Performance Measures

Classification

Accuracy

To measure the performance of your classification model, several performance measures exist. The most well-known measure is **accuracy**.

$$accuracy = \frac{\#\ of\ correct\ classifications}{total\ \#\ of\ instances}$$

Accuracy is the simplest measure to understand, but in many cases, it is not the preferred performance measure because it is very misleading. Imagine you have a dataset of cats and dogs that consists of 100 data points. Of the 100 data points, 85 are dogs. If one were optimizing for accuracy, an approach to achieving a high accuracy would be to assign every instance as dogs to achieve an accuracy of 85%. This metric is misleading as it oversells the performance of the model when, in reality, it is not any better than random guessing. Other metrics offer better ways to measure the performance of classification models.

Precision

$$precision = \frac{TP}{TP + FP}$$

This can be easily interpreted as:

$$precision = \frac{\#\ of\ correctly\ predicted\ positive\ instances}{total\ \#\ of\ positive\ predictions}$$

Precision is a useful metric when minimizing false positives is important. For example, a company that makes a video filter to remove unsafe videos for children's websites would want to prioritize high precision because keeping the safe videos (positive class) is of the utmost importance. The tradeoff might be that the company will block videos that are perfectly safe, but that might be a tradeoff that the company is willing to make.

Recall (a.k.a. True Positive Rate)

$$recall = \frac{TP}{TP + FN}$$

This can be easily interpreted as:

$$recall = \frac{\#\ of\ correctly\ predicted\ positive\ instances}{ground\ truth\ positive\ instances}$$

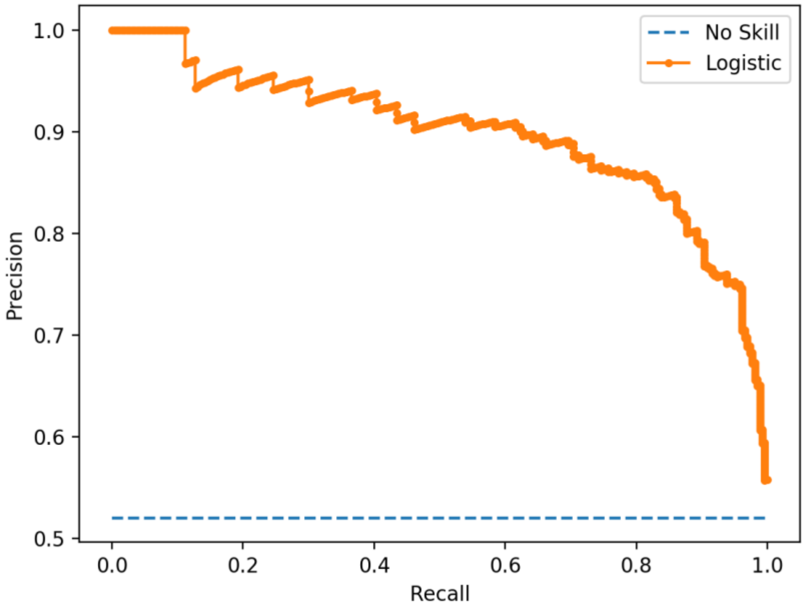
Recall is a useful metric when minimizing the false negatives are important. For example, a retail chain would want to aim for high recall and minimize false negatives. They would want to catch everyone suspected of shoplifting. The retailer would run the risk of catching a few innocent people, but the retailer might be willing to absorb that cost if not catching shoplifters is a higher cost.

Precision-Recall Tradeoff

Let's consider another scenario of traffic police pulling over drivers suspected of speeding. If the police department wants to crack down on reckless driving, then the department may emphasize recall. This comes with the cost that police officers may spend time ticketing drivers that were not speeding. Over-ticketing would lead to police offers wasting time, unnecessary paperwork, and those tickets being overturned.

On the other hand, the police department could instruct police offers to prioritize only those drivers that are recklessly speeding. The cost would be that drivers who are technically speeding would not be ticketed, the number of tickets issued would be fewer, which could result in a loss of revenue, but they would have to deal with less paperwork. Additionally, the policeman's time would be utilized better elsewhere.

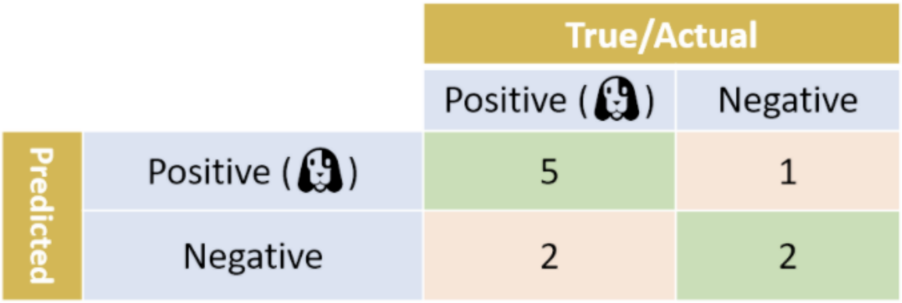
The tradeoff is commonly known as the *precision-recall tradeoff*. Choosing one metric or the other depends on the business priority and the task at hand, and a result means decreasing the other. Precision and recall can be graphed to get a better idea of where to threshold. In most cases, choosing the precision-recall tradeoff is done before a significant drop, so in the example below, somewhere around a recall of 60%. Obviously, this decision is up to you and your business priorities.



.

Precision and Recall with our Animal Classifier

Initially, our animal classifier only separates between dogs and animals that are not-dogs. After doing testing, we get a confusion matrix that looks like this. What is the precision & recall of our dogs-not-dogs classifier?

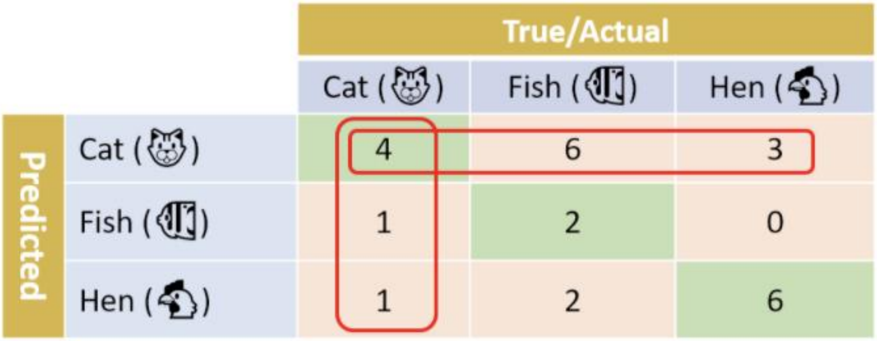


		True/Actual	
		Positive (🐶)	Negative
Predicted	Positive (🐶)	5	1
	Negative	2	2

$$\mathit{precision} = \frac{5}{5+1} = 0.83$$

$$\mathit{recall} = \frac{5}{5+2} = 0.714$$

Now, lets say we expand our classifier to include cats, hens, and fishes. What is our precision and recall then? In multiple classes, the precision & recall has to be calculated for each class. In this case we are going to do just the cat.



		True/Actual		
		Cat (🐱)	Fish (🐟)	Hen (🐔)
Predicted	Cat (🐱)	4	6	3
	Fish (🐟)	1	2	0
	Hen (🐔)	1	2	6

$$\mathit{precision}_{\text{cat}} = \frac{4}{4+6+3} = 0.308$$

$$\mathit{recall}_{\text{cat}} = \frac{4}{4+1+1} = 0.667$$

F1-Score

The F1 Score is the harmonic mean of precision and recall. The value ranges from 0 to 1, perfect precision and recall.

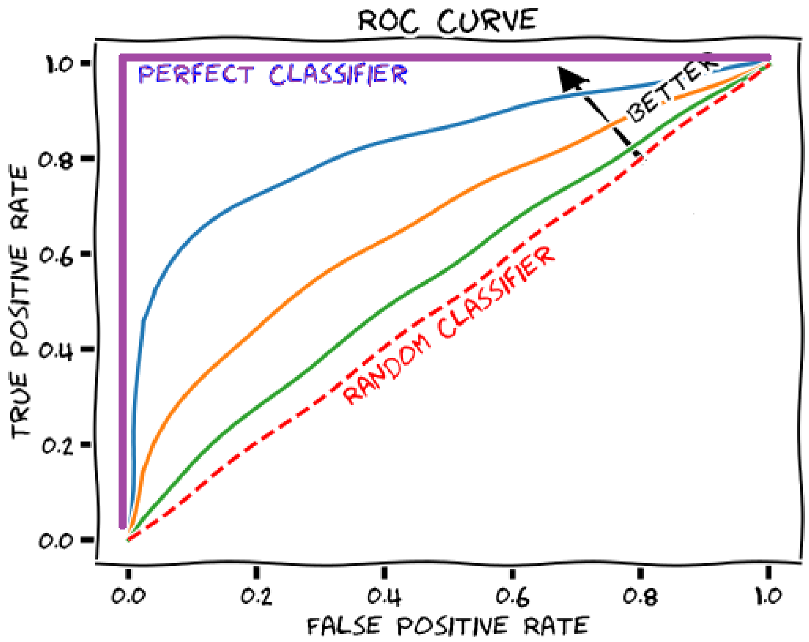
$$F_1 = \frac{2 * precision * recall}{precision + recall}$$

ROC/AUC

Receiver operating characteristic, ROC, is a graph showing recall vs. false-positive-rate, FPR.

$$false\ positive\ rate,\ FPR = \frac{FP}{TP + FP}$$

The ROC curve shows the tradeoffs between the recall (true-positive-rate) and the fall-positive-rate; in other words, it shows the ability of the classifier to differentiate between different classes at various thresholds. Classifiers that maximize the area underneath the ROC curve are better than ones that do not. A perfect classifier would have a point at (0,1) where the area-under-the-curve, AUC, would be 1. A classifier that does random guessing has an AUC of 0.50.



Which metric to choose?

How to choose between precision, recall, F1, AUC? If you want to balance precision and recall, then F1 score is the way to go. Now, the precision-recall curve and ROC curve are very similar. How do you choose between those? A good rule of thumb is to select the precision-recall curve to make your decision when the positive class is rare, and fall-positives are a higher priority, choose the ROC curve otherwise. Note in AutoAI, AUC is chosen by default.

Regression

Mean Square Error (MSE)

Mean Squared Error, MSE, is simply a difference of squares between the predicted value and the ground truth.

$$Mean\ Squared\ Error,\ MSE = \frac{1}{n}\Sigma_{k=1}^{n}\widehat{y_k}^2 - y_k^2;\ where\ \hat{y}\ is\ the\ predicted\ value$$

More commonly, the square root of the MSE is taken to get the *Root Mean Squared Error, RMSE.*

$$Root\ Mean\ Squared\ Error,\ RMSE = \sqrt{\frac{1}{n}\Sigma_{k=1}^{n}\widehat{y_k}^2 - y_k^2};\ where\ \hat{y}\ is\ the\ predicted\ value$$

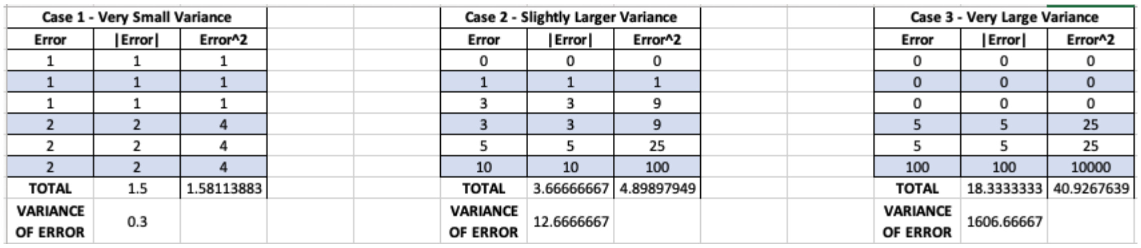
Mean Absolute Error (MAE)

Mean Absolute Error, MAE, is similar to RMSE but instead of the square root of the error, the absolute value of the error.

$$Mean\ Absolute\ Error,\ MAE = \Sigma_{k=1}^{n}|\hat{y}_k - y_k|$$

Which One To Choose?

Choosing MAE or RMSE depends on the problem you are trying to solve. RMSE squares the averages values before the square root is taken, so overly large errors are penalized more heavily. In other words, RMSE tends to grow larger if the frequency of the error magnitudes tends to grow larger. In the example below we see three cases of the RMSE error growing with the frequency of the error magnitudes.



Case 1 - Very Small Variance		
Error	|Error|	Error^2
1	1	1
1	1	1
1	1	1
2	2	4
2	2	4
2	2	4
TOTAL	1.5	1.58113883
VARIANCE OF ERROR	0.3	

Case 2 - Slightly Larger Variance		
Error	|Error|	Error^2
0	0	0
1	1	1
3	3	9
3	3	9
5	5	25
10	10	100
TOTAL	3.66666667	4.89897949
VARIANCE OF ERROR	12.6666667	

Case 3 - Very Large Variance		
Error	|Error|	Error^2
0	0	0
0	0	0
0	0	0
5	5	25
5	5	25
100	100	10000
TOTAL	18.3333333	40.9267639
VARIANCE OF ERROR	1606.66667	

In short, if your problem case requires that being off by 10 is **twice** as bad as being off by 5, then MAE is more suitable. However, if being off by 10 is **more than twice** as bad as being off by 5, then RMSE is the way to go. By default, for regression problems, AutoAI chooses RMSE.

Sources

Multi-Class Metrics Made Simple, Part I: Precision and Recall