2024

Evaluation Metrices for Classification in ML

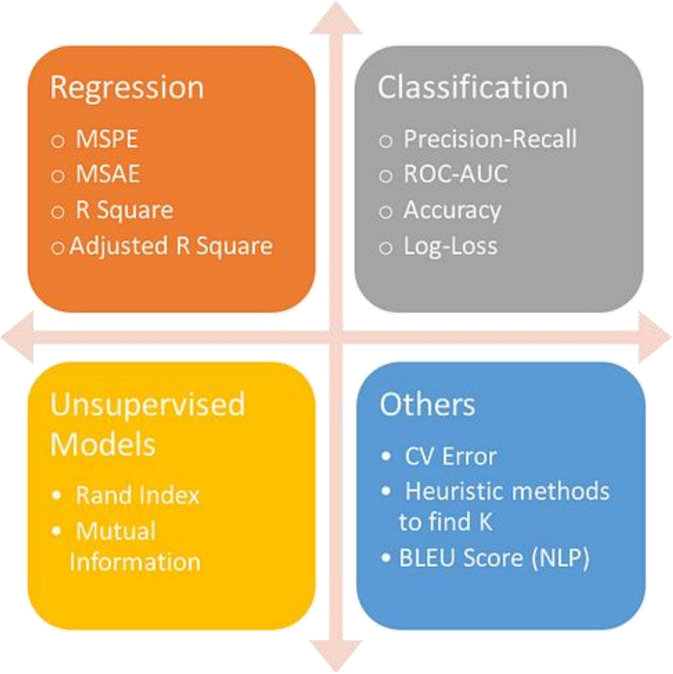
MIP-ML-08

VAIBHAV SINGH

MENTORNESS |

Introduction:

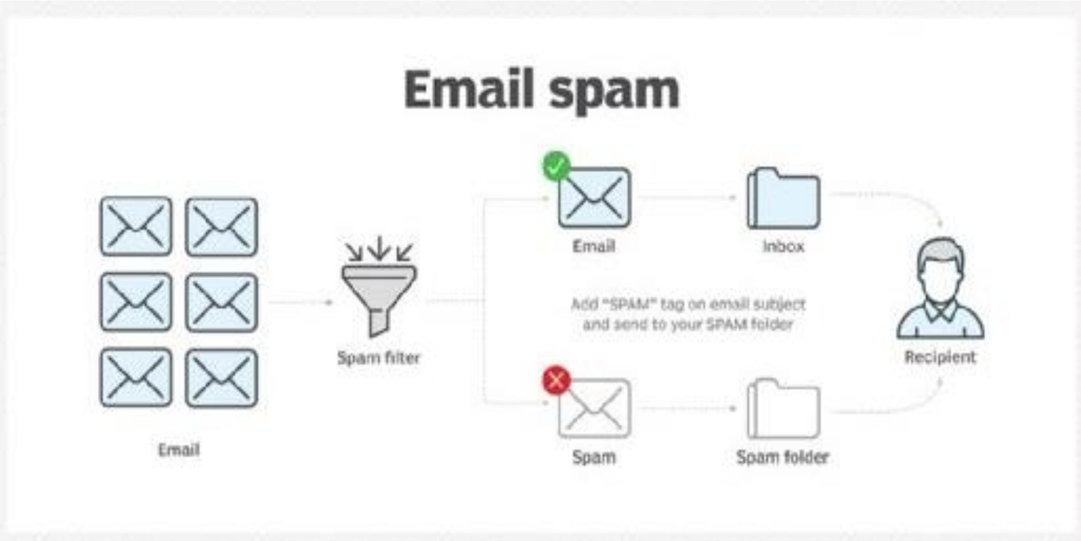
Evaluating the performance of a Machine learning model is one of the important steps while building an effective ML model. *To evaluate the performance or quality of the model, different metrics are used, and these metrics are known as performance metrics or evaluation metrics.* These performance metrics help us understand how well our model has performed for the given data. In this way, we can improve the model's performance by tuning the hyper-parameters. Each ML model aims to generalize well on unseen/new data, and performance metrics help determine how well the model generalizes on the new dataset. 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. 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.



Various Metrices:

In a classification problem, the category or classes of data is identified based on training data. The model learns from the given dataset and then classifies the new data into classes or groups based on the training. It predicts class labels as the output, such as *Yes or No, 0 or 1, Yes or No,* etc. In binary classification, there are only two possible output classes. In multiclass classification, more than two possible classes can be present.

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."* Sometimes, people use some other names also for the two classes: "positive" and "negative," or "class 1" and "class 0."

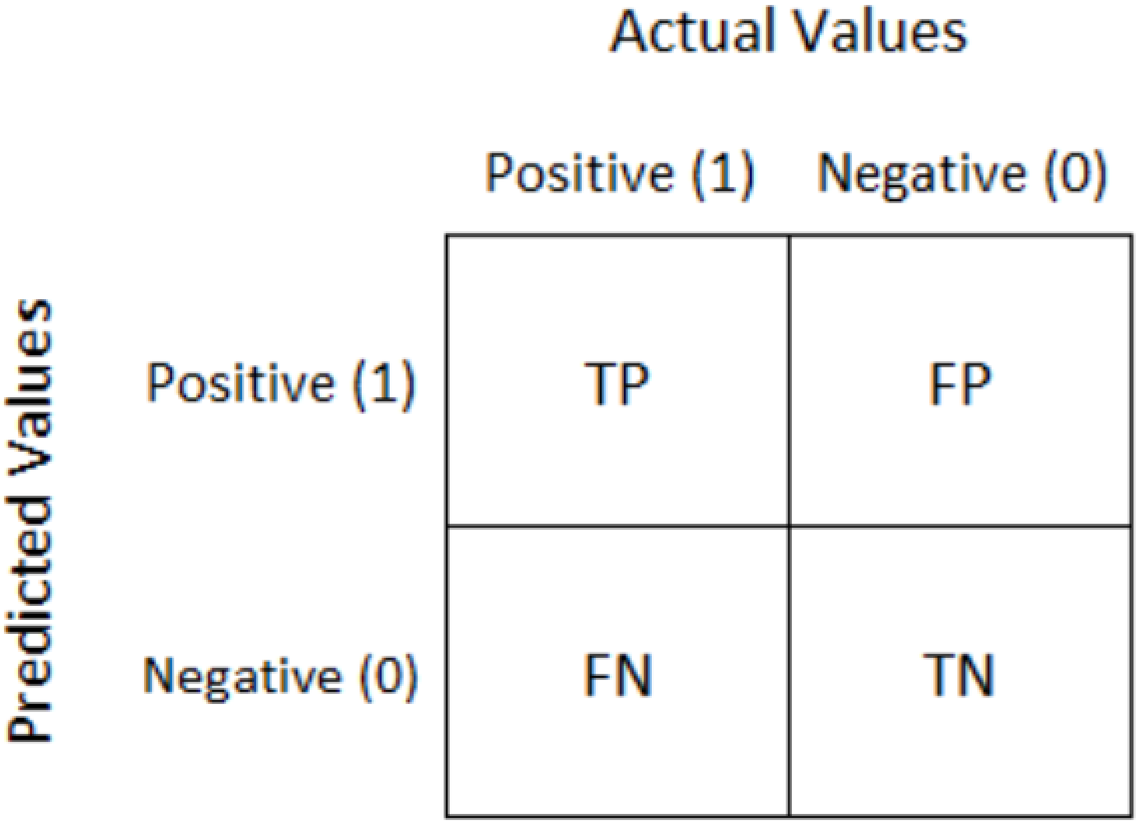

Email spam
Email
Spam filter
Email
Inbox
Add "SPAM" tag on email subject and send to your SPAM folder
Recipient
Spam
Spam folder

There are various ways for measuring classification performance, like:

- Accuracy
- Confusion Matrix
- F1-Score
- Precision-Recall
- AUC(Area under Curve)-ROC

Confusion Matrix:

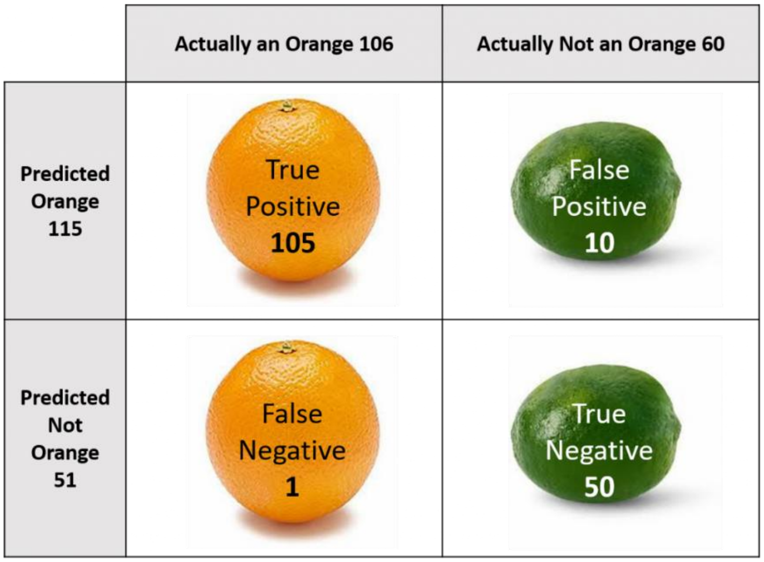
Confusion Matrix is a performance measurement for the machine learning classification problems where the output can be two or more classes. It is a table with combinations of predicted and actual values. It is a tabular representation of prediction outcomes of any binary classifier, which is used to describe the performance of the classification model on a set of test data when true values are known.



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

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

Let's understand confusion matrix with an example of Orange and Lemon:



	Actually an Orange 106	Actually Not an Orange 60
Predicted Orange 115	True Positive **105**	False Positive **10**
Predicted Not Orange 51	False Negative **1**	True Negative **50**

- True Positive: We predicted positive and it’s true. In the image, we predicted that the fruit is an orange and it actually is.
- False Positive (Type 1 Error): We predicted positive and it’s false. In the image, we predicted that the fruit is an orange but it is a lemon.
- False Negative (Type 2 Error): We predicted negative and it’s false. In the image, we predicted that the fruit is not an orange but it actually is.
- True Negative: We predicted negative and it’s true. In the image, we predicted that the fruit is lemon and it’s actually not.

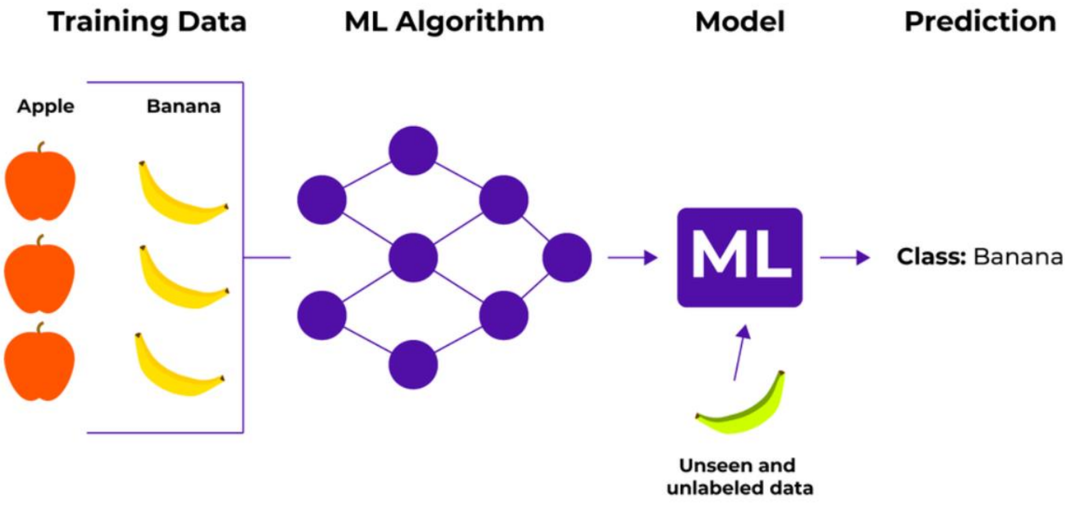
Accuracy:

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.

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

Let's understand accuracy with an 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 apple or banana 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 a banana into the training model. Suppose the model predicts that this is a banana, and then we compare the prediction to the correct label. If the model predicts that this image is a apple and then we again 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.

When to use?

- ► When the target variable is *well balanced.*

When not to use?

- ► When the target is *unbalanced.*

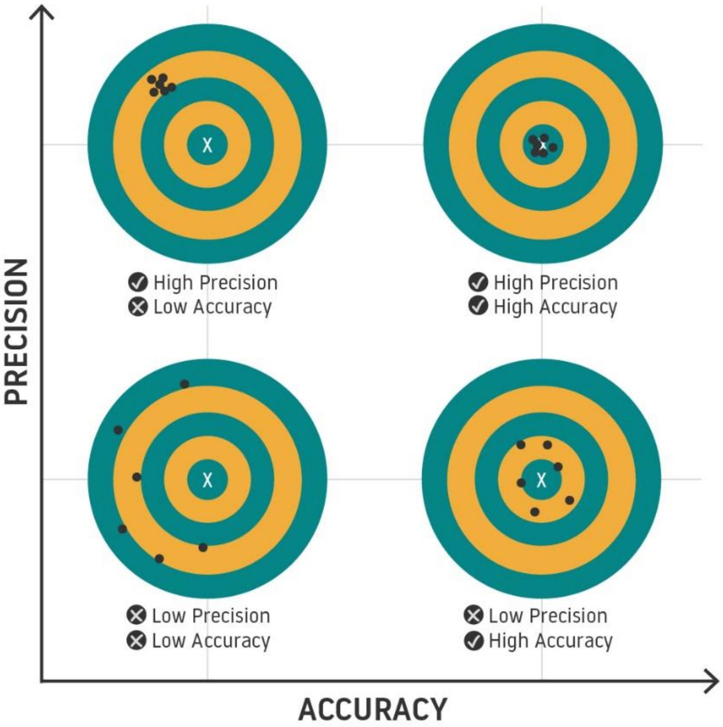
In reality, data is always imbalanced; hence, if we want to do better model evaluation other metrices like precision-recall should also be considered.

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.*

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:

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.

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:

F1 Score is a metric to evaluate a binary classification model on the basis of predictions that are made for the positive class. It is calculated with the help of Precision and Recall. It is a type of single score that represents both Precision and Recall. 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.

$$F1 = 2. \frac{Precision \ \times Recall}{Precision + Recall}$$

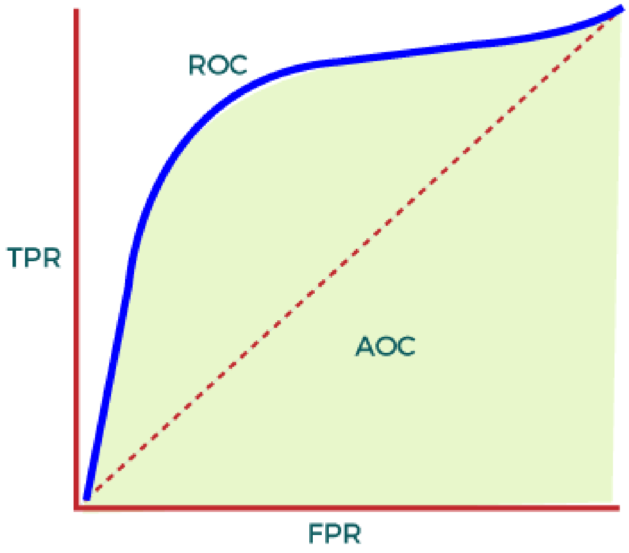
The F1 score punishes extreme values more. F1 Score could be an effective evaluation metric in the following cases:

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

AUC-ROC:

Sometimes we need to visualize the performance of the classification model on charts; then, we can use the AUC-ROC curve. It is one of the popular and important metrics for evaluating the performance of the classification model. The Receiver Operator Characteristic (ROC) is a probability curve that plots the TPR(True Positive Rate) also known as recall 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.



AUC calculates the performance across all the thresholds and provide an aggregate measure. The value of AUC ranges from 0 to 1, it means a model with 100% wrong prediction will have an AUC of 0.0, whereas models with 100% correct predictions will have an AUC of 1.0.

When to use?

- AUC in independent of classification threshold.

When not to use?

- AUC is scale-invariant, which is not always desirable.