Evaluation Metrics

The idea of building machine learning models works on a constructive feedback principle. You build a model, get feedback from metrics, make improvements and continue until you achieve a desirable accuracy. Evaluation metrics explain the performance of a model. An important aspect of evaluation metrics is their capability to discriminate among model results.

1. Confusion Matrix

A confusion matrix is an N X N matrix, where N is the number of classes being predicted.

2. F1 Score

F1-Score is the harmonic mean of precision and recall values for a classification problem. The formula for F1-Score is as follows:

$$F_1 = \left(\frac{\text{recall}^{-1} + \text{precision}^{-1}}{2} \right)^{-1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

3. Gain and Lift charts

Gain and Lift chart are mainly concerned to check the rank ordering of the probabilities. Here are the steps to build a Lift/Gain chart:

Step 1 : Calculate probability for each observation

Step 2 : Rank these probabilities in decreasing order.

Step 3 : Build deciles with each group having almost 10% of the observations.

Step 4 : Calculate the response rate at each deciles for Good (Responders) ,Bad (Non-responders) and total.

4. Kolomogorov Smirnov chart

K-S or Kolmogorov-Smirnov chart measures performance of classification models. More accurately, K-S is a measure of the degree of separation between the positive and negative distributions. The K-S is 100, if the scores partition the population into two separate groups in which one group contains all the positives and the other all the negatives.

5. Area Under the ROC curve (AUC – ROC)

This is again one of the popular metrics used in the industry. The biggest advantage of using ROC curve is that it is independent of the change in proportion of responders.

6. Log Loss

AUC ROC considers the predicted probabilities for determining our model's performance. However, there is an issue with AUC ROC, it only takes into account the order of probabilities and hence it does not take into account the model's capability to predict higher probability for samples more likely to be positive. In that case, we could us the log loss which is nothing but negative average of the log of corrected predicted probabilities for each instance.

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

- p(yi) is predicted probability of positive class
- 1-p(yi) is predicted probability of negative class
- yi = 1 for positive class and 0 for negative class (actual values)

7. Gini Coefficient

Gini coefficient is sometimes used in classification problems. Gini coefficient can be straigh away derived from the AUC ROC number. Gini is nothing but ratio between area between the ROC curve and the diagnol line & the area of the above triangle. Following is the formulae used :

Gini = 2*AUC – 1

Gini above 60% is a good model. For the case in hand we get Gini as 92.7%.

8. Concordant – Discordant ratio

This is again one of the most important metric for any classification predictions problem. To understand this let's assume we have 3 students who have some likelihood to pass this year. Following are our predictions :

A – 0.9

B – 0.5

C – 0.3

Now picture this. if we were to fetch pairs of two from these three student, how many pairs will we have? We will have 3 pairs : AB , BC, CA. Now, after the year ends we saw that A and C passed this year while B failed. No, we choose all the pairs where we will find one responder and other non-responder. How many such pairs do we have?

We have two pairs AB and BC. Now for each of the 2 pairs, the concordant pair is where the probability of responder was higher than non-responder. Whereas discordant pair is where the vice-versa holds true. In case both the probabilities were equal, we say its a tie. Let's see what happens in our case :

AB – Concordant

BC – Discordant

Hence, we have 50% of concordant cases in this example. Concordant ratio of more than 60% is considered to be a good model. This metric generally is not used when deciding how many customer to target etc. It is primarily used to access the model's predictive power. For decisions like how many to target are again taken by KS / Lift charts.

9. Root Mean Squared Error (RMSE)

RMSE is the most popular evaluation metric used in regression problems. It follows an assumption that error are unbiased and follow a normal distribution.

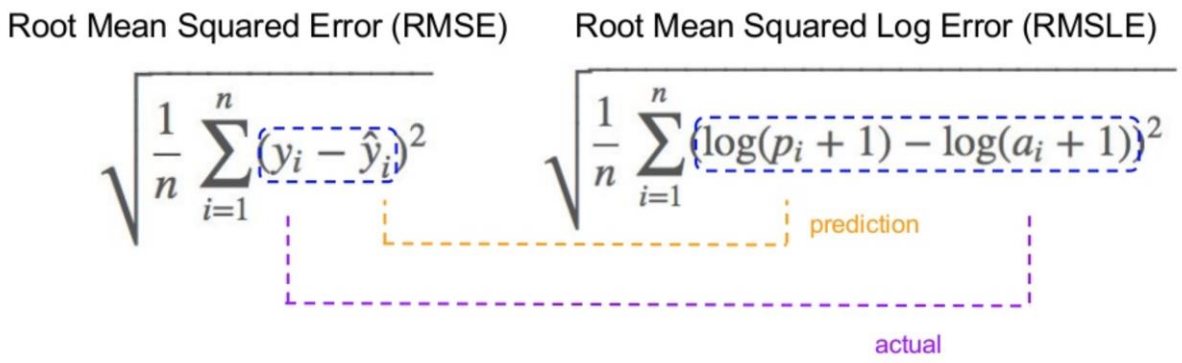
RMSE metric is given by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

where, N is Total Number of Observations.

10. Root Mean Squared Logarithmic Error

In case of Root mean squared logarithmic error, we take the log of the predictions and actual values. So basically, what changes are the variance that we are measuring. RMSLE is usually used when we don't want to penalize huge differences in the predicted and the actual values when both predicted and true values are huge numbers.



1. If both predicted and actual values are small: RMSE and RMSLE are same.
2. If either predicted or the actual value is big: RMSE > RMSLE
3. If both predicted and actual values are big: RMSE > RMSLE (RMSLE becomes almost negligible)

11. R-Squared/Adjusted R-Squared

In the case of a classification problem, if the model has an accuracy of 0.8, we could gauge how good our model is against a random model, which has an accuracy of 0.5. So the random model can be treated as a benchmark. But when we talk about the RMSE metrics, we do not *have* a benchmark to compare.

This is where we can use R-Squared metric. The formula for R-Squared is as follows:

$$R^2 = 1 - \frac{\text{MSE(model)}}{\text{MSE(baseline)}}$$

$$\frac{\text{MSE(model)}}{\text{MSE(baseline)}} \qquad \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(\overline{y}_i - \hat{y}_i)^2}$$

MSE(model): Mean Squared Error of the predictions against the actual values

MSE(baseline): Mean Squared Error of mean prediction against the actual values

In other words how good our regression model as compared to a very simple model that just predicts the mean value of target from the train set as predictions.

12. Cross Validation

Cross Validation is one of the most important concepts in any type of data modelling. It simply says, try to leave a sample on which you do not train the model and test the model on this sample before finalizing the model.

Above diagram shows how to validate model with in-time sample. We simply divide the population into 2 samples, and build model on one sample. Rest of the population is used for in-time validation.