Classification: Performance Metrics for Classification Models

Prof. (Dr.) Honey Sharma

Performance Evaluation Measures for Classification Models

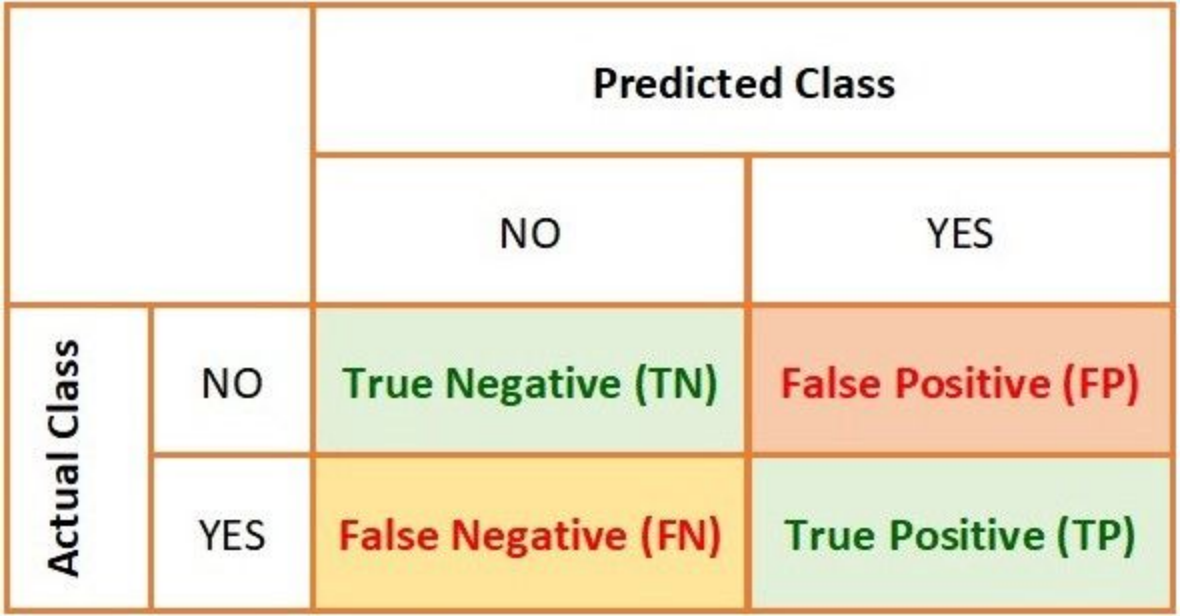
- Confusion Matrix
- Precision
- Recall/ Sensitivity
- Specificity
- F1-Score
- AUC & ROC Curve

Confusion Matrix

Confusion Matrix is used to know the performance of a Machine learning classification.

It is represented in a matrix form, N x N matrix, where N is the number of classes or outputs.

Confusion Matrix gives a comparison between actual and predicted values. It contains the count of observations that fall in each category.



		Predicted Class	
		NO	YES
Actual Class	NO	True Negative (TN)	False Positive (FP)
	YES	False Negative (FN)	True Positive (TP)

Let’s take an example of a patient who has gone to a doctor with certain symptoms of Covid fever, cough, throat ache, and cold. These are symptoms that can occur during any seasonal changes too. Hence, it is tricky for the doctor to do the right diagnosis.

True Positive (TP):

Let’s say the patient was actually suffering from Covid and on doing the required assessment, the doctor classified him as a Covid patient. This is called TP or True Positive.

False Positive (FP):

Let’s say the patient was not suffering from Covid and he was only showing symptoms of seasonal flu but the doctor diagnosed him with Covid. This is an unnecessary inconvenience for him and others as he will get unwanted treatment and quarantine. **This is called Type I Error.**

True Negative (TN):

Let’s say the patient was not suffering from Covid and the doctor also gave him a clean chit. This is called TN or True Negative.

False Negative (FN):

Let’s say the patient was suffering from Covid and the doctor did not diagnose him with Covid. This is called FN or False Negative. This is a highly dangerous situation in this example. **This is also called Type II Error.**

In this particular example, both FN and FP are dangerous and the classification model which has the lowest FN and FP values needs to be chosen for implementation.

But in case there is a tie between few models which score very similar when it comes to FP and FN, in this scenario the model with the least FN needs to be chosen.

This is because we simply cannot afford to have FNs! The goal of the hospital would be to not let even one patient go undiagnosed (no FNs) even if some patients get diagnosed wrongly (FPs) and asked to go under quarantine and special care.

Accuracy

Ratio of correct predictions to total predictions.

Accuracy = (TP + TN) / (TP + FP +TN + FN)

Important when: you have symmetric datasets (FN & FP counts are close)

Used when: false negatives & false positives have similar costs.

The best accuracy is 100% indicating that all the predictions are correct. For an imbalanced dataset, accuracy is not a valid measure of model performance.

Sensitivity/Recall

Ratio of true positives to total (actual) positives in the data.

Sensitivity or Recall = TP/(TP+FN)

Important when: identifying the positives is crucial.

Used when: the occurrence of false negatives is unacceptable/intolerable. You would rather have some extra false positives (false alarms) over saving some false negatives. For example, when predicting financial default or a deadly disease.

Recall or sensitivity gives us information about a model’s performance on false negatives (incorrect prediction of customers who will default)

Precision

Ratio of true positives to total predicted positives.

Important when: you want to be more confident of your predicted positives.

Used when: the occurrence of false positives is unacceptable/intolerable. For example, Spam emails. You’d rather have some spam emails in your inbox than miss out some regular emails that were incorrectly sent to your spam box.

Precision = TP/(TP+FP)

Specificity

Ratio of true negatives to total negatives in the data.

Important when: you want to cover all true negatives.

Used when: you don’t want to raise false alarms. For example, you’re running a drug test in which all people who test positive will immediately go to hospital.

Specificity = TN/(TN+FP)

F1-Score

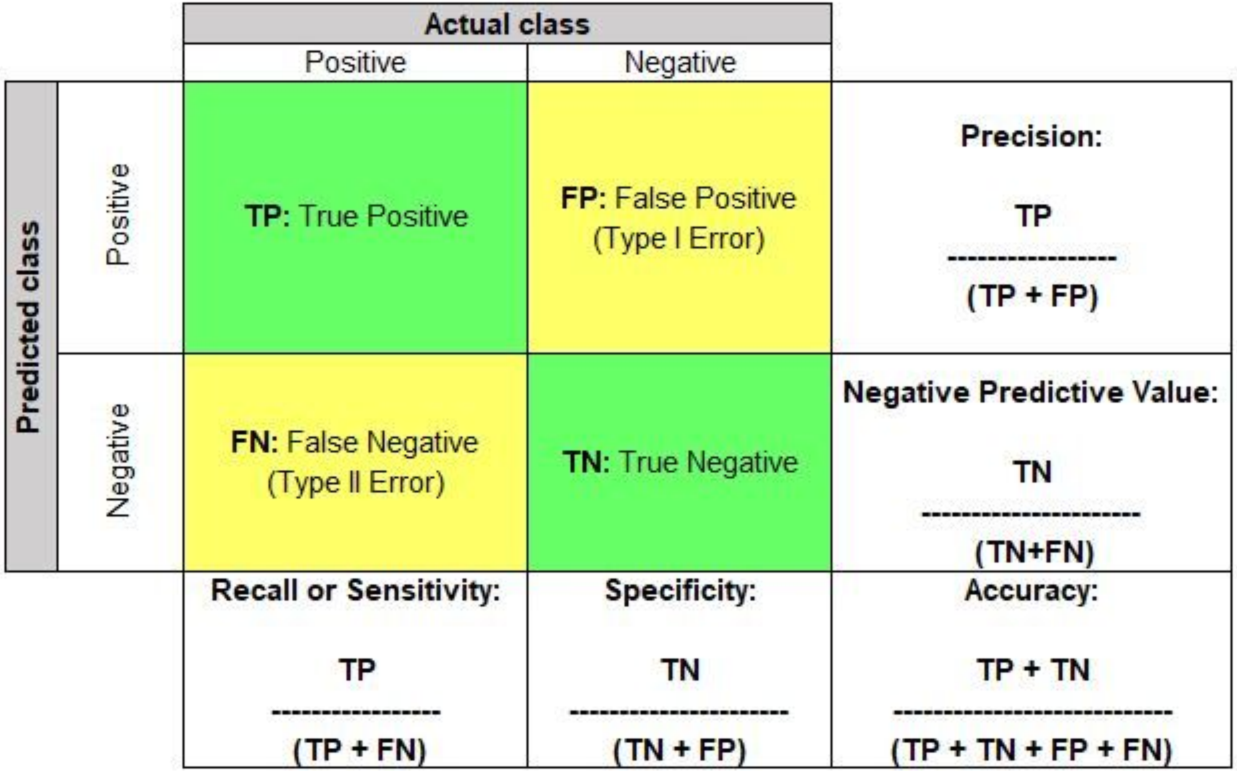
Considers both precision and recall. It’s the harmonic mean of the precision and recall.

Important when: you have an uneven class distribution.

Used when: This is a very useful metric compared to “Accuracy”. The problem with using accuracy is that if we have a highly imbalanced dataset for training (for example, a training dataset with 95% positive class and 5% negative class), the model will end up learning how to predict the positive class properly and will not learn how to identify the negative class. But the model will still have very high accuracy in the test dataset too as it will know how to identify the positives really well.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Let’s take an example where we must give equal importance to both the classes – classify an email as Spam and non-Spam. Let’s assume that the model was trained only a highly imbalanced training dataset. Here, Spam is “positive” and non-Spam is “negative” and the training dataset was 90% spam emails and 10% non-spam emails. A model with high accuracy will know to correctly identify all the spam emails but will have trouble identifying non-spam emails. Hence, a lot of important emails will end up going to the spam folder. But if we select a model that has a high F1 score, it would perform better in classifying non-spam from spam.



		Actual class		
		Positive	Negative	
Predicted class	Positive	**TP:** True Positive	**FP:** False Positive (Type I Error)	**Precision:** **TP** **----------------** **(TP + FP)**
	Negative	**FN:** False Negative (Type II Error)	**TN:** True Negative	**Negative Predictive Value:** **TN** **---------------------** **(TN+FN)**
		Recall or Sensitivity: **TP** **----------------** **(TP + FN)**	**Specificity:** **TN** **---------------------** **(TN + FP)**	**Accuracy:** **TP + TN** **---------------------------** **(TP + TN + FP + FN)**

Optimal Probability Threshold — ROC Curve

Say we were building an email classification model to detect suspicious communication between terrorists over email.

In that case a terrorist email is Class YES and a non-terrorist email is Class NO.

We choose **Sensitivity** as a metric to improve this model. Because it is absolutely necessary for the model to identify the terrorists’ emails correctly.

For the model to be considered useful, its true positive rate should be high. In this pursuit, we might end up having a **few false positives/false alarms** but that might be a compromise that we’ll have to make.

Let us say we end up with 2 really good models which have the same Sensitivity score.

The most of classification algorithms predict the probability that an observation belongs to class YES. We need to decide a threshold for these probabilities, to classify the observations into one of the two classes. Say, we get the probability of an email being a terrorist email as 0.75. If we have set the threshold of our system as 0.8, then we will classify this email as non-terrorist email. If we have set the threshold as 0.7, we will classify the email as a terrorist email.

The performance of our system would vary as we change this threshold. This threshold can be adjusted to tune the behavior of the model for a specific problem.

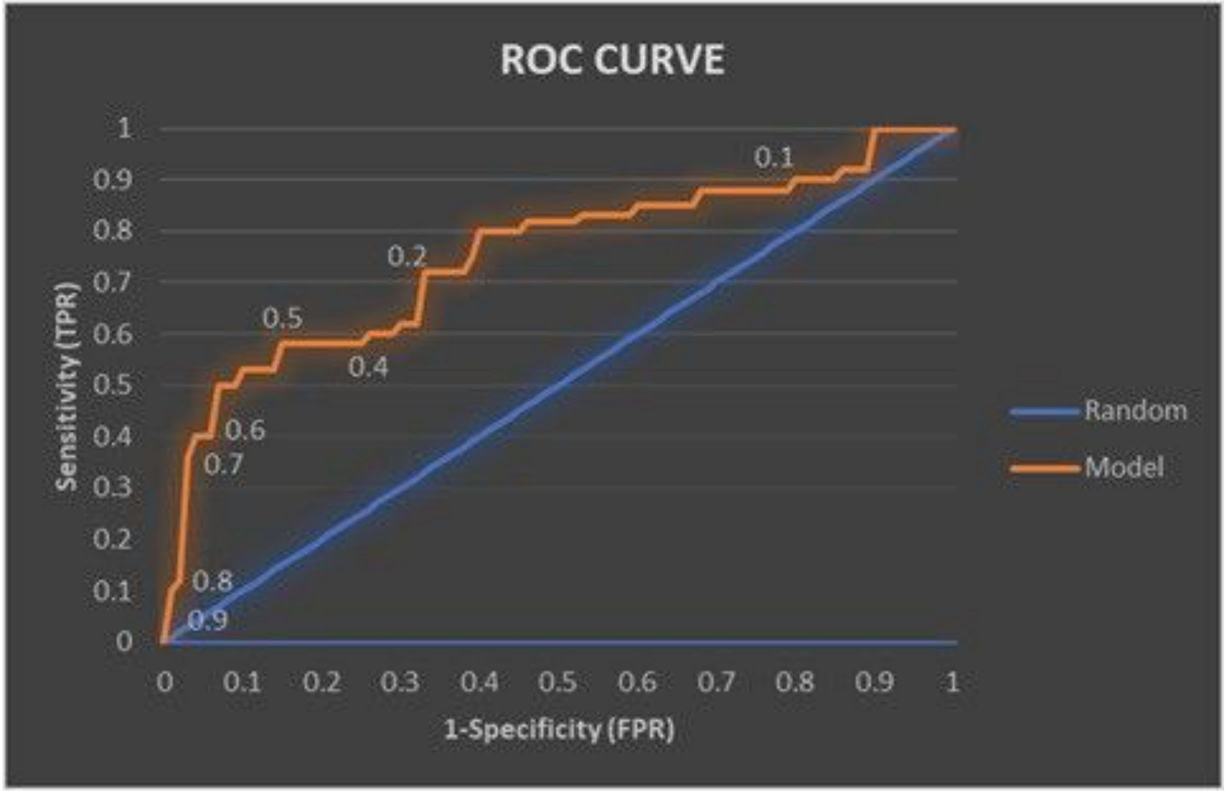
Two models with the same sensitivity (TPR) are not equivalent. Among these two, the model with a lower FPR (Specificity) is obviously a better, more reliable model. We do not want to waste any of our investigative resources on non-terrorist emails that were misclassified as terrorist emails.

The threshold that we set, can help us increase or decrease the TPR. If we choose a low threshold, more emails will be classified as terrorist emails, we will be able to catch more true positives but then the false positive rate would increase. The choice of threshold entails a trade-off between false positives and false negatives.

Receiver Operating Characteristic (ROC) Curve

The ROC Curve is a plot of the True Positive Rate/Sensitivity (y-axis) versus the False Positive Rate/1-Specificity (x-axis) for candidate threshold values between 0.0 and 1.0.

ROC curve is plot on all possible thresholds.


ROC CURVE
Sensitivity (TPR)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1-Specificity (FPR)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.1
0.2
0.5
0.4
0.6
0.7
0.8
0.9
Random
Model

1. In the above curve if you wanted a model with a very low false positive rate, you might pick 0.8 as your threshold of choice. If you favour a low FPR, but you don’t want an abysmal TPR, you might go for 0.5.

If you prefer a low false negative rate/high Sensitivity (because you don’t want to miss potential terrorists, for example), then you might decide that somewhere between 0.2 and 0.1 is the region where you start getting severely diminishing returns for improving the Sensitivity any further.

2. Notice the graph at threshold 0.5 and 0.4. The Sensitivity at both the thresholds is ~0.6, but FPR is higher at threshold 0.4. It’s clear that if we are happy with Sensitivity = 0.6 we should choose threshold = 0.5.

The ROC curve is great for choosing a threshold. Its shape contains a lot of information:

a) Smaller values on the x-axis of the plot indicate lower false positives and higher true negatives.

b) Larger values on the y-axis of the plot indicate higher true positives and lower false negatives.

c) A model that has high y values at low x values is a good model.

The area under the ROC Curve is also known as AUC (Area Under the Curve).

AUC is another performance metric that we can use to improve our models on. AUC represents degree or measure of separability. It tells us how much the model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting class YES as YES and NO as NO.

AUC ignores the threshold and prevalence and gives us a measure of separability of the model.