CELEBAL TECHNOLOGY

PRESENTATION

Group Members

1. Jawale Aniket Shrikrishna
2. Abhale Sonali Prakash
3. Patil Aniruddha Ajay
4. Jadhav Vishal Valmik
5. Zaware Rahul Raosaheb
6. Dhage Prashant Shivnath
7. Khalekar Shreyas Dipak
8. Daware Sanket Sanjay

Assignment Title:

Performance Metrics

1. **Confusion Matrix**
2. **Receive Operating Curve**
3. **Area Under Curve**
4. **Log Loss**

INTRODUCTION

We can evaluate the performance of ML algorithms, classification algorithms, and regression algorithms using a variety of measures. Because the measures we use to evaluate ML performance must be carefully chosen,

The metric you choose will determine how the performance of ML algorithms is measured and compared.

The metric you choose will have a huge impact on how you weight the relevance of various attributes in the end outcome.

The following are some performance metrics that can be used to evaluate classification problem predictions:

Note:

Classification and Regression is a supervised learning model that is used to categorise and predict labelled data.

PERFORMANCE METRICS

There are numerous metrics that can be used to assess the performance of ML algorithms, classification algorithms, and regression algorithms. We must choose the metrics for evaluating ML performance with care because

- The metric you choose will entirely determine how the performance of ML algorithms is measured and compared.
- The metric you choose will completely influence how you weight the importance of various characteristics in the result.

performance metrics that can be used to evaluate predictions for classification problems are as follows:

Confusion Matrix

Have you been in a situation where you expected your machine learning model to perform really well but it has given a poor accuracy? You've done all the hard work – so where did the classification model go wrong? How can you correct this?

There are plenty of ways to gauge the performance of your classification model but none have stood the test of time like the confusion matrix.

It is the easiest way to measure the performance of a classification problem where the output can be of two or more type of classes. A confusion matrix is nothing but a table with two dimensions viz. “Actual” and “Predicted” and furthermore, both the dimensions have “True Positives (TP)”, “True Negatives (TN)”, “False Positives (FP)”, “False Negatives (FN)” as shown below −

	Predicted Negative	**Predicted Positive**
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

Explanation of the terms associated with confusion matrix are as follows −

- **True Positives (TP)** − It is the case when both actual class & predicted class of data point is 1.
- **True Negatives (TN)** − It is the case when both actual class & predicted class of data point is 0.
- **False Positives (FP)** − It is the case when actual class of data point is 0 & predicted class of data point is 1.
- **False Negatives (FN)** − It is the case when actual class of data point is 1 & predicted class of data point is 0.

We can use confusion matrix function of sklearn.metrics to compute Confusion Matrix of our classification model.

Accuracy for the matrix can be calculated by taking average of the values lying across the **“main diagonal”** i.e.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Sample}}$$

ROC (Receiver Operating characteristic)

The Receiver Operating characteristic (ROC) curve is explicitly used for binary classification. However, it can be extended for multiclass classification.

In binary classification, when a model gives probability scores as output, we use 0.5 as a threshold for the simplest model. If the probability of a query point is greater than 0.5, the model will classify it as class 1 (say positive) otherwise class 0 (negative). To measure the performance of the model, we can use accuracy, confusion matrix, precision, recall, and F1 score.

A question that arises here is using 0.5 as a threshold will give significant results for every model?

NO

Selecting a threshold that produces significant results is problem specific. For example, in cancer detection problems, if we keep a low threshold, more people will be predicted as positive by the model. Moreover, if we set an incredibly high threshold, there is a possibility of missing actual patients.

How should we decide the appropriate threshold value?

ROC is one technique that can give the best threshold value.
STEPS:

1. Take unique probability scores (in descending order) as a threshold and predict the class labels. If we have k unique probability scores, there will be k thresholds.
2. For each threshold, we measure the class labels for all the query points.

For each prediction, we calculate the true positive rate (TPR) and false positive rate (FPR).

Formulae:

- TPR=TP/Actual Position=TP/TP+FN
- FPR=FP/Actual Negative =FP/TN+FP

The appropriate threshold will be the point where TPR is maximum, and FPR is minimum because we want true positive and true negative to be more than a false positive and false negative.

AUC (Area Under curve)

Area Under Curve (AUC) is one of the most widely used metrics for evaluation. It is also mainly used for binary classification problems. The AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example.

The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The maximum value of AUC can be 1. Because the maximum value of TPR and FPR is 1, respectively. So, the maximum area will be 1.

When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative class points correctly. If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives, and all Positives as Negatives.

When 0.5

When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points. Meaning either the classifier is predicting a random class or a constant class for all the data points.

So, the higher the AUC value for a classifier, the better its ability to distinguish between positive and negative classes.

We can use roc_auc_score function of sklearn.metrics to compute AUC-ROC.A **limitation** of AUC scores is they are **sensitive to imbalanced data.** If the model is dumb, AUC scores can be high.

LOGLOSS (Logarithmic Loss)

Log Loss is one of the most important classification metrics based on probabilities. It’s hard to interpret raw log-loss values, but log-loss is still a good metric for comparing models. For any given problem, a lower log loss value means better predictions. Log Loss is the negative average of the log of corrected predicted probabilities for each instance.

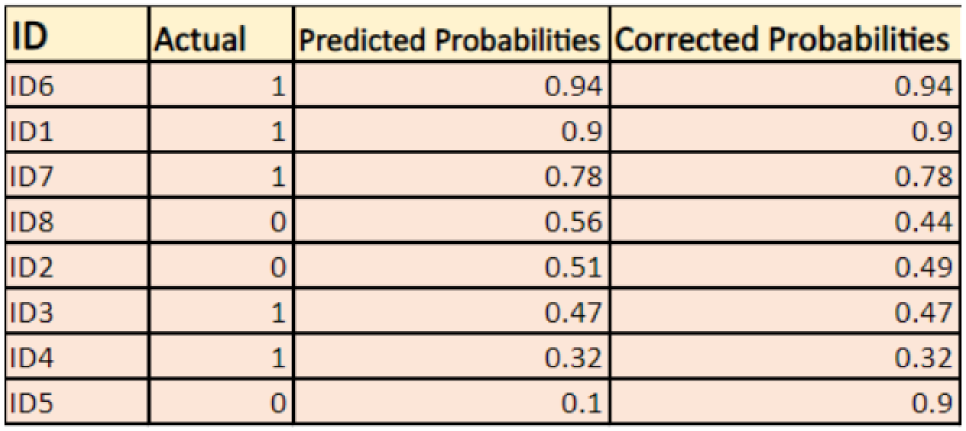
There are three steps to find Log Loss:

1. To find corrected probabilities.
2. Take a log of corrected probabilities.
3. Take the negative average of the values we get in the 2nd step.

Let us understand it with an example:

1.To find corrected probabilities.

. The model is giving predicted probabilities and corrected probabilities as shown below: -

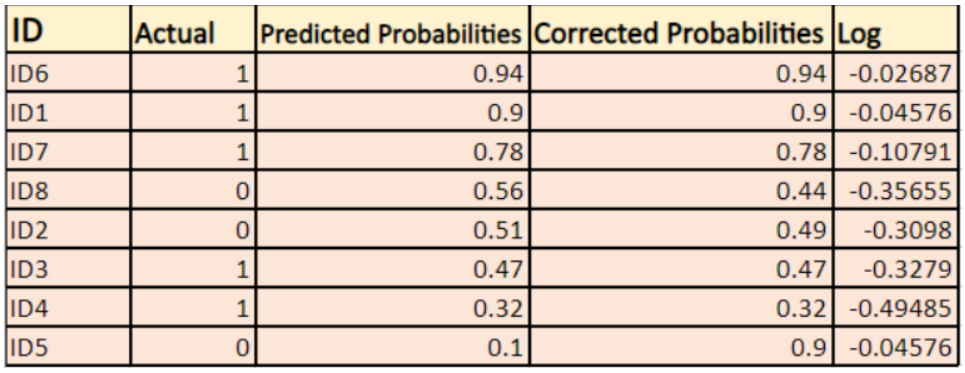


ID	Actual	Predicted Probabilities	Corrected Probabilities
ID6	1	0.94	0.94
ID1	1	0.9	0.9
ID7	1	0.78	0.78
ID8	0	0.56	0.44
ID2	0	0.51	0.49
ID3	1	0.47	0.47
ID4	1	0.32	0.32
ID5	0	0.1	0.9

For finding out corrected probabilities, let us consider one example -The probability that a person with ID5 will buy a jacket (i.e., belong to class 1) is 0.1 but the actual class for ID5 is 0, so the probability for the class is (1-0.1) =0.9. 0.9 is the correct probability for ID5.

2.Take a log of corrected probabilities.

We will find a log of corrected probabilities for each instance.



ID	Actual	Predicted Probabilities	Corrected Probabilities	Log
ID6	1	0.94	0.94	-0.02687
ID1	1	0.9	0.9	-0.04576
ID7	1	0.78	0.78	-0.10791
ID8	0	0.56	0.44	-0.35655
ID2	0	0.51	0.49	-0.3098
ID3	1	0.47	0.47	-0.3279
ID4	1	0.32	0.32	-0.49485
ID5	0	0.1	0.9	-0.04576

3.Take the negative average of the values we get in the 2nd step.

As you can see these log values are negative. To deal with the negative sign, we take the **negative average of these values**, to maintain a common convention that lower loss scores are better.

$$log\ loss = -1/N \sum_{i=1}^{N} (log\,(P\,i\,))$$

We may use the formula to summarize all of the preceding steps: -

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Binary Cross-Entropy / Log Loss

Reference

- https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/#:~:text=A%20Confusion%20matrix%20is%20an,by%20the%20machine%20learning%20model.
- https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_algorithms_performance_metrics.htm
- https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
- https://towardsdatascience.com/performance-metrics-receiver-operating-characteristic-roc-area-under-curve-auc-79d6d5b0b977