Classification Accuracy

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

Confusion Matrix

Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

Lets assume we have a binary classification problem. We have some samples belonging to two classes : YES or NO. Also, we have our own classifier which predicts a class for a given input sample. On testing our model on 165 samples ,we get the following result.

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

Confusion Matrix

There are 4 important terms :

- **True Positives** : The cases in which we predicted YES and the actual output was also YES.

- **True Negatives** : The cases in which we predicted NO and the actual output was NO.

- **False Positives** : The cases in which we predicted YES and the actual output was NO.

- **False Negatives** : The cases in which we predicted NO and the actual output was YES.

Accuracy for the matrix can be calculated by taking average of the values lying across the **"main diagonal"** i.e

$$Accuracy = \frac{True Positive + True Negative}{Total Sample}$$

$$\therefore Accuracy = \frac{100 + 50}{165} = 0.91$$

Area Under Curve

Area Under Curve(AUC) is one of the most widely used metrics for evaluation. It is used for binary classification problem. *AUC* of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. Before defining *AUC*, let us understand two basic terms :

- **True Positive Rate (Sensitivity)** : True Positive Rate is defined as $TP/(FN+TP)$. True Positive Rate corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points.

$$TruePositiveRate = \frac{TruePositive}{FalseNegative + TruePositive}$$

- **False Positive Rate** : False Positive Rate is defined as *FP / (FP+TN)*. False Positive Rate corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points.

$$FalsePositiveRate = \frac{FalsePositive}{TrueNegative + FalsePositive}$$

Receiver operating characteristic example
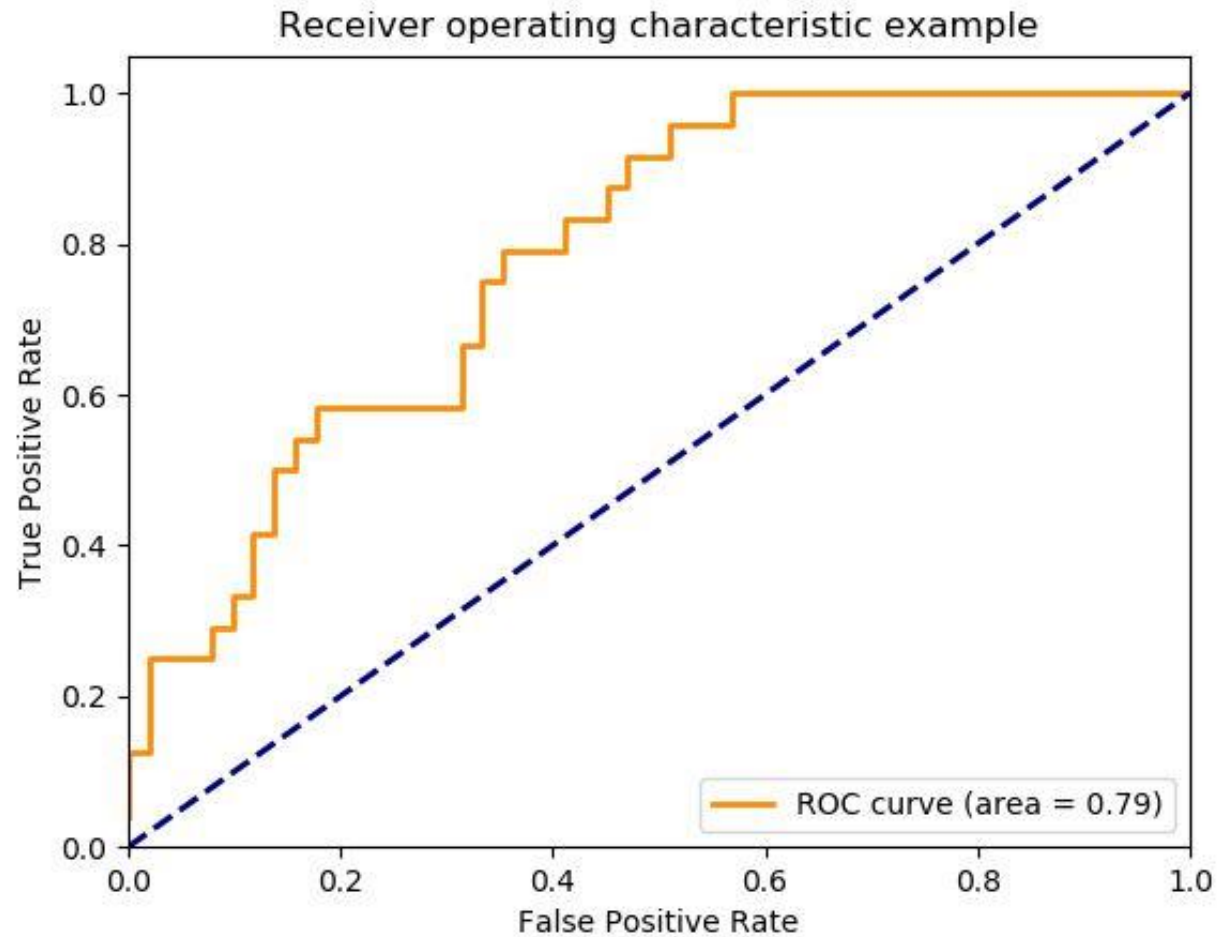
As evident, *AUC* has a range of [0, 1]. The greater the value, the better is the performance of our model.

F1 Score

F1 Score is used to measure a test's accuracy

F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).

High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model. Mathematically, it can be expressed as :

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

F1 Score

F1 Score tries to find the balance between precision and recall.

- **Precision :** It is the number of correct positive results divided by the number of positive results predicted by the classifier.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

Precision

- **Recall :** It is the number of correct positive results divided by the number of *all* relevant samples (all samples that should have been identified as positive).

$$Precision = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Recall

Mean Absolute Error

Mean Absolute Error is the average of the difference between the Original Values and the Predicted Values. It gives us the measure of how far the predictions were from the actual output. However, they don't gives us any idea of the direction of the error i.e. whether we are under predicting the data or over predicting the data. Mathematically, it is represented as :

$$MeanAbsoluteError = \frac{1}{N} \sum_{j=1}^{N} |y_j - \hat{y}_j|$$

Mean Squared Error

Mean Squared Error(MSE) is quite similar to Mean Absolute Error, the only difference being that MSE takes the average of the **square** of the difference between the original values and the predicted values. The advantage of MSE being that it is easier to compute the gradient, whereas Mean Absolute Error requires complicated linear programming tools to compute the gradient. As, we take square of the error, the effect of larger errors become more pronounced then smaller error, hence the model can now focus more on the larger errors.

$$MeanSquaredError = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2$$

Mean Squared Error

- Assume that we have a binary classification problem. We have some samples belonging to two classes: YES or NO. Also, we have our own classifier which predicts a class for a given input sample. On testing our model on 180 samples, we get the confusion matrix in which Ture Positives = 100, True Negatives = 55, False Positives = 15, False Negatives = 10. Calculate accuracy (approximate value).

- Assume that we have a binary classification problem. We have some samples belonging to two classes: YES or NO. Also, we have our own classifier which predicts a class for a given input sample. On testing our model on 180 samples, we get the confusion matrix in which Ture Positives = 100, True Negatives = 55, False Positives = 15, False Negatives = 10. Calculate recall (approximate value).

- Assume that we have a binary classification problem. We have some samples belonging to two classes: YES or NO. Also, we have our own classifier which predicts a class for a given input sample. On testing our model on 180 samples, we get the confusion matrix in which Ture Positives = 100, True Negatives = 55, False Positives = 15, False Negatives = 10. Calculate precision (approximate value).

- Assume that we have a binary classification problem. We have some samples belonging to two classes: YES or NO. Also, we have our own classifier which predicts a class for a given input sample. On testing our model on 180 samples, we get the confusion matrix in which Ture Positives = 100, True Negatives = 55, False Positives = 15, False Negatives = 10. Calculate F1 score (approximate value).

- Assume that we have a binary classification problem. We have some samples belonging to two classes: YES or NO. Also, we have our own classifier which predicts a class for a given input sample. On testing our model on 180 samples, we get the confusion matrix in which Ture Positives = 100, True Negatives = 55, False Positives = 15, False Negatives = 10. Calculate True Positive Rate (approximate value).

- Assume that we have a binary classification problem. We have some samples belonging to two classes: YES or NO. Also, we have our own classifier which predicts a class for a given input sample. On testing our model on 180 samples, we get the confusion matrix in which Ture Positives = 100, True Negatives = 55, False Positives = 15, False Negatives = 10. Calculate False Positive Rate (approximate value).