# Session – 4

# Model Evaluation & Testing

## What is Model Evaluation?

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses. Model evaluation is important to assess the efficacy of a model during initial research phases, and it also plays a role in model monitoring.

To understand if your model(s) is working well with new data, we can add number of evaluation metrics.

## What are Evaluation Metrics?

Evaluation metrics are used to measure the quality of the statistical or machine learning model. Evaluating machine learning model or algorithms is essential for any project. There are many different types of evaluation metrics available to test a model. These include classification accuracy, logarithmic loss, confusion matrix and others. Evaluation metrics involves using a combination of these individual evaluation metrics to test a model or algorithm.

**Why is this useful?**

It is very important to use multiple evaluation metrics to evaluate your model. This is because a model may perform well using one measurement from one evaluation metric, but may perform poorly using another measurement from another evaluation metric. Using evaluation metrics are critical in ensuring that your model is operating correctly and optimally.

## Applications of evaluation Metrics

Statistical Analysis

Machine learning

## Classification

The most popular metrics for measuring classification performance include accuracy, precision, confusion matrix, log-loss, and AUC (area under the ROC curve).

## 1. Confusion Matrix

A confusion matrix is an N X N matrix, where N is the number of classes being predicted. For the problem in hand, we have N=2, and hence we get a 2 X 2 matrix. Here are a few definitions, you need to remember for a confusion matrix

- **Accuracy** : the proportion of the total number of predictions that were correct.
- **Positive Predictive Value or Precision** : the proportion of positive cases that were correctly identified.
- **Negative Predictive Value** : the proportion of negative cases that were correctly identified.
- **Sensitivity or Recall** : the proportion of actual positive cases which are correctly identified.
- **Specificity** : the proportion of actual negative cases which are correctly identified.

| Confusion Matrix | | Target | | | |
|---|---|---|---|---|---|
| | | Positive | Negative | | |
| **Model** | Positive | a | b | *Positive Predictive Value* | a/(a+b) |
| | Negative | c | d | *Negative Predictive Value* | d/(c+d) |
| | | *Sensitivity* | *Specificity* | **Accuracy** = (a+d)/(a+b+c+d) | |
| | | a/(a+c) | d/(b+d) | | |

### 2. F1 Score

In the last section, we discussed precision and recall for classification problems and also highlighted the importance of choosing precision/recall basis our use case. What if for a use

case, we are trying to get the best precision and recall at the same time? 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. Precision**

This refers to the proportion (total number) of all observations that have been predicted to belong to the positive class and are actually positive. The formula for Precision Evaluation Metric is as follows:

$$Precision = \frac{TP}{TP + FP}$$

**4. Recall**

This is the proportion of observation predicted to belong to the positive class, that truly belongs to the positive class. It indirectly tells us the model's ability to randomly identify an observation that belongs to the positive class. The formula for Recall is as follows:

$$Recall = \frac{TP}{TP + FN}$$

## Coefficient of Determination (R Squared)

The coefficient of determination, $R^2$, is used to analyze how differences in one variable can be explained by a difference in a second variable.

More specifically, R-squared gives you the percentage variation in y explained by x-variables. The range is 0 to 1 (i.e. 0% to 100% of the variation in y can be explained by the x-variables).

The coefficient of determination, $R^2$, is similar to the correlation coefficient, R. The correlation coefficient formula will tell you how strong of a linear relationship there is

between two variables. R Squared is the square of the correlation coefficient, *r* (hence the term r squared).

Finding R Squared / The Coefficient of Determination

**Step 1:** *Find the* correlation coefficient*, r (it may be given to you in the question).* Example, r = **0.543**.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\,n\Sigma x^2 - (\Sigma x)^2\,][\,n\Sigma y^2 - (\Sigma y)^2\,]}}$$

***Step 2:*** *Square the correlation coefficient.*

$0.543^2$ = **.295**

**Step 3:** *Convert the correlation coefficient to a percentage*.

.295 = **29.5%**

Meaning of the Coefficient of Determination

The coefficient of determination can be thought of as a percent. It gives you an idea of how many data points fall within the results of the line formed by the regression equation. The higher the coefficient, the higher percentage of points the line passes through when the data points and line are plotted. If the coefficient is 0.80, then 80% of the points should fall within the regression line. Values of 1 or 0 would indicate the regression line represents all or none of the data, respectively. A higher coefficient is an indicator of a better goodness of fit for the observations.

The CoD can be **negative**, although this usually means that your model is a poor fit for your data.

## 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. Here are the key points to consider on RMSE:

1. The power of 'square root' empowers this metric to show large number deviations.
2. The 'squared' nature of this metric helps to deliver more robust results which prevents cancelling the positive and negative error values. In other words, this metric aptly displays the plausible magnitude of error term.
3. It avoids the use of absolute error values which is highly undesirable in mathematical calculations.
4. When we have more samples, reconstructing the error distribution using RMSE is considered to be more reliable.
5. RMSE is highly affected by outlier values. Hence, make sure you've removed outliers from your data set prior to using this metric.
6. As compared to mean absolute error, RMSE gives higher weightage and punishes large errors.

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.

## Optimize regression model

These regression models involve the use of an optimization algorithm to find a set of coefficients for each input to the model that minimizes the prediction error. Because the models are linear and well understood, efficient optimization algorithms can be used.

In the case of linear regression, the coefficients can be found by least squares optimization, which can be solved using linear algebra. In the case of logistic regression, a local search optimization algorithm is commonly used.

It is possible to use any arbitrary optimization algorithm to train linear and logistic regression models.

That is, we can define a regression model and use a given optimization algorithm to find a set of coefficients for the model that result in a minimum of prediction error or a maximum of classification accuracy.

Using alternate optimization algorithms is expected to be less efficient on average than using the recommended optimization. Nevertheless, it may be more efficient in some specific cases, such as if the input data does not meet the expectations of the model like a Gaussian distribution and is uncorrelated with outer inputs

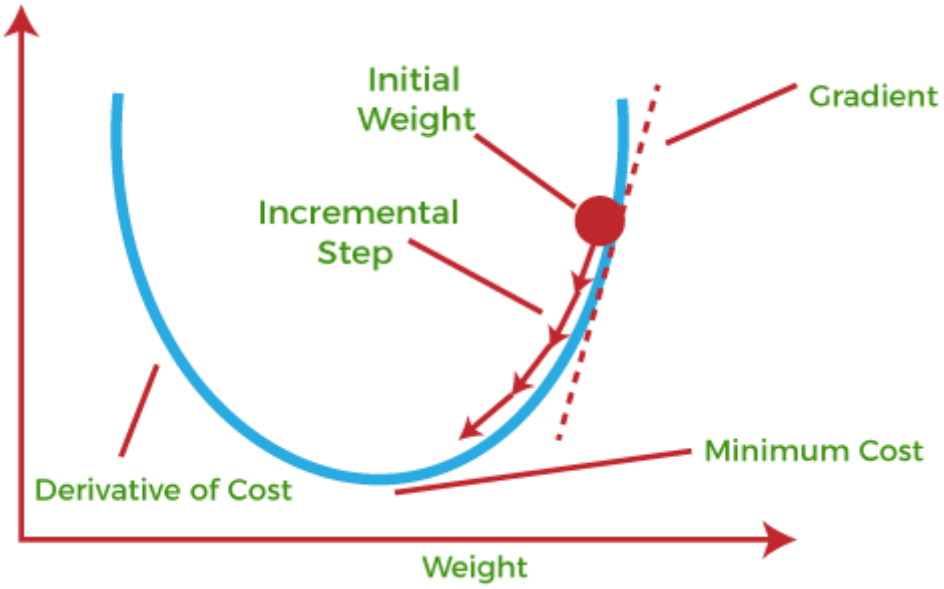
Gradient Descent

*Gradient Descent is defined as one of the most commonly used iterative optimization algorithms of machine learning to train the machine learning and deep learning models. It helps in finding the local minimum of a function.*

The best way to define the local minimum or local maximum of a function using gradient descent is as follows:

If we move towards a negative gradient or away from the gradient of the function at the current point, it will give the local minimum of that function.

Whenever we move towards a positive gradient or towards the gradient of the function at the current point, we will get the local maximum of that function.




This entire procedure is known as Gradient Ascent, which is also known as steepest descent. The main objective of using a gradient descent algorithm is to minimize the cost function using iteration. To achieve this goal, it performs two steps iteratively:

Calculates the first-order derivative of the function to compute the gradient or slope of that function.

Move away from the direction of the gradient, which means slope increased from the current point by alpha times, where Alpha is defined as Learning Rate. It is a tuning parameter in the optimization process which helps to decide the length of the steps.

.