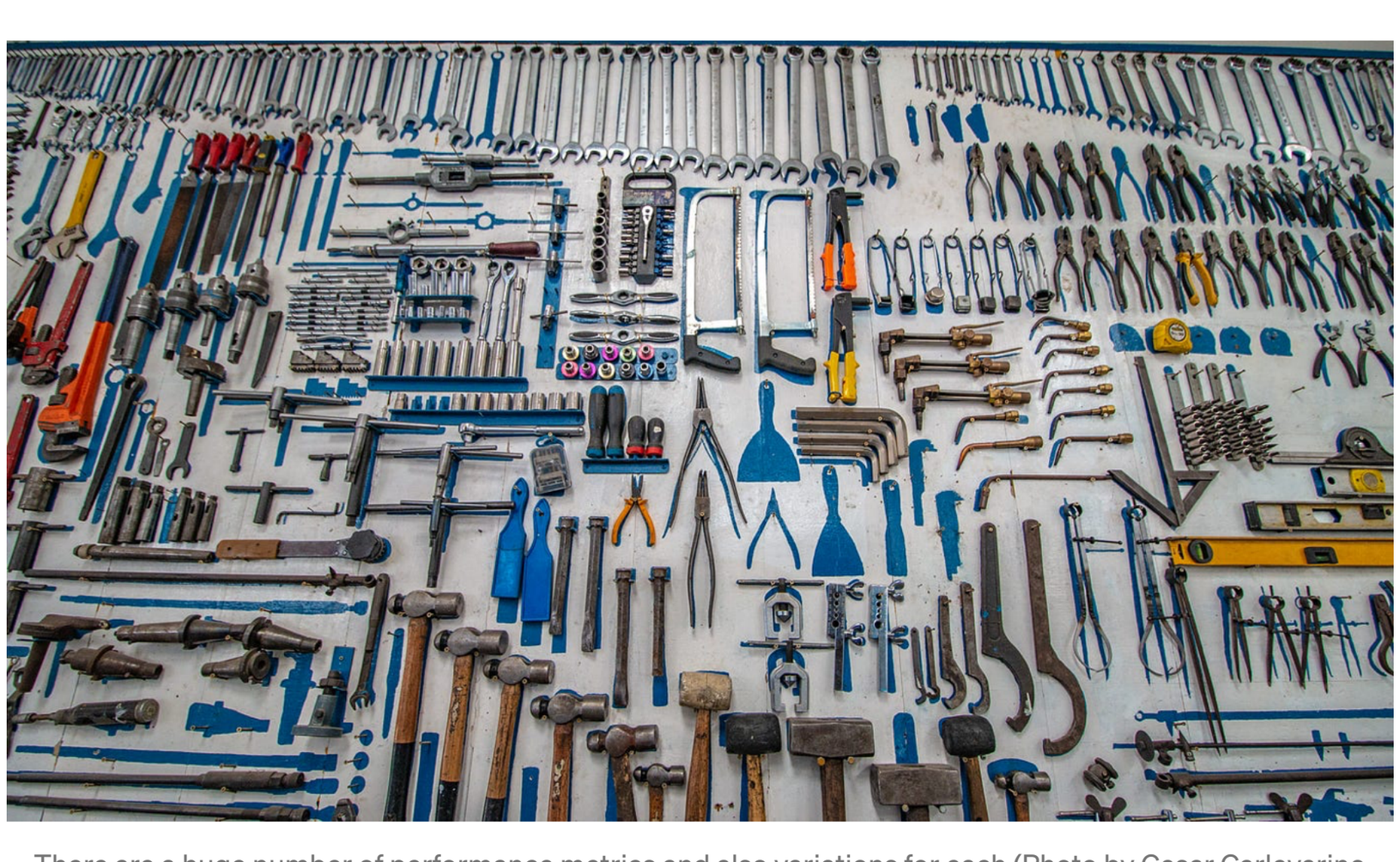
Published in Towards Data Science

Ramya Vidiyala
Jul 26, 2020 · 5 min read · Listen



There are a huge number of performance metrics and also variations for each (Photo by Cesar Carlevarino Aragon on Unsplash)

Performance Metrics for Classification Machine Learning Problems

Accuracy, Precision, Recall, F1 Score, ROC AUC, Log loss

Many learning algorithms have been proposed. It is often valuable to assess the efficacy of an algorithm. In many cases, such assessment is relative, that is, evaluating which of several alternative algorithms is best suited to a specific application.

People even end up creating metrics that suit the application. In this article, we will see some of the most common metrics in a classification setting of a problem.

The most commonly used Performance metrics for classification problem are as follows,

- Accuracy
- Confusion Matrix
- Precision, Recall, and F1 score
- ROC AUC
- Log-loss

. . .

Accuracy

Accuracy is the simple ratio between the number of correctly classified points to the total number of points.

To calculate accuracy, scikit-learn provides a utility function.

```
from sklearn.metrics import accuracy_score

#predicted y values
y_pred = [0, 2, 1, 3]

#actual y values
y_true = [0, 1, 2, 3]

accuracy_score(y_true, y_pred)
0.5
```

Accuracy is simple to calculate but has its own disadvantages.

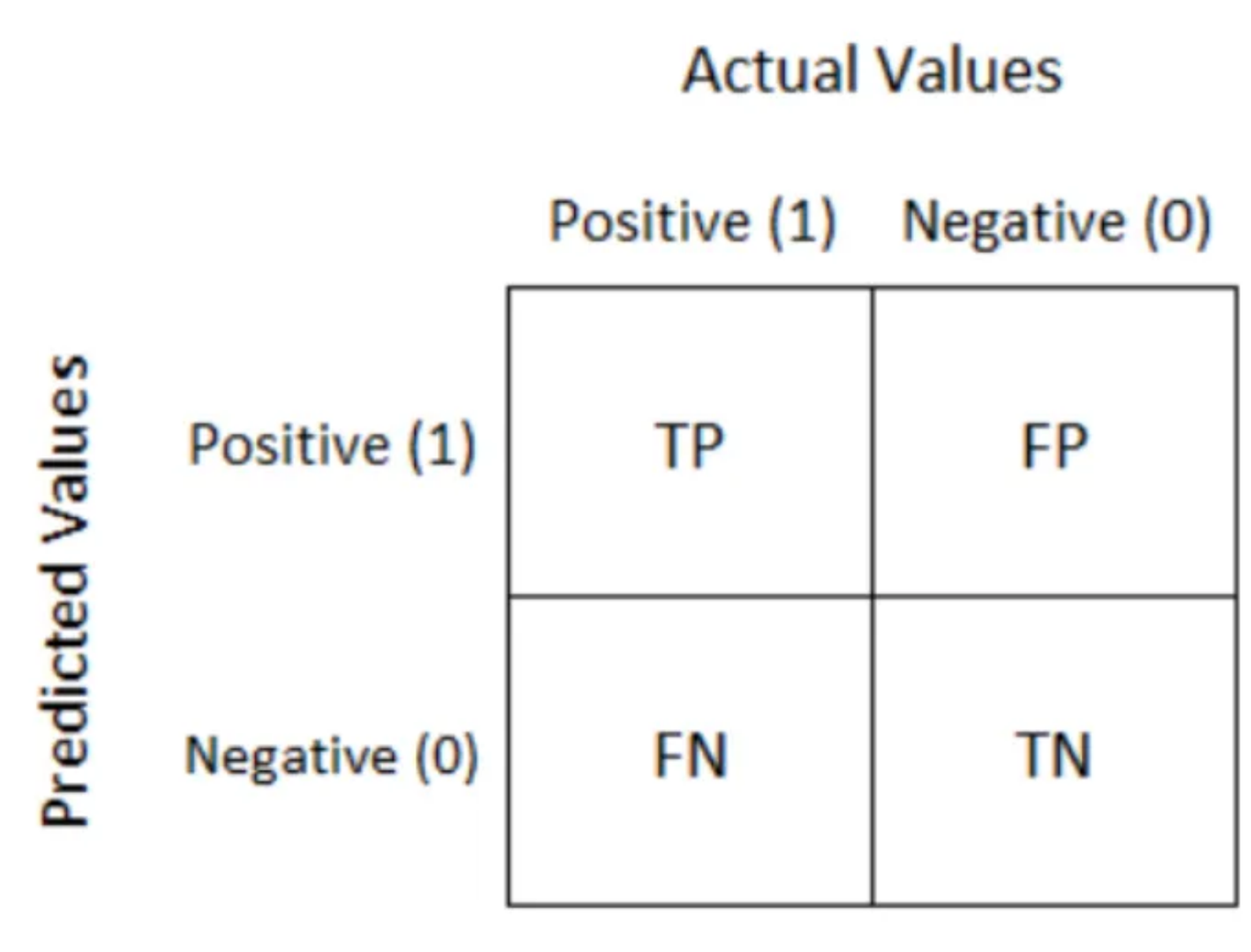
Limitations of accuracy

- If the data set is highly imbalanced, and the model classifies all the data points as the majority class data points, the accuracy will be high. This makes accuracy not a reliable performance metric for imbalanced data.
- From accuracy, the probability of the predictions of the model can be derived. So from accuracy, we can not measure how good the predictions of the model are.

. . .

Confusion Matrix

Confusion Matrix is a summary of predicted results in specific table layout that allows visualization of the performance measure of the machine learning model for a binary classification problem (2 classes) or multi-class classification problem (more than 2 classes)



Confusion matrix of a binary classification

- TP means **True Positive**. It can be interpreted as the model predicted positive class and it is True.
- FP means **False Positive**. It can be interpreted as the model predicted positive class but it is False.
- FN means **False Negative**. It can be interpreted as the model predicted negative class but it is False.
- TN means **True Negative**. It can be interpreted as the model predicted negative class and it is True.

> *For a sensible model, the principal diagonal element values will be high and the off-diagonal element values will be below i.e., TP, TN will be high.*

To get an appropriate example in a real-world problem, consider a diagnostic test that seeks to determine whether a person has a certain disease. A false positive in this case occurs when the person tests positive but does not actually have the disease. A false negative, on the other hand, occurs when the person tests negative, suggesting they are healthy when they actually do have the disease.

For a multi-class classification problem, with 'c' class labels, the confusion matrix will be a (c*c) matrix.

To calculate confusion matrix, sklearn provides a utility function

```
from sklearn.metrics import confusion_matrix
y_true = [2, 0, 2, 2, 0, 1]
y_pred = [0, 0, 2, 2, 0, 2]
confusion_matrix(y_true, y_pred)
array([[2, 0, 0],
       [0, 0, 1],
       [1, 0, 2]])
```

Advantages of a confusion matrix:

- The confusion matrix provides detailed results of the classification.
- Derivates of the confusion matrix are widely used.
- Visual inspection of results can be enhanced by using a heat map.

. . .

Precision, Recall, and F-1 Score

Precision is the fraction of the correctly classified instances from the total classified instances. **Recall** is the fraction of the correctly classified instances from the total classified instances. Precision and recall are given as follows,

$$Precision = \frac{TP}{TP+FP} \qquad Recall = \frac{TP}{TP+FN}$$

Mathematical formula of Precision and Recall using the confusion matrix

For example, consider that a search query results in 30 pages, out of which 20 are relevant. And the results fail to display 40 other relevant results. So the precision is 20/30 and recall is 20/60.

> Precision helps us understand how useful the results are. Recall helps us understand how complete the results are.

But to reduce the checking of pockets twice, the F1 score is used. F1 score is the harmonic mean of precision and recall. It is given as,

$$F1\ score = \frac{2*Precision*Recall}{Precision+Recall}$$

When to use the F1 Score?

- The F-score is often used in the field of information retrieval for measuring search, document classification, and query classification performance.
- The F-score has been widely used in the natural language processing literature, such as the evaluation of named entity recognition and word segmentation.

. . .

Log Loss

Logarithmic loss (or log loss) measures the performance of a classification model where the prediction is a probability value between 0 and 1. Log loss increases as the predicted probability diverge from the actual label. Log loss is a widely used metric for Kaggle competitions.

$$\text{log-loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log p_i + (1-y_i)\log(1-p_i).$$

Here 'N' is the total number of data points in the data set, yi is the actual value of y and pi is the probability of y belonging to the positive class.

> Lower the log-loss value, better are the predictions of the model.

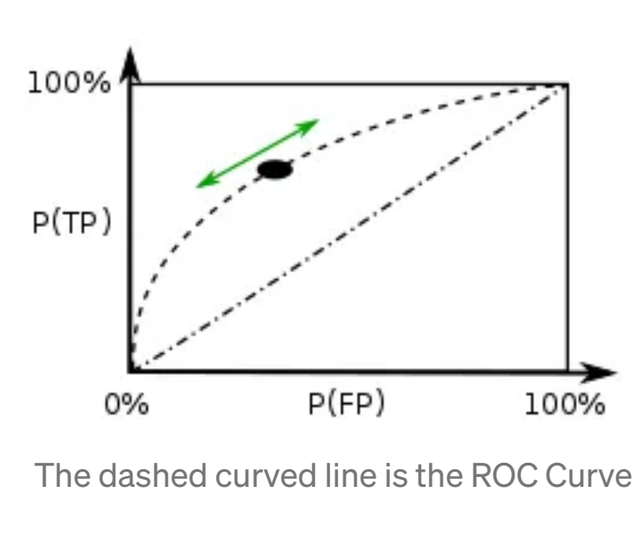
To calculate log-loss, scikit-learn provides a utility function.

```
from sklearn.metrics import log_loss
log_loss(y_true, y_pred)
```

. . .

ROC AUC

A **Receiver Operating Characteristic curve** or **ROC curve** is created by plotting the True Positive (TP) against the False Positive (FP) at various threshold settings. The ROC curve is generated by plotting the cumulative distribution function of the True Positive in the y-axis versus the cumulative distribution function of the False Positive on the x-axis.



The dashed curved line is the ROC Curve

The area under the ROC curve (ROC AUC) is the single-valued metric used for evaluating the performance.

> The higher the AUC, the better the performance of the model at distinguishing between the classes.

In general, an AUC of 0.5 suggests no discrimination, a value between 0.5–0.7 is acceptable and anything above 0.7 is good-to-go-model. However, medical diagnosis models, usually AUC of 0.95 or more is considered to be good-to-go-model.

When to use ROC?

- ROC curves are widely used to compare and evaluate different classification algorithms.
- ROC curve is widely used when the dataset is imbalanced.
- ROC curves are also used in verification of forecasts in meteorology

. . .

Thanks for the read. I am going to write more beginner-friendly posts in the future. Follow me up on Medium to be informed about them. I welcome feedback and can be reached out on Twitter ramya_vidiyala and LinkedIn RamyaVidiyala. Happy learning!

Ramya Vidiyala
447 Followers
Interested in computers and machine learning. Likes to write about it | https://www.linkedin.com/in/ramya-vidiyala/
Follow