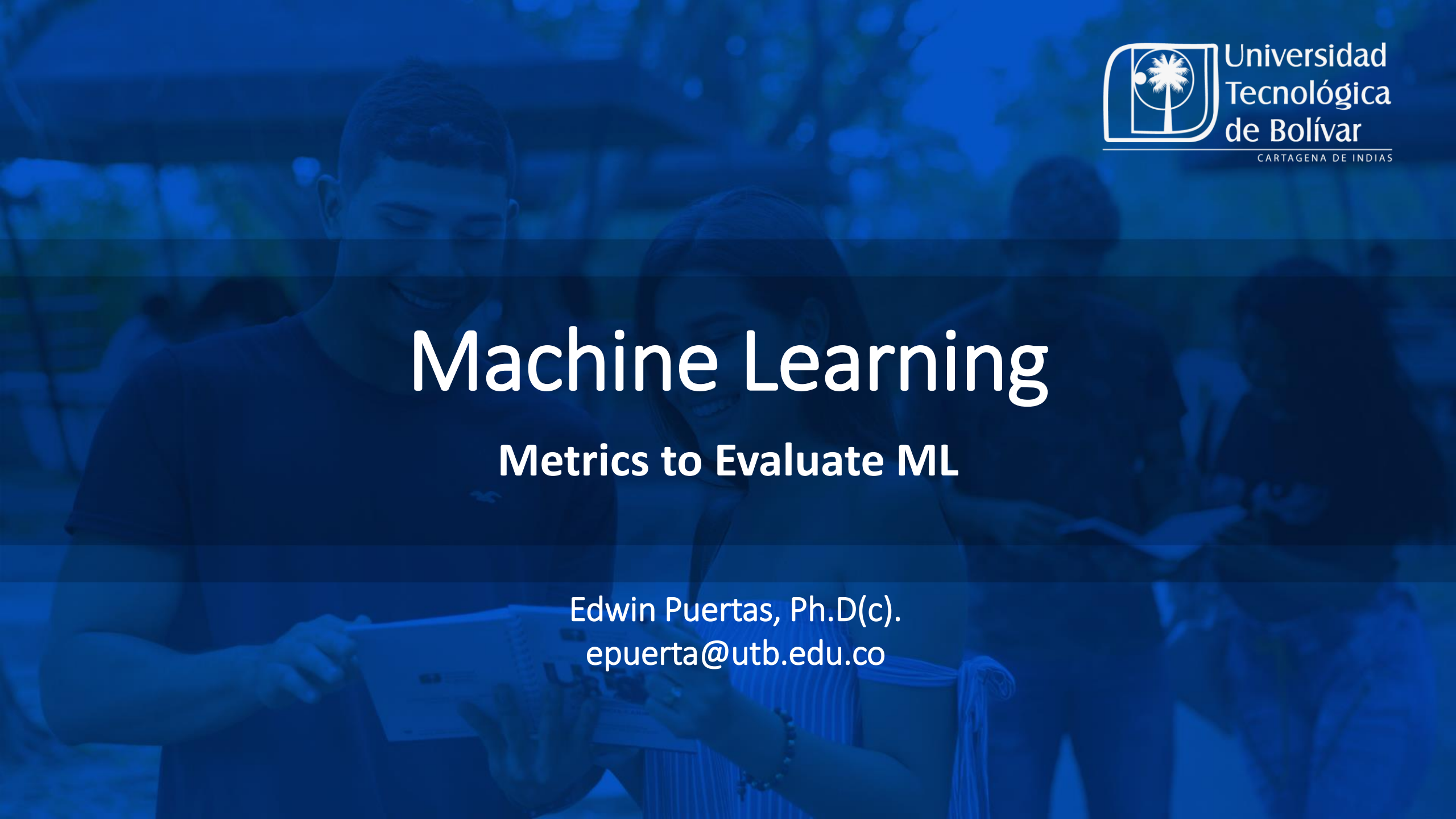


Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

Machine Learning

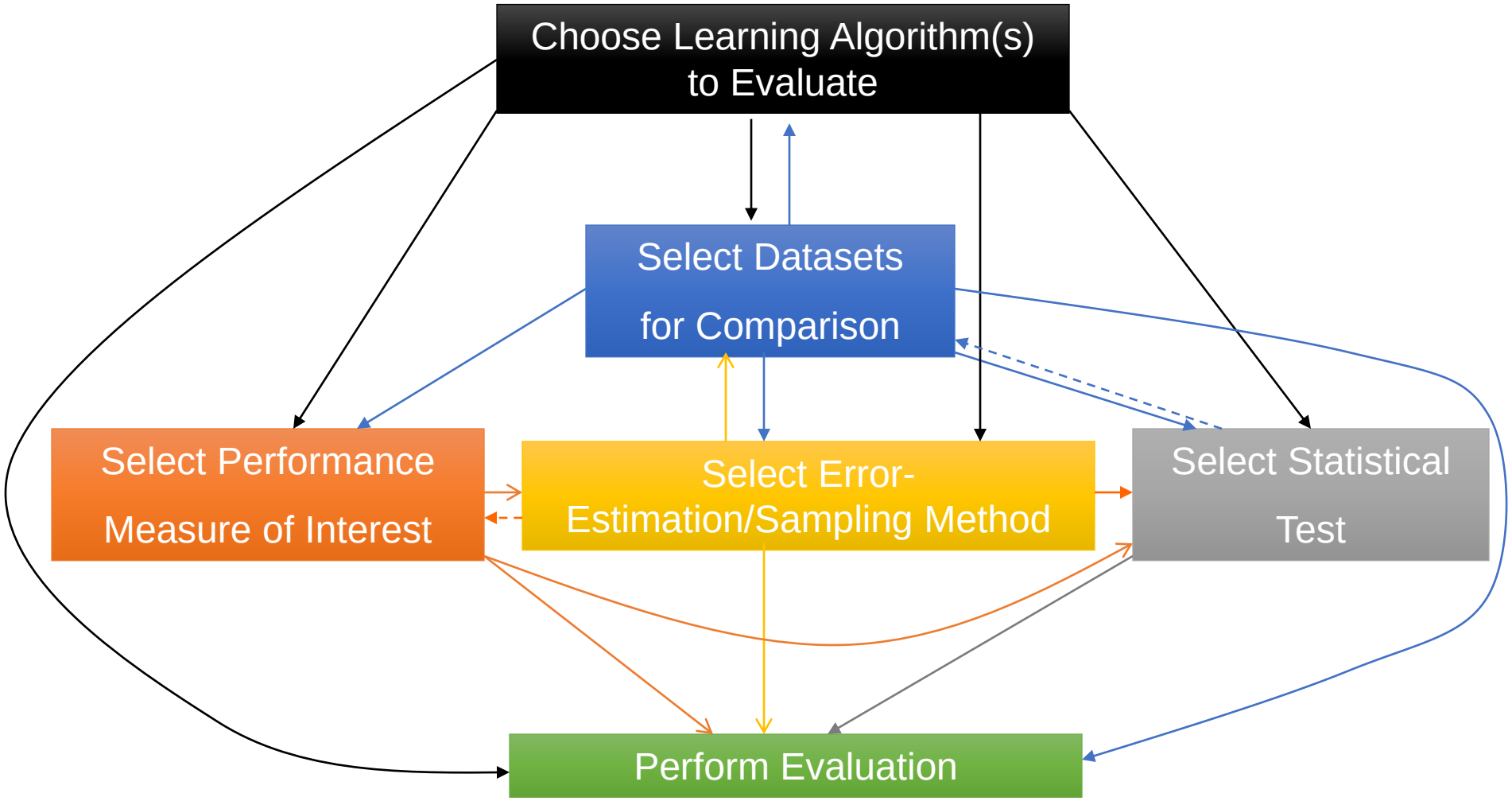
Metrics to Evaluate ML

Edwin Puertas, Ph.D(c).
epuerta@utb.edu.co

Motivation

- Evaluating the performance of learning systems is important because:
  - Learning systems are usually designed to predict the class of "future" unlabeled data points.
  - In some cases, evaluating hypotheses is an integral part of the learning process (example, when pruning a decision tree)

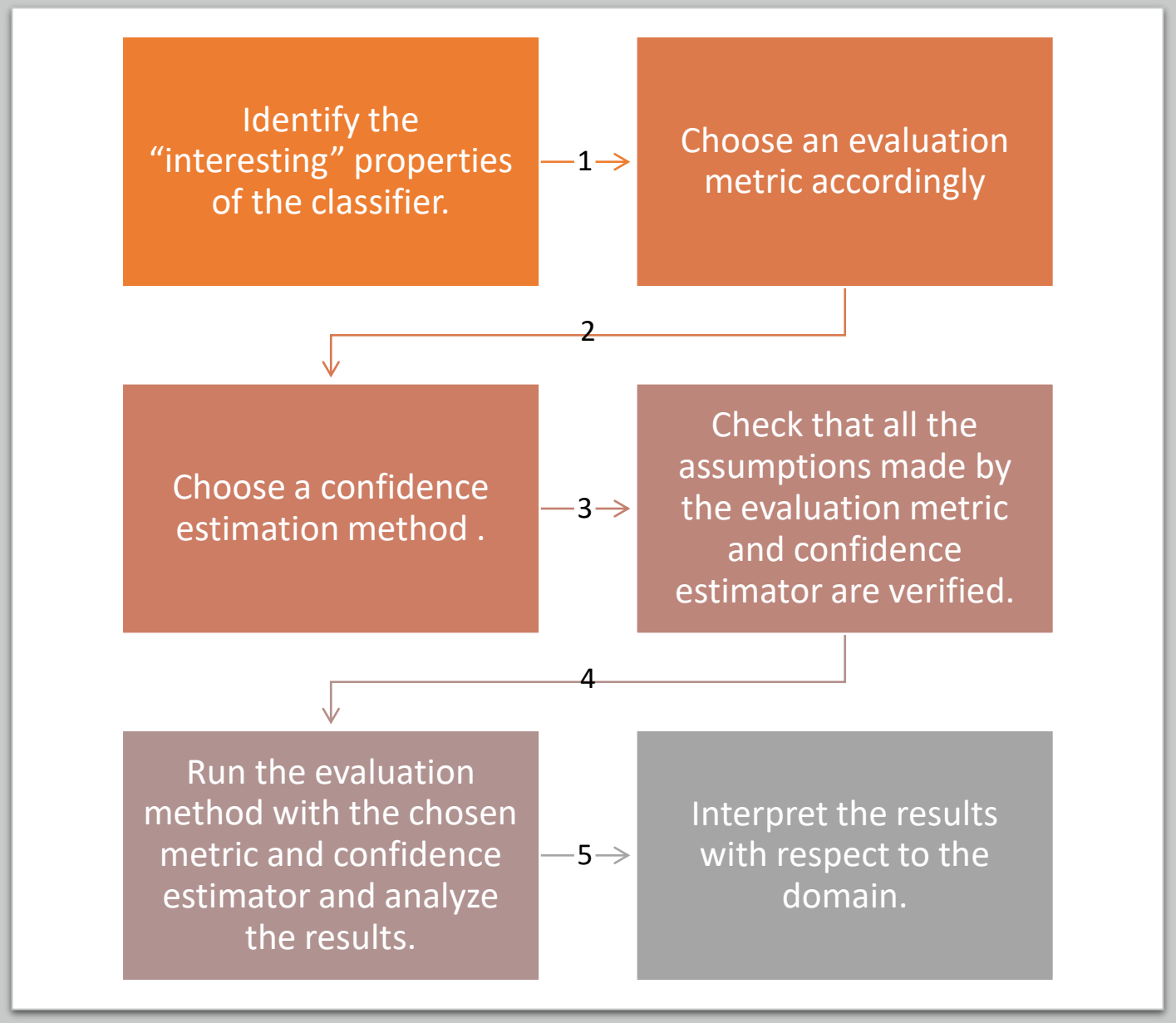
The Classifier Evaluation Procedure



1 ——→ 2 Means knowledge of 1 is necessary for 2

1 - - - - - - - - - → 2 Means feedback from 1 should be used to adjust 2

Typical Choices



Typical Choices II

Typical choices for Performance Evaluation:

- Accuracy
- Precision/Recall

Typical choices for Sampling Methods:

- Train/Test Sets (Why is this necessary?)
- K-Fold Cross-validation

Typical choices for significance estimation

- t-test (often a very bad choice, in fact!)

Classification Accuracy

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

Logarithmic Loss

Logarithmic Loss or Log Loss, works by penalising the false classifications. It works well for multi-class classification. When working with Log Loss, the classifier must assign probability to each class for all the samples. Suppose, there are N samples belonging to M classes, then the Log Loss is calculated as below :

$$LogarithmicLoss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * \log(p_{ij})$$

where,

- *y_ij,* indicates whether sample i belongs to class j or not
- *p_ij*, indicates the probability of sample i belonging to class j
- Log Loss has no upper bound and it exists on the range [0, ∞). Log Loss nearer to 0 indicates higher accuracy, whereas if the Log Loss is away from 0 then it indicates lower accuracy.

F1 Score

- F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).
- High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model. Mathematically, it can be expressed as :

F1 Score tries to find the balance between precision and recall.

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Precision: It is the number of correct positive results divided by the number of positive results predicted by the classifier

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

Recall: It is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

$$Precision = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Confusion Matrix

- Accuracy = (TP+TN)/(P+N)
- Precision = TP/(TP+FP)
- Recall/TP rate = TP/P
- FP Rate = FP/N

True class →	Pos	Neg
Hypothesized | class V		
Yes	TP	FP
No	FN	TN
	P=TP+FN	N=FP+TN

Mean Squared Error

Mean Squared Error(MSE) is quite like Mean Absolute Error, the only difference being that MSE takes the average of the square of the difference between the original values and the predicted values. The advantage of MSE being that it is easier to compute the gradient, whereas Mean Absolute Error requires complicated linear programming tools to compute the gradient. As, we take square of the error, the effect of larger errors become more pronounced than smaller error, hence the model can now focus more on the larger errors.

$$MeanSquaredError = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2$$

Sampling and Significance Estimation: Questions Considered

GIVEN THE OBSERVED ACCURACY OF A HYPOTHESIS OVER A LIMITED SAMPLE OF DATA, HOW WELL DOES THIS ESTIMATE ITS ACCURACY OVER ADDITIONAL EXAMPLES?

GIVEN THAT ONE HYPOTHESIS OUTPERFORMS ANOTHER OVER SOME SAMPLE DATA, HOW PROBABLE IS IT THAT THIS HYPOTHESIS IS MORE ACCURATE, IN GENERAL?

WHEN DATA IS LIMITED WHAT IS THE BEST WAY TO USE THIS DATA TO BOTH LEARN A HYPOTHESIS AND ESTIMATE ITS ACCURACY?

k-Fold Cross-Validation

1. Partition the available data D_0 into k disjoint subsets $T_1, T_2, \ldots, T_k$ of equal size, where this size is at least 30.
2. For i from **1** to k, do

   use T_i for the test set, and the remaining data for training set S_i

   - $S_i \leftarrow \{D_0 - T_i\}$
   - $h_A \leftarrow L_A(S_i)$
   - $h_B \leftarrow L_B(S_i)$
   - $\delta i \leftarrow error_{Ti}(h_A) - error_{Ti}(h_B)$
3. Return the value $avg(\delta)$, where

   $$avg(\delta) = 1/k \sum_{i=1}^{k} \delta_i$$

Confidence of the k-fold Estimate

The most used approach to confidence estimation in Machine learning is:

- To run the algorithm using 10-fold cross-validation and to record the accuracy at each fold.
- To compute a confidence interval around the average of the difference between these reported accuracies and a given gold standard, using the t-test, i.e., the following formula:

  $\delta +/- \ t_{N,9} * s_{\delta}$ where

  - δ is the average difference between the reported accuracy and the given gold standard,
  - $t_{N,9}$ is a constant chosen according to the degree of confidence desired,
  - $s_{\delta} = \text{sqrt}(1/90 \ \Sigma_{i=1}^{10} (\delta_i - \delta)^2)$ where δ_i represents the difference between the reported accuracy and the given gold standard at fold i.