EVALUATING THE PERFORMANCE OF MACHINE LEARNING MODELS USING METRICS

Nimmy N .Abraham, R.L. Raheemaa Khan*

ABSTRACT

Machine learning has gained widespread popularity in recent years. It allows us to draw inferences about new situations using previous data, and there are a wide range of algorithms available for this purpose. Popular machine learning algorithms used in the modern day world include linear regression, naïve Bayes, random forests, logistic regression and also some others like k-means clustering and decision trees. When making predictions with machine learning, we often try out multiple algorithms to determine which one produces the most accurate results on the data.

Keywords: Evaluation Metrics, Confusion Matrix, Prediction Score, Classification

I. INTRODUCTION

Regression and classification are the two primary types of models used in machine learning. To make sure a machine learning model is efficient, it is crucial to assess its performance[1]. A model's quality can be assessed using a variety of metrics, commonly referred to as performance or evaluation metrics. These metrics give us insight into the model's performance on a specific dataset and can be used to tweak hyper parameters to enhance the model's performance. Any machine learning model should be able to generalize well to new data, and performance metrics can show us how well a model can does this.

An important phase in the machine learning process is assessing how well a trained machine learning model is doing [2]. A major aspect in assessing whether a model is adaptive or non-adaptive is its capacity to generalize to data that have not yet been observed. Before deploying the model for production on new data, we may improve its predictive capability by evaluating its performance using a variety of criteria. When the model is applied to unobserved data, making bad predictions might result from failing to adequately analyze the model using a variety of metrics and relying exclusively on accuracy. A lack of generalization to new data can result from the model memorizing the training data rather than learning from it.

Department of Computer Science
Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India
*Corresponding Author

II. MODEL EVALUATION METRICS

Model evaluation metrics have the primary aim of measuring the level of effective completion of a machine learning model on a given data file. These metrics provide a quantitative measure of the model's potentiality to make precise forecasts and can help to identify areas where the model may be lacking. There are many different evaluation metrics that can be used, depending on the essence of the data and the objectives of the analysis.

1. Confusion Matrix

The standard table used to assess a classification model's efficacy is a confusion matrix[3]. It gives a summary of how the model's predictions performed in comparison to the actual results of the data (or labels).

Usually, a binary classification problem is visualized using the confusion matrix (i.e., a problem with two possible outcomes). In this case, the table has two rows and two columns, representing the two possible outcomes of the classification problem (e.g., "positive" and “Negative”).

The table below demonstrates a confusion matrix for a binary classification problem

	Predicted Positive	Predicted Negative
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

Table1. Confusion matrix for a binary classification problem

The top left to bottom right entries in this table reflect correct predictions, whereas the entries that are off-diagonal represent incorrect guesses.

"True positive," "True negative," "false positive," and "false negative" are the four results of a binary classification problem:

True positive: The model successfully anticipated the favorable result (e.g., The affected person is ill).

True negative: The model properly foresaw the unfavorable result (e.g., The affected person doesn't have ill).[4,5,6]

False positive: The model anticipated the positive result in error (e.g., the model predicted the patient has the disease, but they do not).

False negative: The model erroneously forecast a negative result (e.g., the model predicted the patient does not have the disease, but they do).

We can assess the model's overall performance as well as its capacity to accurately categorise various types of outcomes by taking a look at the values in the confusion matrix. For instance, if the model is being used to predict an illness, a significant number of true positives and a limited amount of false negatives may be more crucial than the number of false positives.

Some Common evaluation matrix used in machine learning are as follows:

The percentage of accurate predictions provided by a categorization model serves as a gauge of its overall performance. [7, 8] Accuracy in a confusion matrix is calculated as the sum of true positive (TP) and true negative (TN) judgments divided by the total number of predictions [9,10,11,12]. Here is the formula for calculating accuracy from a confusion matrix:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$ [13,14]

For example, suppose you have the following confusion matrix for a binary classification model:

	Predicted Positive	Predicted Negative
Actual Positive	8	2
Actual Negative	3	7

Table2. Confusion matrix

Using the formula above, the accuracy of the model can be calculated as:

$$\text{Accuracy} = (8 + 7) / (8 + 2 + 3 + 7) = 0.75$$

This means that the model made correct predictions 75% of the time

2. Precision

A classification model's precision, which is calculated as the percentage of accurate predictions the model makes, is measured in terms of how well it performs. Precision in a confusion matrix is determined by dividing the total number of positive predictions by the actual number of positive predictions.

The formula for determining precision from a confusion matrix is given below:

$$\text{Precision} = TP / (TP + FP)$$

For example, suppose you have the following confusion matrix for a binary classification model:

	Predicted Positive	Predicted Negative
Actual Positive	8	2
Actual Negative	3	7

Table3. Confusion matrix for precision

Using the formula above, the precision of the model can be calculated as:

$$\text{Precision} = 8 / (8 + 3) = 0.73$$

This indicates that 73% of the model's positive predictions were accurate.

3. Sensitivity

Sensitivity is a statistic used to evaluate how effectively a classification model performs. In machine learning, it is often referred to as recall or true positive rate. The percentage of actual positive cases that the model accurately anticipated is how it is determined. By dividing the total number of true positive cases by the total number of true positive forecasts, one can calculate the sensitivity of a confusion matrix.

The following formula can be used to determine sensitivity from a confusion matrix:

$$\text{Sensitivity} = TP / (TP + FN)$$

For example, suppose you have the following confusion matrix for a binary classification model:

	Predicted Positive	Predicted Negative
Actual Positive	8	2
Actual Negative	3	7

Table4. Confusion matrix for sensitivity

Using the formula above, the sensitivity of the model can be calculated as:

Sensitivity = 8 / (8 + 2) = 0.80

This means that the model correctly predicted 80% of the actual positive cases.

Sensitivity is frequently used to assess the model's capacity to identify positive cases, especially when it is crucial to reduce false negatives (e.g. in medical diagnosis).

4. F1 Score

The F1 score is a metric that is used to evaluate the performance of a classification model[15]. It is calculated as the harmonic mean of the model's recall and precision, where recall is the proportion of true positive predictions made by the model among all positive predictions and precision is the proportion of true positive predictions made by the model among all instances of true positive predictions[13,16].

Because it integrates precision and recall into one score, the F1 score is frequently used as a single statistic to assess the effectiveness of a classification model[14,17].

The F1 score is defined as follows:

F1 = 2 * (precision * recall) / (precision + recall)

Higher F1 scores, which range from 0 to 1, indicate superior performance. If the model had an F1 score of 1, it would have flawless precision and recall.

Depending on the particulars of the problem, it could sometimes be more crucial to optimize for either precision or recall. For instance, it may be more crucial to optimize for recall in a work of medical diagnosis (i.e., to ensure that the model does not miss any real positive cases), but it may be more crucial to optimize for precision in a task of fraud detection (i.e., to ensure that the model only produces a small number of false positives).

For example, suppose the model makes the following predictions on a test set of 1000 patients:

True positive: 500 patients

True negative: 400 patients

False positive: 50 patients

False negative: 50 patients

The precision of the model would be 500 / (500 + 50) = 0.91, and the recall of the model would be 500 / (500 + 50) = 0.91.

The F1 score would then be calculated as:

F1 = 2 * (0.91 * 0.91) / (0.91 + 0.91) = 0.91

This would indicate that the model has good performance, with a high F1 score of 0.91.

It's important to note that in a medical diagnosis setting, it may be more important to optimize for recall (to ensure that the model does not miss any true positive cases), rather than precision. Depending on the specifics of the task at hand, it may occasionally be more crucial to optimize for either precision or recall. For instance, it may be more crucial to optimize for recall in a medical diagnostic work (i.e., to make sure the model does not miss any true positive cases) than it may be to optimize for precision in a fraud detection task (i.e., to ensure that the model only produces a small number of false positives).

5. Specificity

Specificity is a metric that is used to evaluate the performance of a classification model, particularly in the context of medical diagnosis or other situations where false positives are costly[8][14]. It is defined as the fraction of actual negative instances that are correctly predicted by the model out of all negative predictions made by the model.

For example, consider a classification model that is trying to diagnose a particular disease based on certain symptoms. Specificity would be calculated as the number of patients who do not have the disease and are correctly diagnosed by the model as not having the disease, divided by the total number of patients who are diagnosed by the model as not having the disease.

Suppose the model makes the following predictions on a test set of 1000 patients:

True positive: 500 patients

True negative: 400 patients
False positive: 50 patients
False negative: 50 patients
The specificity of the model would be calculated as 400 / (400 + 50) = 0.89. This means that out of all the patients who were predicted by the model to not have the disease, 89% were actually negative cases.

6. ROC Curve

A binary classification model's effectiveness is graphically depicted by a receiver operating characteristic (ROC) curve[5]. It contrasts the genuine positive rate (sensitivity or recall) on the y-axis with the false positive rate (1 - specificity) on the x-axis[15]. When you change the threshold for identifying occurrences as positive or negative, the ROC curve makes it possible to see how the trade-off between sensitivity and specificity is affected[13].

An illustration of how to create a ROC curve for a classification model is given below:

To create the ROC curve, you must first figure out the true positive rate (sensitivity or recall) and false positive rate at various thresholds. For example, if you set the threshold at 0.5, patients will be classified as testing positive or negative based on projected probabilities that are more than or equal to 0.5.

With this threshold, the genuine positive rate and false positive rate can be calculated in the following ways:

True positive rate (TPR) = TP / (TP + FN) [6]
False positive rate (FPR) = FP / (FP + TN)

Let's say the model correctly predicts the following outcomes for a test population of 1000 patients:

True positive: 500 patients
True negative: 400 patients
False positive: 50 patients
False negative: 50 patients

Given these numbers, the following formula is used to get the true positive rate and false positive rate at a threshold of 0.5:

True positive rate (TPR) = 500 / (500 + 50) = 0.91
False positive rate (FPR) = 50 / (50 + 400) = 0.11

With these numbers, the ROC curve's point (0.11, 0.91) may be plotted. To add more points to the ROC curve, repeat this procedure at various levels.

7. Log Loss

A classification model's performance is measured using a statistic called log loss, also known as cross-entropy loss[16]. The mean log loss for all test set samples is calculated by using the negative log likelihood of the actual labels, given the predicted probabilities.

This is how the log loss is defined:

$$\text{Log loss} = -(1/N) * \sum (y \log(p) + (1 - y) \log(1 - p))$$

where:

In the test set, N represents the total number of samples.
The true label here is y (either 0 or 1)
p depicts the predicted probability of the positive class

The log loss ranges from 0 to ∞, with lower values indicating better performance. A log loss of 0 indicates perfect performance, while a log loss of 1 indicates that the model is predicting the opposite of the true labels.

Because it considers both the real labels and the predicted probabilities, log loss is frequently employed as a single statistic to assess the effectiveness of a classification model. It is useful metric when the costs of false positives and false negatives are not equal, as it penalizes confident but incorrect predictions more heavily than uncertain predictions. This characteristic makes it particularly useful in situations where accurately identifying false positives and false negatives is critical.

To calculate the log loss for a model that is trying to diagnose a skin disease, you would need to consider the true labels and predicted probabilities of the instances in a test dataset. The log loss is calculated as the negative logarithm of the likelihood of the model's predictions, given the true labels of the instances.

Here is an example of how to calculate the log loss for a model that is trying to diagnose a particular skin disease based on certain symptoms:

Suppose you have a classification model that is trying to diagnose a particular skin disease based on certain symptoms. The model makes the following predictions on a test set of 1000 patients:

Patient ID	True Label	Predicted Probability
1	0	0.1
2	0	0.9
3	1	0.7
4	1	0.3
...	...	...
1000	0	0.2

Table5. Predicted Probability

The log loss for the model is calculated as follows:
Log loss = -(1/1000) * Σ[y * log(p) + (1 - y) * log(1 - p)]
= -(1/1000) * (0 * log(0.1) + 1 * log(0.9) + 1 * log(0.7) + 0 * log(0.3) + ... + 0 * log(0.2))
= 0.45
This indicates that the model has a log loss of 0.45, which is a relatively high value. This may indicate that the model is not making very accurate predictions, or that the predicted probabilities are not well calibrated to the true labels.
It's important to note that the log loss metric is sensitive to the predicted probabilities, and can be affected by imbalances in the classes. In situations where the classes are excessive, the log loss metric may be particularly useful for evaluating the performance of the model.

8. Jaccard coefficient

The Jaccard coefficient is beneficial to measure the similarity between two sets[17]. You need to build a confusion matrix first before you can calculate the Jaccard coefficient. A confusion matrix is a table that contrasts a diagnostic test's expected results with the actual outcomes. The matrix's columns indicate the expected results, while the rows show the actual findings. The matrix contains values that indicate the number of times a particular outcome was accurately or inaccurately predicted by the diagnostic test..
You must use the following formula to determine the Jaccard coefficient:
Jaccard coefficient = (true positive) / (true positive + false positive + false negative)
The Jaccard coefficient is a measure of the degree of agreement between expected and observed results. It runs from 0 to 1, with 1 representing perfect agreement. There was no overlap between the expected and actual results, as shown by a value of 0.
For example, consider the following confusion matrix for a diagnostic test for a viral disease:

	Predicted Positive	Predicted Negative
Actual Positive	10	5
Actual Negative	2	8

Table6. Confusion matrix for a diagnostic test

To calculate the Jaccard coefficient, we would use the following formula:
Jaccard coefficient = (10) / (10 + 5 + 2) = 0.67
This indicates that the diagnostic test has a moderate level of agreement with the actual results, but is not perfectly accurate.

9. Gain and Lift Chart

Gain and lift charts can be used to assess how well a classification model categorizes a skin condition[10]. To create a gain chart, you would first need to divide the target population (patients with the skin disease) into 10 equal-sized groups (deciles) based on the predicted probability of the skin disease. The gain at each decile is then calculated as the percentage of the target population that is correctly identified by the model at that decile, relative to the overall percentage of the target population in the dataset.
Here is an example of a gain chart for a classification model that is designed to identify patients with a particular skin disease:

Decile	Percentage of Target Population	Percentage Identified by Model
1 (top)	10%	80%
2	10%	70%
3	10%	60%
4	10%	50%
5	10%	40%
6	10%	30%
7	10%	20%
8	10%	10%
9	10%	5%
10 (bottom)	10%	0%

Table7. Classification model

This gain chart shows that the model is able to correctly identify a large percentage of the target population at the top deciles, but its performance decreases as we move down the chart. This suggests that the model is generally effective at identifying patients with the skin disease, but may have some limitations.
To create a lift chart, you would follow a similar process, but instead of showing the percentage of the target population identified at each decile, you would show the ratio of this

percentage to the overall percentage of the target population in the dataset. This allows you to see how much "lift" the model is providing at each decile, or how much better it is at identifying the target population compared to randomly selecting from the entire dataset.

Here is an example of a lift chart for the same classification model:

Decile	Lift (Relative to Random Selection)
1 (top)	8
2	7
3	6
4	5
5	4
6	3
7	2
8	1
9	0.5
10 (bottom)	0

Table8. Lift chart

This lift chart shows that the model is providing a significant amount of lift at the top deciles, but the lift decreases as we move down the chart. This indicates that the model is generally effective at identifying patients with the skin disease, but may not be as accurate at identifying patients at the lower deciles.

III. CONCLUSION

The specific properties of the data and the objectives of the machine learning model will determine which evaluation metric is best to employ. For example, accuracy is a simple and widely used metric that is appropriate for many types of classification tasks, but it may not be the most informative metric for imbalanced datasets or for tasks where the cost of false negatives or false positives is high. In these cases, precision, recall, or F1 score may be more appropriate.

It's important to precisely consider which evaluation standard is most applicable for your machine learning task, and to use multiple evaluation criteria to get a more complete understanding of the model's performance. It's also important to keep in mind that no single evaluation standard can capture all aspects of a model's performance, and that different evaluation criteria may give disagreeing or deficient information.

Choosing the right evaluation metric is essential for accurately analyzing the model's capabilities and pinpointing areas for improvement. Evaluation metrics are a useful tool for determining the performance of a machine learning model.

REFERENCES

1. Dastile, Xolani, Turgay Celik, and Moshe Potsane. "Statistical and machine learning models in credit scoring: A systematic literature survey." Applied Soft Computing 91 (2020): 106263.

2. Dalianis, Hercules. "Evaluation metrics and evaluation." Clinical text mining. Springer, Cham, 2018. 45-53.

3. Ahammed, Mostafiz, Md Al Mamun, and Mohammad Shorif Uddin. "A machine learning approach for skin disease detection and classification using image segmentation." Healthcare Analytics 2 (2022): 100122.

4. Tharwat A. (August 2018). "Classification assessment methods". Applied Computing and Informatics. doi:10.1016/j.aci.2018.08.003

5. Powers, David M. W. (2011). "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation". Journal of Machine Learning Technologies. 2 (1): 37–63.

6. Fawcett, Tom (2006). "An Introduction to ROC Analysis" (PDF). Pattern Recognition Letters. 27 (8): 861–874. doi:10.1016/j.patrec.2005.10.010.

7. Powers, David M. W., Recall and Precision versus the Bookmaker, Proceedings of the International Conference on Cognitive Science (ICSC-2003), Sydney Australia, 2003, pp. 529-534

8. Ravi Manne(2020) Classification of Skin cancer using deep learning, ConvolutionalNeural Networks -

Opportunities and vulnerabilities- A systematic Review: International Journal for Modern Trends in Science and Technology, 6(11): 101-108, 2020 DOI: https://doi.org/10.46501/IJMTST061118

9. https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826

10. G Shmueli (2019). "LIFT UP AND ACT! CLASSIFIER PERFORMANCE IN RESOURCE-CONSTRAINED APPLICATIONS" (PDF) https://arxiv.org/pdf/1906.03374

11. Ali, Najat, Daniel Neagu, and Paul Trundle. "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets." SN Applied Sciences 1.12 (2019): 1-15.

12. Mandrekar, Jayawant N. "Receiver operating characteristic curve in diagnostic test assessment." Journal of Thoracic Oncology 5.9 (2010): 1315-1316.

13. Sokolova, Marina, Nathalie Japkowicz, and Stan Szpakowicz. "Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation." Australasian joint conference on artificial intelligence. Springer, Berlin, Heidelberg, 2006.

14. Hossin, Mohammad, and Md Nasir Sulaiman. "A review on evaluation metrics for data classification evaluations." International journal of data mining & knowledge management process 5.2 (2015): 1.

15. Le CT. A solution for the most basic optimization problem associated with an ROC curve. Stat Methods Med Res. 2006 Dec;15(6):571-84. doi: 10.1177/0962280206070637. PMID: 17260924.

16. Vovk, Vladimir. (2015). The Fundamental Nature of the Log Loss Function. 10.1007/978-3-319-23534-9_20.

17. Chahal, Manoj. (2016). Information Retrieval using Jaccard Similarity Coefficient. International Journal of Computer Trends and Technology. 36. 140-143. 10.14445/22312803/IJCTT-V36P124.