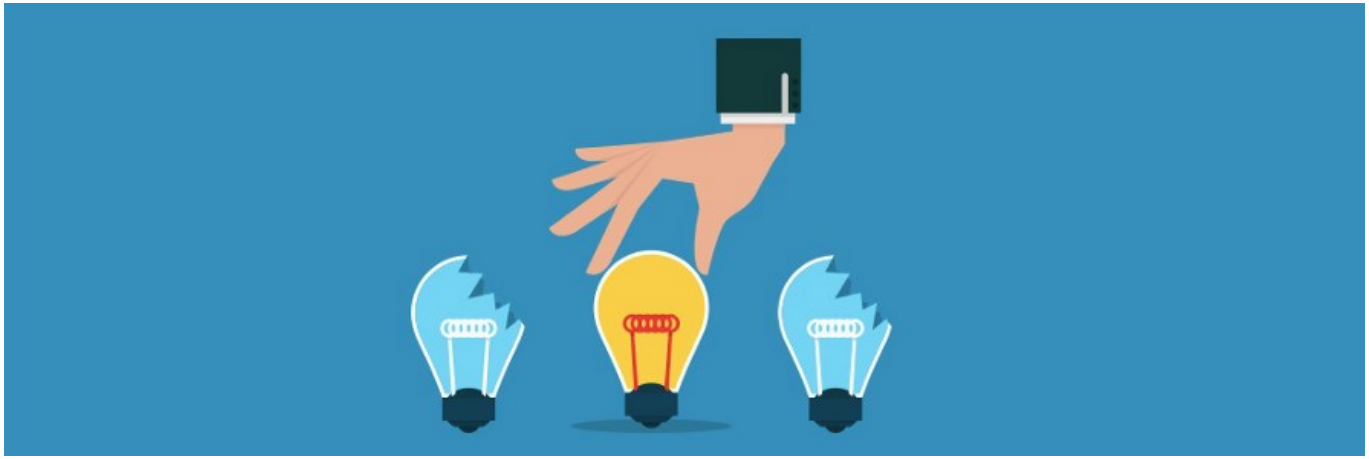
How to select Performance Metrics for Classification Models

Accuracy, Sensitivity, Specificity, Precision, F1 Score, Probability Threshold, AUC, ROC Curve

Ruchi Toshniwal Follow

Jan 9, 2020 · 8 min read



We use Classification Models to predict class labels for a given input data. To evaluate such a model, we can choose any of the various metrics available to us, like Accuracy, Sensitivity, Specificity, Precision, F1 Score, Probability Threshold, AUC, ROC Curve . **It is important that this choice is backed by analytical reasoning**.
Often, we choose *Model Accuracy* to evaluate the model. It's a popular choice because it is very easy to understand and explain. **Accuracy coincides well with the general aim of building a classification model, i.e. to predict the class of new observations accurately.**

Accuracy might not be the best model evaluation metric every time. It can convey the health of a model well only when all the classes have similar prevalence in the data.

Say we were predicting if an asteroid will hit the earth?

If our model says NO every time, it will be highly accurate but it would not be of much value to us. The number of asteroids that will hit the earth is very low but missing even one of them might prove very costly. When the classes' distribution is imbalanced, accuracy is not a good model evaluation metric.

For this article let us focus on binary classification (two output classes). These metrics can be extended to multi-class classification problems also.

Confusion Matrix

Confusion matrix is a very **intuitive cross tab** of actual class values and predicted class values. It contains the count of observations that fall in each category.

Build model → make class predictions on test data using the model → create a confusion matrix for each model. Use one of the following ratios to compare any two models.

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

Let us see all the metrics that can be derived from confusion matrix and when to use them:

1. **Accuracy** — Ratio of correct predictions to total predictions.
Important when: you have symmetric datasets (FN & FP counts are close)
Used when: false negatives & false positives have similar costs.

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

2. **Sensitivity/Recall** — Ratio of true positives to total (actual) positives in the data.
Important when: identifying the positives is crucial.
Used when: the occurrence of false negatives is unacceptable/intolerable. You'd rather have some extra false positives (false alarms) over saving some false negatives. For example, when predicting financial default or a deadly disease.

Sensitivity or Recall = TP/(TP+FN)

3. **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)

4. **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 jail.

Specificity = TN/(TN+FP)

5. **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: the cost of false positives and false negatives are different. F1 score conveys the balance between the precision and the recall. It is higher if there is a balance between Precision and Recall. F1 Score isn't so high if

one of these measures, Precision or Recall, is improved at the expense of the other.

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

Notice that each one of these is defined in such a way that they capture different aspects of a model's performance. When we are choosing one of these metrics to improve our model on, we need to keep in mind:

a) The problem that we are trying to solve

b) The dataset that we have with us (prevalence of each class in the data)

c) The cost that we have to pay for either types of misclassifications (false positives and false negatives)

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.

Why?
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. Does that mean the two models have equal predictive power? NO.

Recall that most 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. The observation having probability higher than the threshold are classified as class YES.
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.**

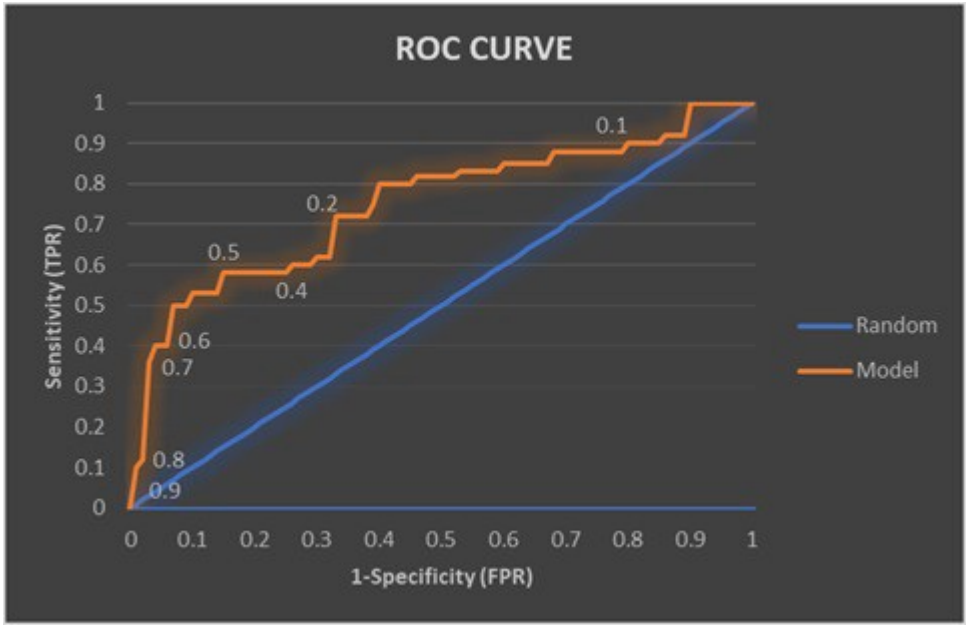
This threshold can be adjusted to tune the behavior of the model for a specific problem. An example would be to reduce more of one or another type of error (FP/FN).

Now, these are two different but inter-related problems that we have just discussed. Two models with the same sensitivity (TPR) are not equivalent. Among these two, the model with a lower FPR 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 even the false positive rate would increase. The choice of threshold entails a trade-off between false positives and false negatives.

An ROC curve is a useful resource in this regard.

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.



The points (grey) on the orange curve are the corresponding thresholds.

ROC curve is plot on all possible thresholds.

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, the point where the curve starts turning hard to the right.

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.

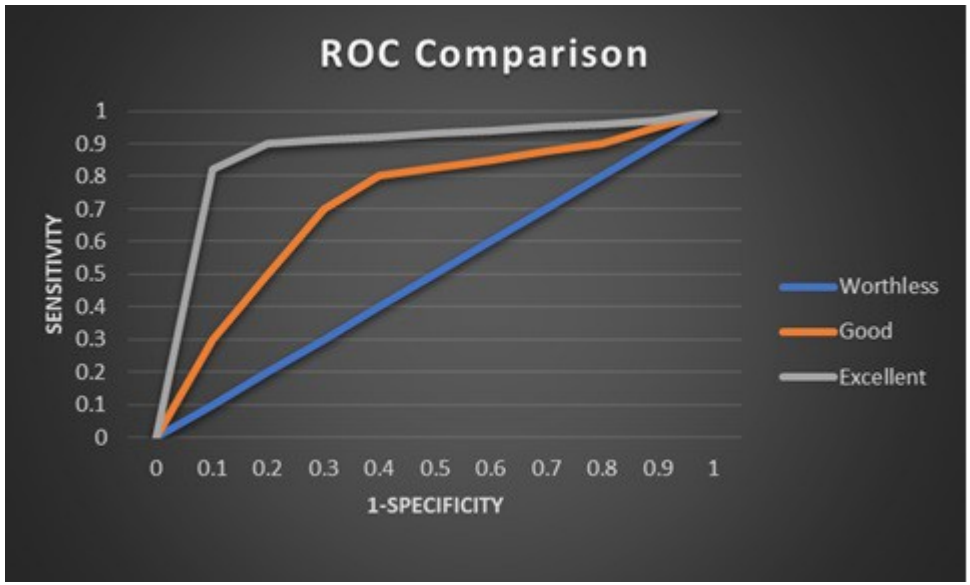
c) A model that has high y values at low x values is a good model.

The ROC curve is a very useful tool for a few additional reasons:

a) The curves of different models can be compared directly in general or for different thresholds.

b) The area under the curve (AUC) can be used as a summary of the model skill.

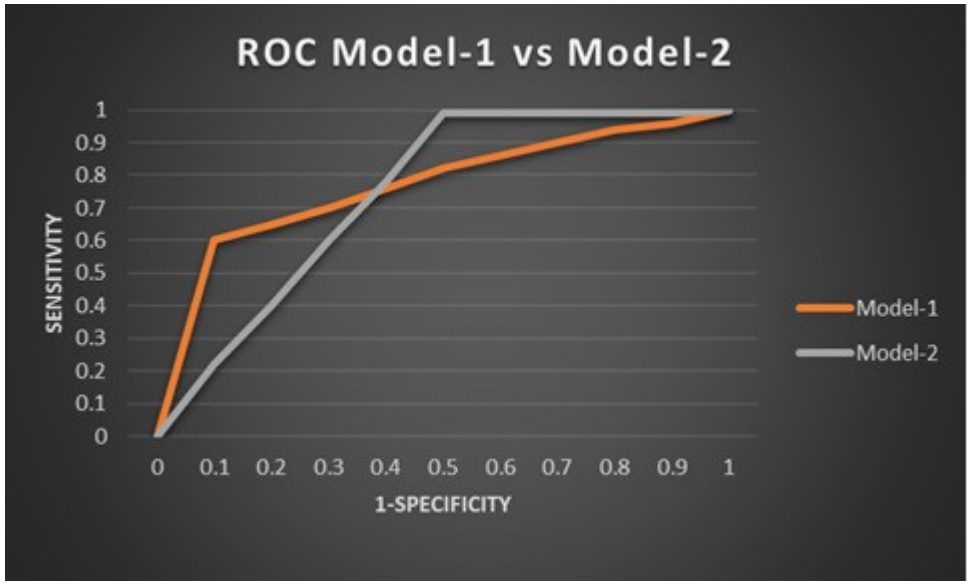
Comparing ROC Curves



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.

Greater the AUC the better the classifier/model.



ROC Curves of both Model-1 and Model-2 have the same area under the curve. But when we pick a threshold, we want to look where the steepest and flattest parts of the curve start and stop.

Even though the AUC is same for both the models, the choice of model and threshold depends on the problem that we are trying to solve and the cost that we can afford to pay for misclassifications.

In Conclusion

All the performance metrics that we have discussed above are derived from the confusion matrix. They are inter-related with each other. It is important to identify the metric that suits the problem at hand the best, as we start building the model. At the same time, it is very important to recognize that none of these metrics can convey the full health of a model individually. It is therefore suggested that we use an appropriate combination of these metrics when we are working on improving a machine learning model.

Thanks for reading. Looking forward to hear from you :)

Machine Learning Sensitivity Roc Curve Specificity Model Performance