Home › Beginner

› Metrics to Evaluate your Classification Model to take the right dec...

S Sumeet Kumar Agrawal
15 Feb, 2024 • 7 min read

Introduction

Evaluation metrics are tied to machine learning tasks. There are different metrics for the tasks of classification and regression. Some metrics, like precision-recall, are useful for multiple tasks. Classification and regression are examples of supervised learning, which constitutes a majority of machine learning applications. Using different Classification metrics for performance evaluation, we should be able to improve our model's overall predictive power before we roll it out for production on unseen data. Without doing a proper evaluation of the Machine Learning model by using different evaluation metrics, and only depending on accuracy, can lead to a problem when the respective model is deployed on unseen data and may end in poor predictions.

In the next section, I'll discuss the Classification Evalution metrics that could help in the generalization of the ML classification model.

Learning Objectives

- Introduction to ML Model Evaluation and its significance.
- Exploration of various evaluation metrics tailored to specific use cases.

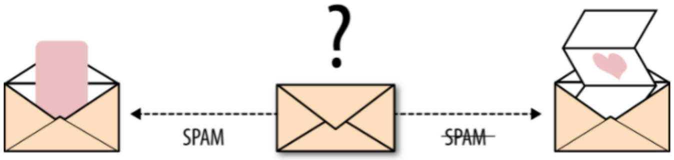
comprehension.

This article was published as a part of the Data Science Blogathon.

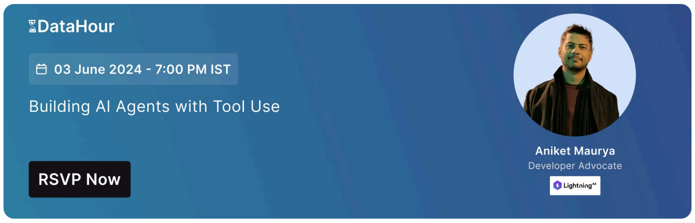
Classification Metrics in Machine Learning

Classification Metrics is about predicting the class labels given input data. In binary classification, there are only two possible output classes(i.e., Dichotomy). In multiclass

A very common example of binary classification is spam detection, where the input data could include the email text and metadata (sender, sending time), and the output label is either *“spam” or “not spam.”* (*See Figure*) Sometimes, people use some other names also for the two classes: “positive” and “negative,” or “class 1” and “class 0.”



Email spam detection is a binary classification problem (source: From Book—Evaluating Machine Learning Model—O’Reilly)



There are many ways for measuring classification performance. Accuracy, confusion matrix, log-loss, and AUC-ROC are some of the most popular metrics.

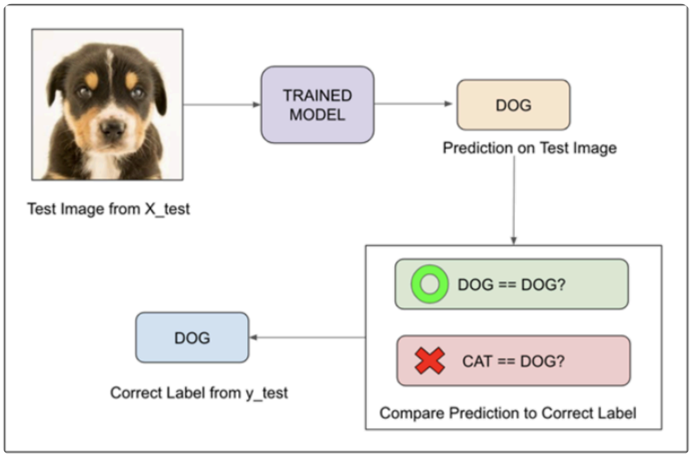
Accuracy simply measures how often the classifier correctly predicts. We can define accuracy as the ratio of the number of correct predictions and the total number of predictions.

$$\textit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

When any model gives an accuracy rate of 99%, you might think that model is performing very good but this is not always true and can be misleading in some situations. I am going to explain this with the help of an example.

Example

Consider a binary classification problem, where a model can achieve only two results, either model gives a **correct** or **incorrect** prediction. Now imagine we have a classification task to predict if an image is a dog or cat as shown in the image. In a supervised learning algorithm, we first **fit/train** a model on training data, then **test** the model on **testing data**. Once we have the model's predictions from the **X_test** data, we compare them to the **true y_values** (the correct labels).



We feed the image of the dog into the training model. Suppose the model predicts that this is a dog, and then compare it to the correct label and it would be incorrect.

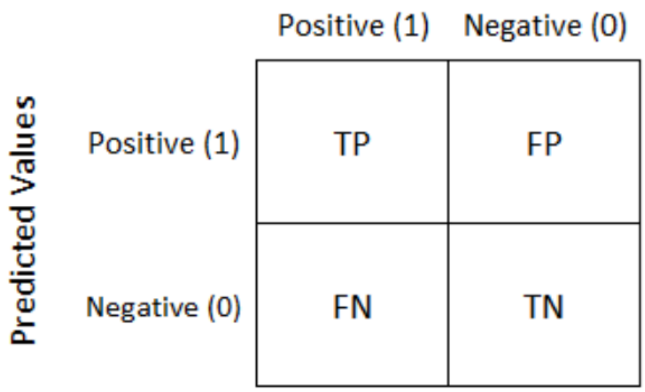
We repeat this process for all images in X_test data. Eventually, we'll have a count of correct and incorrect matches. But in reality, it is very rare that all incorrect or correct matches hold **equal value**. Therefore one metric won't tell the entire story.

Accuracy is useful when the target class is ***well balanced*** but is not a good choice for the unbalanced classes. Imagine the scenario where we had 99 images of the dog and only 1 image of a cat present in our training data. Then our model would always predict the dog, and therefore we got 99% accuracy. In reality, Data is always imbalanced for example Spam email, credit card fraud, and medical diagnosis. Hence, if we want to do a better model evaluation and have a full picture of the model evaluation, other metrics such as recall and precision should also be considered.

Confusion Matrix

Confusion Matrix is a performance measurement for the machine learning classification problems where the output can be two or more classes. It is a table with combinations of predicted and actual values.

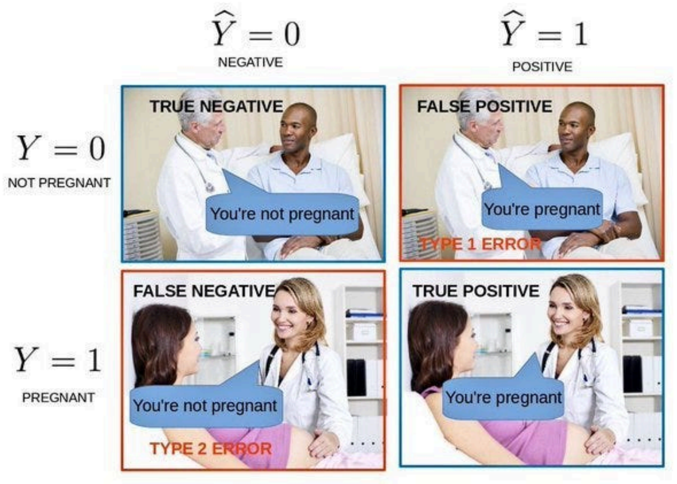
A confusion matrix is defined as thetable that is often used to describe the performance of a classification model on a set of the test data for which the true values are known.



Predicted Values		Positive (1)	Negative (0)
	Positive (1)	TP	FP
	Negative (0)	FN	TN

It is extremely useful for measuring the Recall, Precision, Accuracy, and AUC-ROC curves.

Let's try to understand TP, FP, FN, TN with an example of pregnancy analogy.



- **True Positive:** We predicted positive and it's true. In the image, we predicted that a woman is pregnant and she actually is.
- **True Negative:** We predicted negative and it's true. In the image, we predicted that a man is not pregnant and he actually is not.
- **False Positive (Type 1 Error)**: We predicted positive and it's false. In the image, we predicted that a man is pregnant but he actually is not.
- **False Negative (Type 2 Error)**: We predicted negative

metrics of the confusion matrix

Precision

It explains how many of the correctly predicted cases actually turned out to be positive. Precision is useful in the cases where False Positive is a higher concern than False Negatives. The importance of *Precision is in music or video recommendation systems, e-commerce websites, etc. where wrong results could lead to customer churn and this could be harmful to the business.*

Precision for a label is defined as the number of true positives divided by the number of predicted positives.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Recall (Sensitivity)

It explains how many of the actual positive cases we were able to predict correctly with our model. Recall is a useful metric in cases where False Negative is of higher concern than False Positive. It *is important in medical cases where it doesn't matter whether we raise a false alarm but the actual positive cases should not go undetected!*

Recall for a label is defined as the number of true positives divided by the total number of actual positives.

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

F1 Score

It gives a combined idea about Precision and Recall metrics. It is maximum when Precision is equal to Recall.

F1 Score is the harmonic mean of precision and recall.

$$Precision + Recall$$

The F1 score punishes extreme values more. F1 Score could be an effective evaluation metric in the following cases:

- When FP and FN are equally costly.
- Adding more data doesn't effectively change the outcome
- True Negative is high

AUC-ROC

The Receiver Operator Characteristic (ROC) is a probability curve that plots the TPR(True Positive Rate) against the FPR(False Positive Rate) at various threshold values and separates the 'signal' from the 'noise'.

The **Area Under the Curve (AUC)** is the measure of the ability of a classifier to distinguish between classes. From the graph, we simply say the area of the curve ABDE and the X and Y-axis.

From the graph shown below, the greater the AUC, the better is the performance of the model at different threshold points between positive and negative classes. This simply means that When AUC is equal to 1, the classifier is able to perfectly distinguish between all Positive and Negative class points. When AUC is equal to 0, the classifier would be predicting all Negatives as Positives and vice versa. When AUC is 0.5, the classifier is not able to distinguish between the Positive and Negative classes.

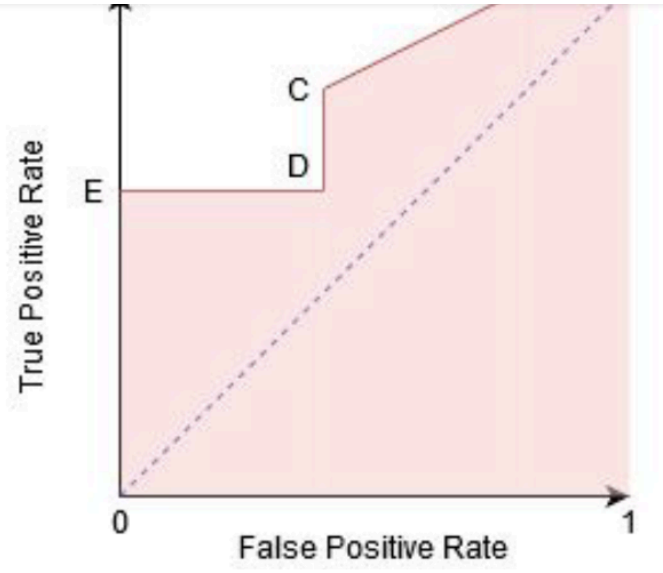


Image Source— https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

Working of AUC

In a ROC curve, the X-axis value shows False Positive Rate (FPR), and Y-axis shows True Positive Rate (TPR). Higher the value of X means higher the number of False Positives(FP) than True Negatives(TN), while a higher Y-axis value indicates a higher number of TP than FN. So, the choice of the threshold depends on the ability to balance between FP and FN.

Log Loss

Log loss (Logistic loss) or Cross-Entropy Loss is one of the major metrics to assess the performance of a classification problem.

For a single sample with true label y∈{0,1} and a probability estimate p=Pr(y=1), the log loss is:

$$logloss_{(\mathrm{N=1})} = y\log(p) + (1-y)\log(1-p)$$

Conclusion

Metrics like accuracy, precision, recall are good ways to evaluate classification models for balanced datasets, but if the data is imbalanced then other methods like ROC/AUC perform better in evaluating the model performance.

ROC curve isn't just a single number but it's a whole curve that provides nuanced details about the behavior of the classifier. It is also hard to quickly compare many ROC curves to each other.

The media shown in this article are not owned by Analytics Vidhya and are used at the Author's discretion.

blogathon

classification model

evaluation metrics

performance metrics for classification

which are evaluation metrics for classification

S

Sumeet Kumar Agrawal

15 Feb 2024

Beginner

Classification

Maths

Statistics

Frequently Asked Questions

Q1. What are the classification metrics?

A. Classification metrics are evaluation measures used to assess the performance of a classification model. Common metrics include accuracy (proportion of correct predictions), precision (true positives over total predicted positives), recall (true positives over total actual positives), F1 score (harmonic mean of precision and recall), and area under the receiver operating characteristic curve (AUC-ROC).

Q2. What are the 4 metrics for evaluating classifier performance?

Responses From Readers

What are your thoughts?...

Submit reply

Barney Darlington
5

Suvojit Hore
9

Company

About Us

Contact Us

Careers

Discover

Blogs

Expert session

Podcasts

Comprehensive Guides

Learn

Free courses

Learning path

BlackBelt program

Gen AI

Engage

Community

Hackathons

Events

Daily challenges

Contribute

Contribute & win

Become a speaker

Become a mentor

Become an instructor

Enterprise

Our offerings

Case studies

Industry report

quexto.ai

Download App

Terms & conditions • Refund Policy • Privacy Policy •
Cookies Policy