**Evaluation metrics**

Evaluation metrics are used to assess the performance and effectiveness of machine learning models. These metrics provide a quantitative measure of how well the model is performing its task. The choice of evaluation metric depends on the specific problem type (classification, regression, clustering, etc.) and the objectives of the machine learning task. Here are some commonly used evaluation metrics for different types of machine learning tasks:

Classification Metrics:

1. Accuracy: The proportion of correctly classified instances to the total number of instances. It is a common metric for balanced datasets but can be misleading for imbalanced datasets.
2. Precision: The proportion of true positive predictions to the total positive predictions. It measures the model's ability to avoid false positives.
3. Recall (Sensitivity or True Positive Rate): The proportion of true positive predictions to the total actual positive instances. It measures the model's ability to capture all positive instances.
4. F1 Score: The harmonic mean of precision and recall. It balances precision and recall and is useful for imbalanced datasets.
5. Specificity (True Negative Rate): The proportion of true negative predictions to the total actual negative instances.
6. ROC-AUC (Receiver Operating Characteristic - Area Under the Curve): It plots the true positive rate against the false positive rate at different classification thresholds and measures the model's ability to distinguish between classes.

Regression Metrics:

1. Mean Squared Error (MSE): The average of the squared differences between predicted and actual values. It penalizes larger errors more heavily.
2. Root Mean Squared Error (RMSE): The square root of the MSE, which gives errors in the same unit as the target variable.
3. Mean Absolute Error (MAE): The average of the absolute differences between predicted and actual values. It is less sensitive to outliers than MSE.
4. R-squared (R2): Measures the proportion of variance in the target variable explained by the model. It ranges from 0 to 1, where 1 indicates a perfect fit.

Clustering Metrics:

1. Silhouette Score: Measures how well each data point is clustered. A higher silhouette score indicates better-defined clusters.
2. Davies-Bouldin Index: Evaluates cluster quality based on the average similarity between each cluster and its most similar cluster.

Other Metrics:

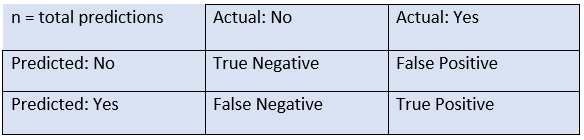
1. Mean Average Precision (MAP): Used in information retrieval tasks to evaluate the quality of ranked results.
2. Mean IoU (Intersection over Union): Used in semantic segmentation tasks to measure the overlap between predicted and actual regions.
3. Cohen's Kappa: Measures the agreement between two annotators or the agreement of a model's predictions with human annotators.

It is important to select the appropriate evaluation metric that aligns with the specific goals and requirements of the machine learning task. Different evaluation metrics provide different insights into the model's performance, and sometimes a combination of metrics is necessary to fully understand the model's behavior.

Confusion Matrix in Machine Learning

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an **error matrix**. Some features of Confusion matrix are given below:

* For the 2 prediction classes of classifiers, the matrix is of 2\*2 table, for 3 classes, it is 3\*3 table, and so on.
* The matrix is divided into two dimensions, that are **predicted values** and **actual values** along with the total number of predictions.
* Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.
* It looks like the below table:



The above table has the following cases:

* **True Negative:** Model has given prediction No, and the real or actual value was also No.
* **True Positive:** The model has predicted yes, and the actual value was also true.
* **False Negative:** The model has predicted no, but the actual value was Yes, it is also called as **Type-II error**.
* **False Positive:** The model has predicted Yes, but the actual value was No. It is also called a **Type-I error.**

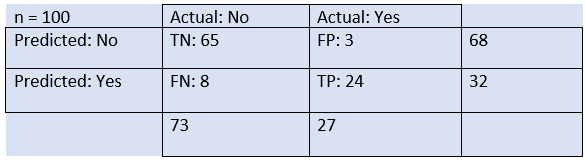
Need for Confusion Matrix in Machine learning

* It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.
* It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error.
* With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.

**Example**: We can understand the confusion matrix using an example.

Suppose we are trying to create a model that can predict the result for the disease that is either a person has that disease or not. So, the confusion matrix for this is given as:

Play Videox



From the above example, we can conclude that:

* The table is given for the two-class classifier, which has two predictions "Yes" and "NO." Here, Yes defines that patient has the disease, and No defines that patient does not has that disease.
* The classifier has made a total of **100 predictions**. Out of 100 predictions, **89 are true predictions**, and **11 are incorrect predictions**.
* The model has given prediction "yes" for 32 times, and "No" for 68 times. Whereas the actual "Yes" was 27, and actual "No" was 73 times.

Calculations using Confusion Matrix:

We can perform various calculations for the model, such as the model's accuracy, using this matrix. These calculations are given below:

* **Classification Accuracy:** It is one of the important parameters to determine the accuracy of the classification problems. It defines how often the model predicts the correct output. It can be calculated as the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers. The formula is given below:  
  Confusion Matrix in Machine Learning
* **Misclassification rate:** It is also termed as Error rate, and it defines how often the model gives the wrong predictions. The value of error rate can be calculated as the number of incorrect predictions to all number of the predictions made by the classifier. The formula is given below:  
  Confusion Matrix in Machine Learning
* **Precision:** It can be defined as the number of correct outputs provided by the model or out of all positive classes that have predicted correctly by the model, how many of them were actually true. It can be calculated using the below formula:  
  Confusion Matrix in Machine Learning
* **Recall:** It is defined as the out of total positive classes, how our model predicted correctly. The recall must be as high as possible.  
  Confusion Matrix in Machine Learning
* **F-measure:** If two models have low precision and high recall or vice versa, it is difficult to compare these models. So, for this purpose, we can use F-score. This score helps us to evaluate the recall and precision at the same time. The F-score is maximum if the recall is equal to the precision. It can be calculated using the below formula:
* Confusion Matrix in Machine Learning

Other important terms used in Confusion Matrix:

* **Null Error rate:** It defines how often our model would be incorrect if it always predicted the majority class. As per the accuracy paradox, it is said that "*the best classifier has a higher error rate than the null error rate.*"
* **ROC Curve:** The ROC is a graph displaying a classifier's performance for all possible thresholds. The graph is plotted between the true positive rate (on the Y-axis) and the false Positive rate (on the x-axis).

AUC-ROC Curve in Machine Learning

In Machine Learning, only developing an ML model is not sufficient as we also need to see whether it is performing well or not. It means that after building an ML model, we need to evaluate and validate how good or bad it is, and for such cases, we use different Evaluation Metrics. *AUC-ROC curve is such an evaluation metric that is used to visualize the performance of a classification model*. It is one of the popular and important metrics for evaluating the performance of the classification model. In this topic, we are going to discuss more details about the AUC-ROC curve.

What is AUC-ROC Curve?

AUC-ROC curve is a performance measurement metric of a classification model at different threshold values. Firstly, let's understand ROC (Receiver Operating Characteristic curve) curve.

ROC Curve

***ROC or Receiver Operating Characteristic curve represents a probability graph to show the performance of a classification model at different threshold levels***. The curve is plotted between two parameters, which are:

* **True Positive Rate or TPR**
* **False Positive Rate or FPR**

In the curve, TPR is plotted on Y-axis, whereas FPR is on the X-axis.

TPR:

TPR or True Positive rate is a synonym for Recall, which can be calculated as:



FPR or False Positive Rate can be calculated as:



Here, **TP**: True Positive

**FP**: False Positive

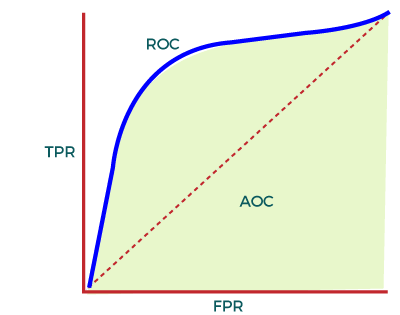
**TN**: True Negative

**FN**: False Negative

Now, to efficiently calculate the values at any threshold level, we need a method, which is AUC.

AUC: Area Under the ROC curve

AUC is known for **Area Under the ROC curve**. As its name suggests, AUC calculates the two-dimensional area under the entire ROC curve ranging from (0,0) to (1,1), as shown below image:



In the ROC curve, AUC computes the performance of the binary classifier across different thresholds and provides an aggregate measure. The value of AUC ranges from 0 to 1, which means an excellent model will have AUC near 1, and hence it will show a good measure of Separability.

When to Use AUC-ROC

**AUC is preferred due to the following cases:**

* AUC is used to measure how well the predictions are ranked instead of giving their absolute values. Hence, we can say AUC is **Scale-Invariant.**
* It measures the quality of predictions of the model without considering the selected classification threshold. It means AUC is **classification-threshold-invariant.**

When not to use AUC-ROC

* AUC is not preferable when we need to calibrate probability output.
* Further, AUC is not a useful metric when there are wide disparities in the cost of false negatives vs false positives, and it is difficult to minimize one type of classification error.

How AUC-ROC curve can be used for the Multi-class Model?

Although the AUC-ROC curve is only used for binary classification problems, we can also use it for multiclass classification problems. For multi-class classification problems, we can plot N number of AUC curves for N number of classes with the One vs ALL method.

For example, if we have three different classes, X, Y, and Z, then we can plot a curve for X against Y & Z, a second plot for Y against X & Z, and the third plot for Z against Y and X.

Applications of AUC-ROC Curve

Although the AUC-ROC curve is used to evaluate a classification model, it is widely used for various applications. Some of the important applications of AUC-ROC are given below:

1. **Classification of 3D model**

The curve is used to classify a 3D model and separate it from the normal models. With the specified threshold level, the curve classifies the non-3D and separates out the 3D models.

1. **Healthcare**  
   The curve has various applications in the healthcare sector. It can be used to detect cancer disease in patients. It does this by using false positive and false negative rates, and accuracy depends on the threshold value used for the curve.
2. **Binary Classification**

AUC-ROC curve is mainly used for binary classification problems to evaluate their performance.