Performance Metrics

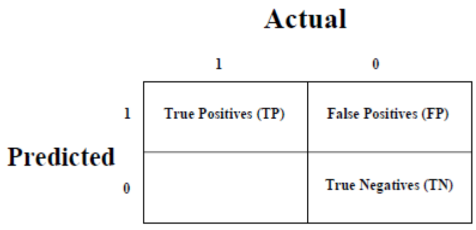
After doing the usual Feature Engineering, Selection, and of course, implementing a model and getting some output in forms of a probability or a class, the next step is to find out how effective is the model based on some metric using test datasets. Different performance metrics are used to evaluate different Machine Learning Algorithms. For now, we will be focusing on the ones used for Classification problems. We can use classification performance metrics such as Log-Loss, Accuracy, AUC(Area under Curve) etc. Another example of metric for evaluation of machine learning algorithms is precision, recall, which can be used for sorting algorithms primarily used by search engines.

The metrics that you choose to evaluate your machine learning model is very important. Choice of metrics influences how the performance of machine learning algorithms is measured and compared. Before wasting any more time, let’s jump right in and see what those metrics are.

Confusion Matrix:

The Confusion matrix is one of the most intuitive and easiest (unless of course, you are not confused) metrics used for finding the correctness and accuracy of the model. It is used for Classification problem where the output can be of two or more types of classes.

A confusion matrix is nothing but a table with two dimensions viz. “Actual” and “Predicted” and furthermore, both the dimensions have “True Positives (TP)”, “True Negatives (TN)”, “False Positives (FP)”, “False Negatives (FN)” as shown below –



		Actual	
		1	0
Predicted	1	True Positives (TP)	False Positives (FP)
	0		True Negatives (TN)

True Positives (TP): True positives are the cases when the actual class of the data point was 1(True) and the predicted is also 1(True).

True Negatives (TN): True negatives are the cases when the actual class of the data point was 0(False) and the predicted is also 0(False).

False Positives (FP): False positives are the cases when the actual class of the data point was 0(False) and the predicted is 1(True). False is because the model has predicted incorrectly and positive because the class predicted was a positive one. (1)

False Negatives (FN): False negatives are the cases when the actual class of the data point was 1(True) and the predicted is 0(False). False is because the model has predicted incorrectly and negative because the class predicted was a negative one. (0)

Classification Accuracy

It is most common performance metric for classification algorithms. It may be defined as the number of correct predictions made as a ratio of all predictions made. We can easily calculate it by confusion matrix with the help of following formula −

Accuracy=TP+TNTP+FP+FN+TNAccuracy=TP+TNTP+FP+FN+TN

We can use *accuracy_score* function of *sklearn.metrics* to compute accuracy of our classification model.

Classification Report

This report consists of the scores of Precisions, Recall, F1 and Support. They are explained as follows

Precision

Precision, used in document retrievals, may be defined as the number of correct documents returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula −

Precision=TPTP+FNPrecision=TPTP+FN

Recall or Sensitivity

Recall may be defined as the number of positives returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula.

Recall=TPTP+FNRecall=TPTP+FN

Specificity

Specificity, in contrast to recall, may be defined as the number of negatives returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula −

Specificity=TNTN+FPSpecificity=TNTN+FP

Support

Support may be defined as the number of samples of the true response that lies in each class of target values.

F1 Score

This score will give us the harmonic mean of precision and recall. Mathematically, F1 score is the weighted average of the precision and recall. The best value of F1 would be 1 and worst would be 0. We can calculate F1 score with the help of following formula −

F1=2∗(precision∗recall)/(precision+recall)F1=2∗(precision∗recall)/(precision+recall)

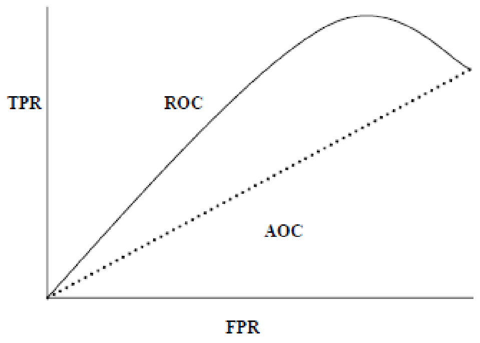
F1 score is having equal relative contribution of precision and recall.

We can use classification_report function of sklearn.metrics to get the classification report of our classification model.

AUC (Area Under ROC curve)

AUC (Area Under Curve)-ROC (Receiver Operating Characteristic) is a performance metric, based on varying threshold values, for classification problems. As name suggests, ROC is a probability curve and AUC measure the separability. In simple words, AUC-ROC metric will tell us about the capability of model in distinguishing the classes. Higher the AUC, better the model.

Mathematically, it can be created by plotting TPR (True Positive Rate) i.e. Sensitivity or recall vs FPR (False Positive Rate) i.e. 1-Specificity, at various threshold values. Following is the graph showing ROC, AUC having TPR at y-axis and FPR at x-axis –



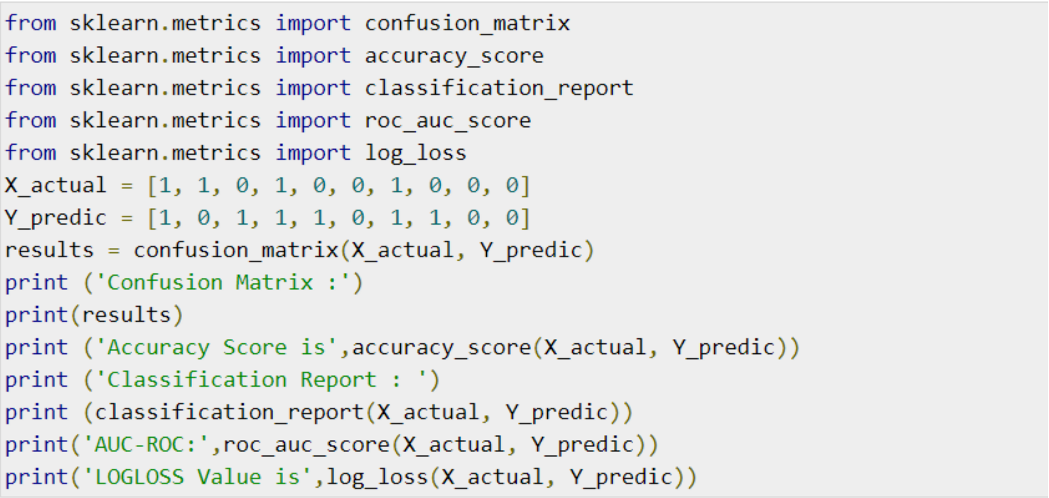
We can use roc_auc_score function of sklearn.metrics to compute AUC-ROC.

LOGLOSS (Logarithmic Loss)

It is also called Logistic regression loss or cross-entropy loss. It basically defined on probability estimates and measures the performance of a classification model where the input is a probability value between 0 and 1. It can be understood more clearly by differentiating it with accuracy. As we know that accuracy is the count of predictions (predicted value = actual value) in our model whereas Log Loss is the amount of uncertainty of our prediction based on how much it varies from the actual label. With the help of Log Loss value, we can have more accurate view of the performance of our model. We can use log_loss function of sklearn.metrics to compute Log Loss.

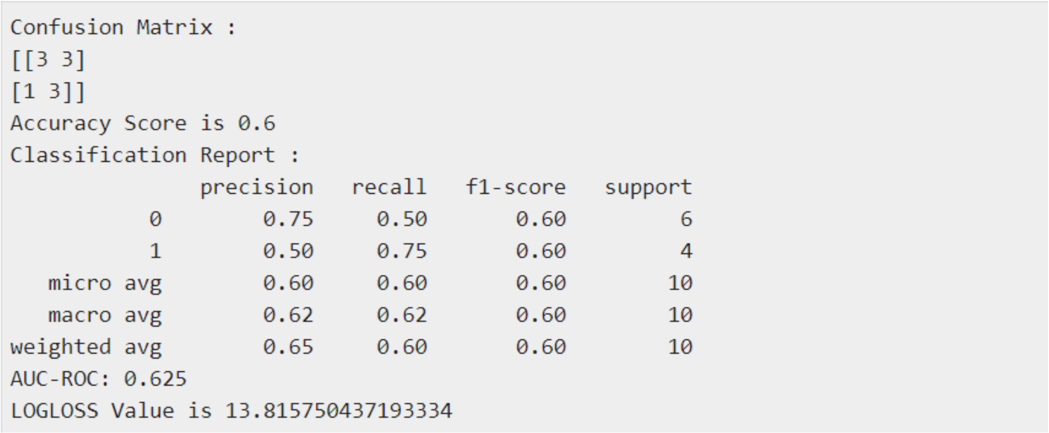
Example

The following is a simple recipe in Python which will give us an insight about how we can use the above explained performance metrics on binary classification model −



```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
from sklearn.metrics import log_loss
X_actual = [1, 1, 0, 1, 0, 0, 1, 0, 0, 0]
Y_predic = [1, 0, 1, 1, 1, 0, 1, 1, 0, 0]
results = confusion_matrix(X_actual, Y_predic)
print ('Confusion Matrix :')
print(results)
print ('Accuracy Score is',accuracy_score(X_actual, Y_predic))
print ('Classification Report : ')
print (classification_report(X_actual, Y_predic))
print('AUC-ROC:',roc_auc_score(X_actual, Y_predic))
print('LOGLOSS Value is',log_loss(X_actual, Y_predic))
```

Output



```
Confusion Matrix :
[[3 3]
[1 3]]
Accuracy Score is 0.6
Classification Report :
              precision    recall  f1-score   support
           0       0.75      0.50      0.60         6
           1       0.50      0.75      0.60         4
   micro avg       0.60      0.60      0.60        10
   macro avg       0.62      0.62      0.60        10
weighted avg       0.65      0.60      0.60        10
AUC-ROC: 0.625
LOGLOSS Value is 13.815750437193334
```