macro avg       0.74      0.71      0.72      1409
weighted avg       0.80      0.81      0.80      1409
```

3. Support Vector Machine



Accuracy of the Model

```
In [37]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
         cm = confusion_matrix(y_test,y_pred)
         print(cm)
         print("\n")
         print("MODEL ACCURACY : ",accuracy_score(y_test, y_pred)*100)
```

```
[[929 101]
 [172 205]]


MODEL ACCURACY :  80.59701492537313
```

Classification Report

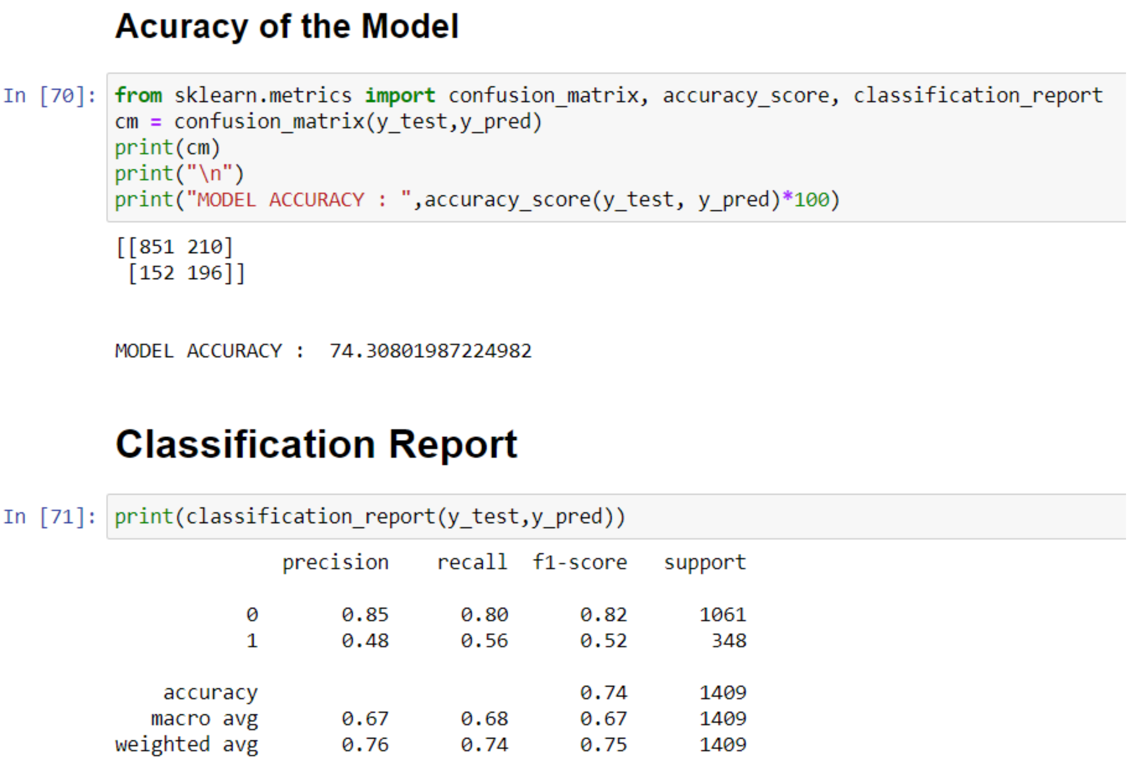
```
In [38]: print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.84      0.90      0.87      1030
           1       0.67      0.54      0.60       377

    accuracy                           0.81      1407
   macro avg       0.76      0.72      0.74      1407
weighted avg       0.80      0.81      0.80      1407
```

4. Decision Tree Classification



Acuracy of the Model

```
In [70]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
         cm = confusion_matrix(y_test,y_pred)
         print(cm)
         print("\n")
         print("MODEL ACCURACY : ",accuracy_score(y_test, y_pred)*100)
```

```
[[851 210]
 [152 196]]


MODEL ACCURACY :  74.30801987224982
```

Classification Report

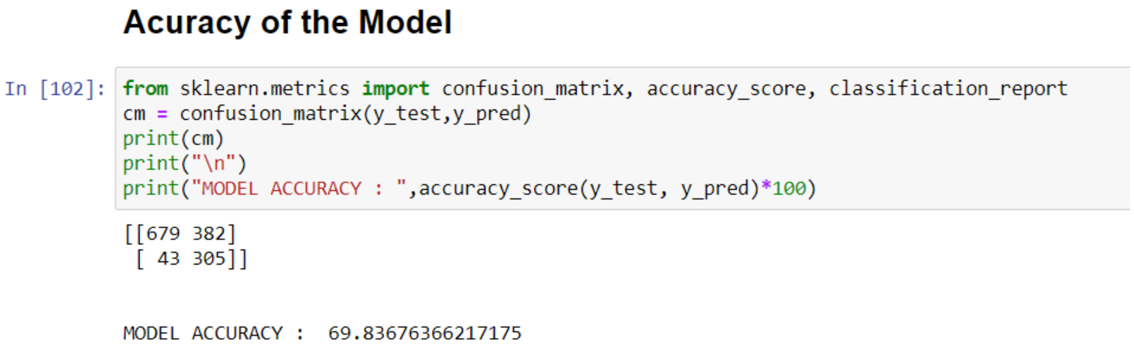
```
In [71]: print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.85      0.80      0.82      1061
           1       0.48      0.56      0.52       348

    accuracy                           0.74      1409
   macro avg       0.67      0.68      0.67      1409
weighted avg       0.76      0.74      0.75      1409
```

5. Naive Bayes classifier

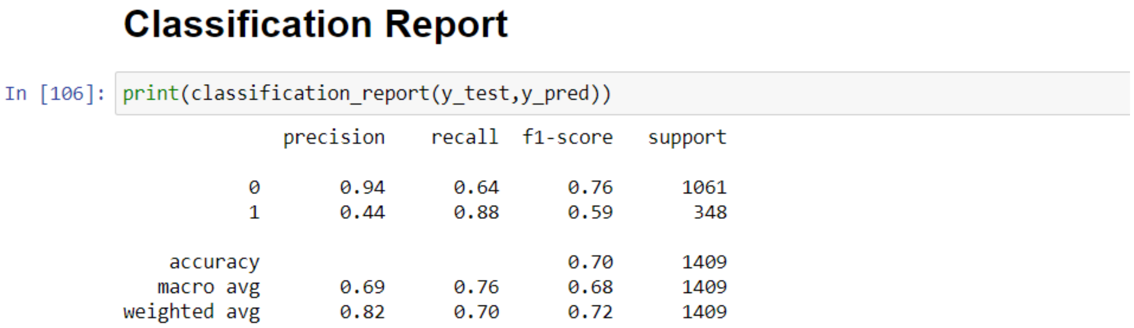


Acuracy of the Model

```
In [102]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
          cm = confusion_matrix(y_test,y_pred)
          print(cm)
          print("\n")
          print("MODEL ACCURACY : ",accuracy_score(y_test, y_pred)*100)

          [[679 382]
           [ 43 305]]


          MODEL ACCURACY :  69.83676366217175
```



Classification Report

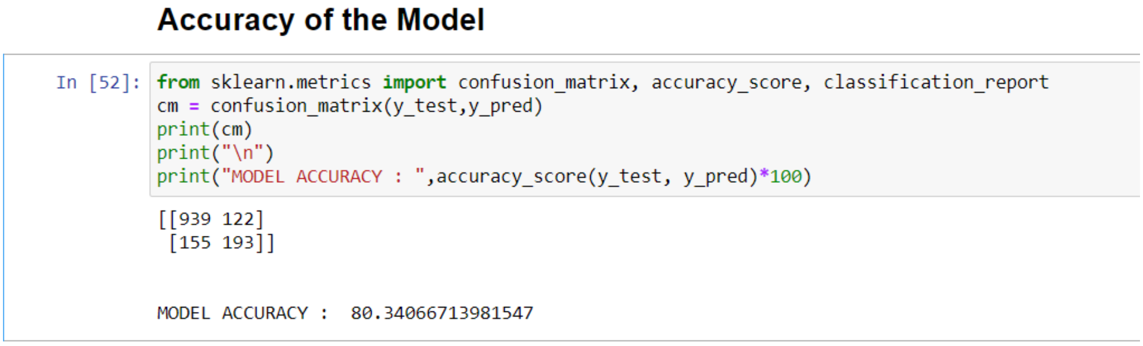
```
In [106]: print(classification_report(y_test,y_pred))

                        precision    recall  f1-score   support

                     0       0.94      0.64      0.76      1061
                     1       0.44      0.88      0.59       348

              accuracy                           0.70      1409
             macro avg       0.69      0.76      0.68      1409
          weighted avg       0.82      0.70      0.72      1409
```

6. Random Forrest Classifiers

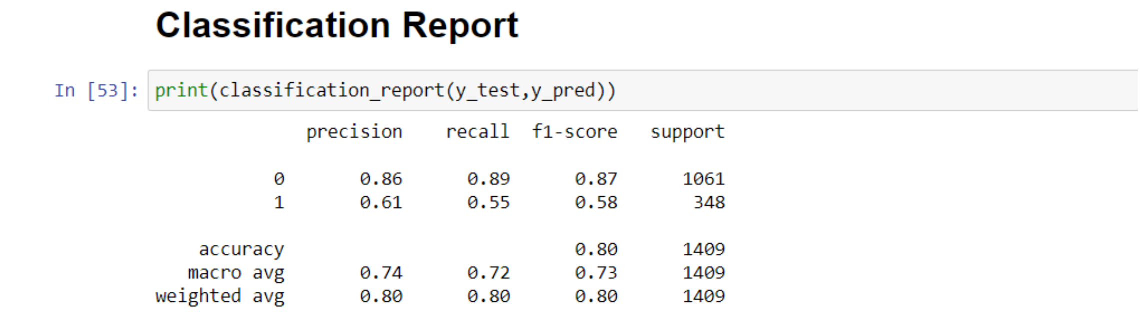


Accuracy of the Model

```
In [52]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
         cm = confusion_matrix(y_test,y_pred)
         print(cm)
         print("\n")
         print("MODEL ACCURACY : ",accuracy_score(y_test, y_pred)*100)

         [[939 122]
          [155 193]]


         MODEL ACCURACY :  80.34066713981547
```



Classification Report

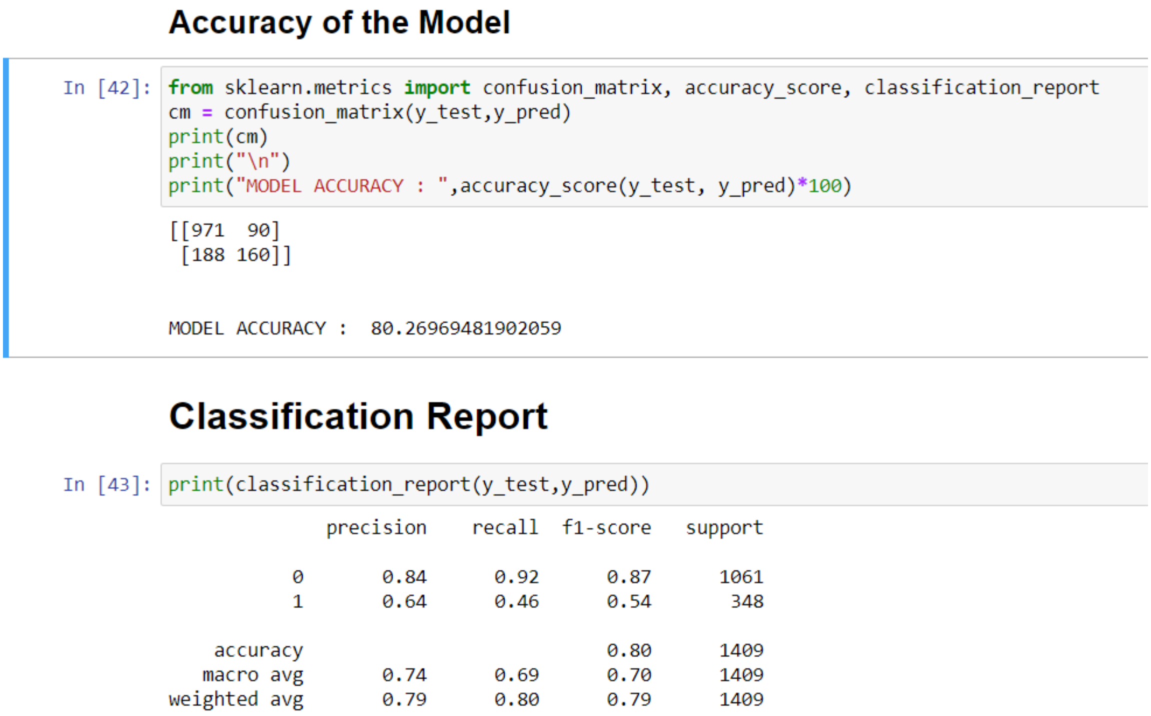
```
In [53]: print(classification_report(y_test,y_pred))

                       precision    recall  f1-score   support

                    0       0.86      0.89      0.87      1061
                    1       0.61      0.55      0.58       348

             accuracy                           0.80      1409
            macro avg       0.74      0.72      0.73      1409
         weighted avg       0.80      0.80      0.80      1409
```

7. Perceptron



Accuracy of the Model

```
In [42]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
         cm = confusion_matrix(y_test,y_pred)
         print(cm)
         print("\n")
         print("MODEL ACCURACY : ",accuracy_score(y_test, y_pred)*100)

         [[971  90]
          [188 160]]


         MODEL ACCURACY :  80.26969481902059
```

Classification Report

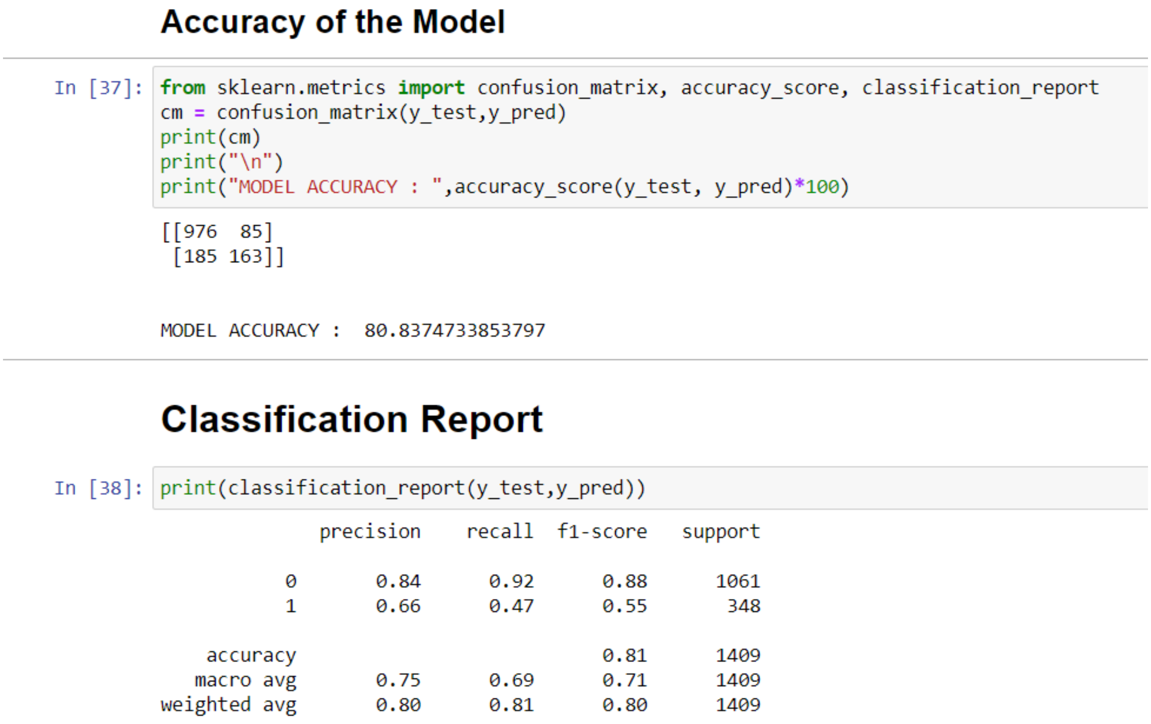
```
In [43]: print(classification_report(y_test,y_pred))

                       precision    recall  f1-score   support

                    0       0.84      0.92      0.87      1061
                    1       0.64      0.46      0.54       348

             accuracy                           0.80      1409
            macro avg       0.74      0.69      0.70      1409
         weighted avg       0.79      0.80      0.79      1409
```

8. Kernel SVM



Accuracy of the Model

```
In [37]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
         cm = confusion_matrix(y_test,y_pred)
         print(cm)
         print("\n")
         print("MODEL ACCURACY : ",accuracy_score(y_test, y_pred)*100)

         [[976  85]
          [185 163]]


         MODEL ACCURACY :  80.8374733853797
```

Classification Report

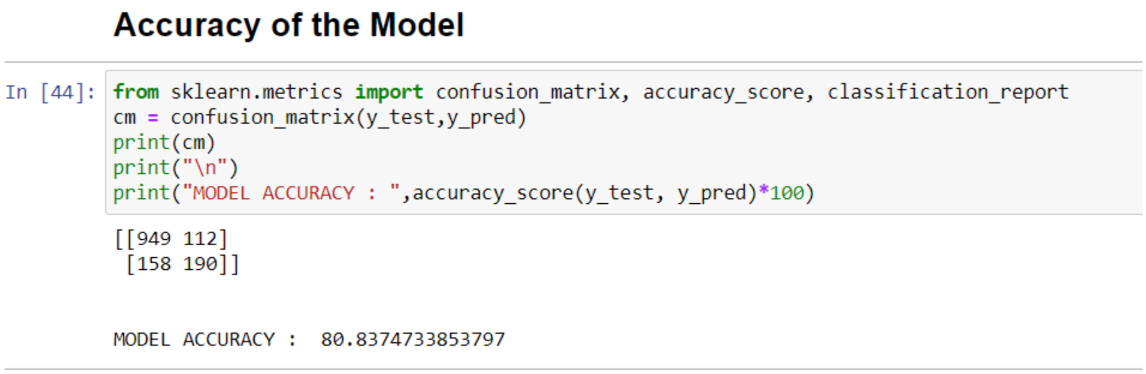
```
In [38]: print(classification_report(y_test,y_pred))

                       precision    recall  f1-score   support

                    0       0.84      0.92      0.88      1061
                    1       0.66      0.47      0.55       348

             accuracy                           0.81      1409
            macro avg       0.75      0.69      0.71      1409
         weighted avg       0.80      0.81      0.80      1409
```

9. AdaBoost Classifier

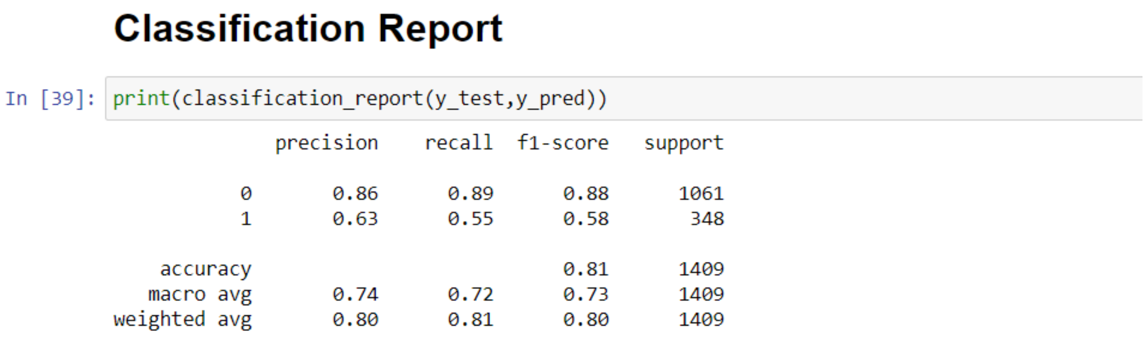


Accuracy of the Model

```
In [44]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
         cm = confusion_matrix(y_test,y_pred)
         print(cm)
         print("\n")
         print("MODEL ACCURACY : ",accuracy_score(y_test, y_pred)*100)

         [[949 112]
          [158 190]]


         MODEL ACCURACY :  80.8374733853797
```



Classification Report

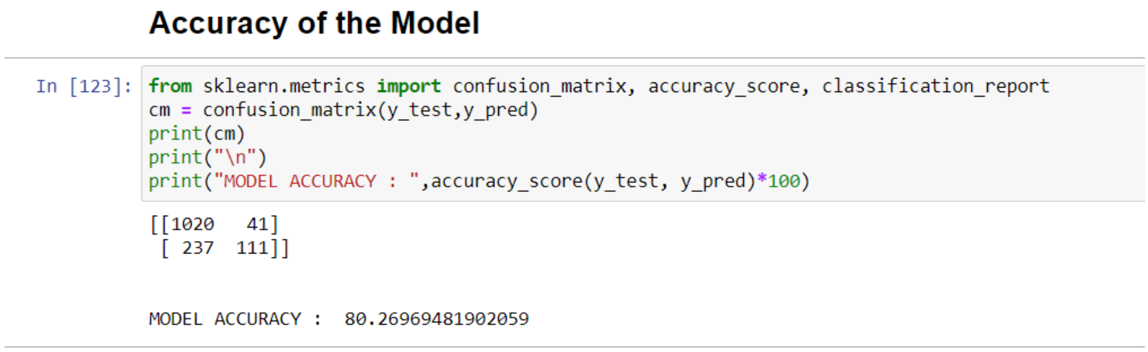
```
In [39]: print(classification_report(y_test,y_pred))

                       precision    recall  f1-score   support

                    0       0.86      0.89      0.88      1061
                    1       0.63      0.55      0.58       348

             accuracy                           0.81      1409
            macro avg       0.74      0.72      0.73      1409
         weighted avg       0.80      0.81      0.80      1409
```

10. Stochastic Gradient Descent Classifier

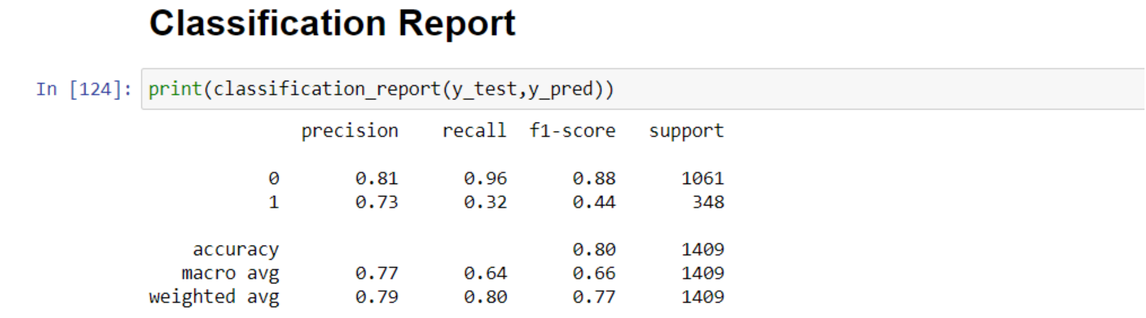


Accuracy of the Model

```
In [123]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
          cm = confusion_matrix(y_test,y_pred)
          print(cm)
          print("\n")
          print("MODEL ACCURACY : ",accuracy_score(y_test, y_pred)*100)

          [[1020   41]
           [ 237  111]]


          MODEL ACCURACY :  80.26969481902059
```



Classification Report

```
In [124]: print(classification_report(y_test,y_pred))

                        precision    recall  f1-score   support

                     0       0.81      0.96      0.88      1061
                     1       0.73      0.32      0.44       348

              accuracy                           0.80      1409
             macro avg       0.77      0.64      0.66      1409
          weighted avg       0.79      0.80      0.77      1409
```