Module 3

Evaluating learning for Intelligence (Arjun Panesar)

3.1 Model development and workflow, evaluation metrics, Parameters and Hyperparameters, Hyperparameter tuning algorithms, multivariate testing, Ethics of Intelligence.

"Intelligence is the ability to adapt to change"

—Stephen Hawking

Model development and workflow

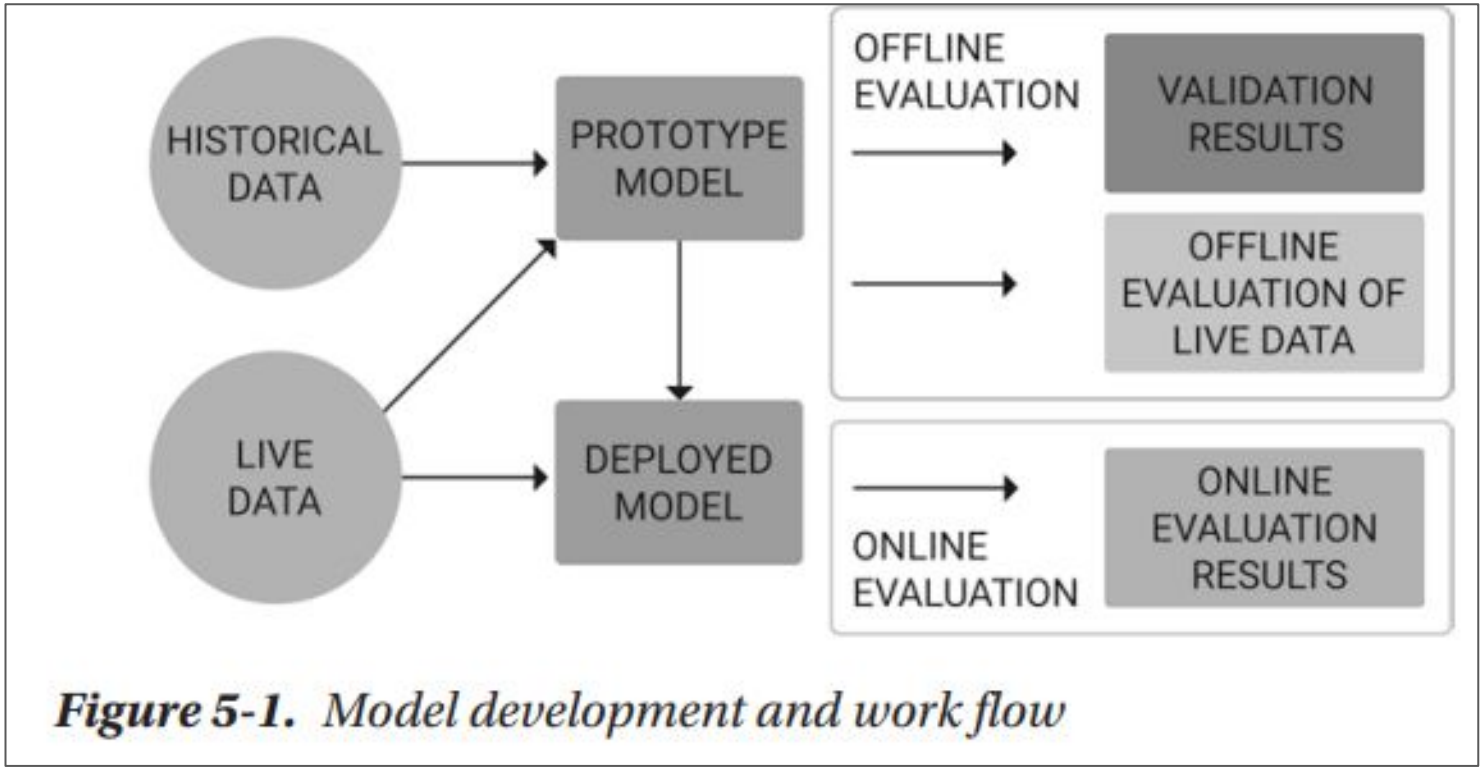


Figure 5-1. *Model development and work flow*

Model development and workflow

During this phase, **a prototype** is created through **testing various models on historical data** to determine the best model.Once the best prototype model is chosen, the model is **tested and validated.** Validating a model requires splitting datasets into **training, testing, and validation sets.** There are two ways of evaluating a machine learning model: **offline evaluation and online (or live) evaluation.**

A *deployed machine learning model consumes data from two sources*: **historical data** (or the data that is used as the experience to be learned from) and **live data.** Many machine learning models assume stationary distribution data—that the data distribution is constant over time. However, this is atypical of real life, as distributions of data often change over time—known as a **distribution shift**. For instance, consider a system that predicts the side effects of medications to patients based on their health profile. Medication side effects may change based on population factors such as ethnicity, disease profile, territory, medication popularity, and new medications.

Model performance that is similar to or **within a threshold of permissibility** when evaluated on live data is deemed as a model that continues to fit the data.

Model development and workflow

Offline evaluation measures the model based on metrics learned and evaluated from the historical, stationary, distributed dataset. Metrics such as *accuracy and precision-recall* are typically used within the offline training stage. Offline evaluation techniques include the hold-back method and n-fold cross-validation.
Online evaluation refers to the evaluation of metrics once the model is deployed. Online evaluation, particularly in the digital age, can support multivariate testing to understand best-performing models
Feedback loops are key to ensuring systems are performing as intended and help **to understand the model in the context of use better**. This can be performed by a **human agent or automated t**hrough a contextually intelligent agent or users of the model

It is important that the **evaluation of a machine learning model** is based on a **statistically independent dataset** and not on the dataset it is trained on. This is because the *evaluation of the training dataset is optimistic about the model's true performance as it adapts to the dataset.* By evaluating the model with previously unseen data, there is a better estimate of the generalization error

Evaluation Metrics

Classification problems seek to give a label or classification to an input. There are several methods by which to measure performance, including **accuracy, precision-recall, confusion matrices, log-loss (logarithmic loss), and AUC (area under the curve).**

Accuracy is the simplest technique used in identifying whether a model is making correct predictions. It is calculated as a percentage of correct prediction over the total predictions made. **Accuracy = number of correct predictions/number of total predictions**

Accuracy is a general metric that does not consider the division between classes. Therefore, it does not consider misclassification or the associated penalty with misclassification.

Table 5-1. *Confusion matrix*

	Prediction: Positive	**Prediction: Negative**
Labeled positive	20	5
Labeled negative	15	10

Evaluation Metrics: accuracy

While accuracy is a fundamental metric, it can be misleading in healthcare due to

Imbalanced datasets:

Many healthcare datasets have a significant imbalance, with a much larger proportion of healthy individuals than those with a specific disease. A model can achieve high accuracy by simply predicting the majority class (healthy) without correctly identifying the minority class (sick).

Varying Costs of Errors:

In healthcare, the cost of a false negative (missing a diagnosis) is often much higher than the cost of a false positive (a false alarm). For example, failing to diagnose cancer can have devastating consequences, while an unnecessary follow-up test for a healthy person is less harmful.

Evaluation Metrics: accuracy

Context is Crucial:

Accuracy doesn't tell the whole story. Two models with the same accuracy might have very different performance characteristics in terms of how they handle different classes. Understanding the specific context and goals of the application is essential.

Focus on Specific Outcomes:

Different applications require different metrics. For example, in a diagnostic test, sensitivity (the ability to correctly identify sick individuals) and specificity (the ability to correctly identify healthy individuals) are often more important than overall accuracy.

Data Quality Issues:

Incomplete, inaccurate, or inconsistent data can significantly impact accuracy and make it an unreliable measure of performance.

Evaluation Metrics

When evaluating healthcare data and applications, it's crucial to consider a range of metrics beyond just accuracy, including:

Sensitivity/Recall: The proportion of actual positives that are correctly identified.

Specificity: The proportion of actual negatives that are correctly identified.

Precision: The proportion of predicted positives that are actually positive.

F1-score: The harmonic mean of precision and recall, providing a balanced measure.

Evaluation Metrics

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

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

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

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

$$F1\ score = 2\ \mathrm{x}\ \frac{Precision * Recall}{Precision + Recall}$$

$$Specificity = \frac{TN}{TN + FP}$$

Evaluation Metrics

Type I Error (False Positive):
Imagine a medical test for a disease. A Type I error would be diagnosing someone with the disease when they are actually healthy. **This leads to unnecessary treatment and potential harm.**

Type II Error (False Negative):
In the same medical test scenario, **a Type II error would be failing to diagnose someone who actually has the disease. This leads to delayed or missed treatment, which can be very dangerous.**

Which is more crucial?

The "worse" error depends on the specific situation. In some cases, a Type I error might be more problematic. For example, in criminal justice, wrongly convicting an innocent person (Type I) is often considered more severe than letting a guilty person go free (Type II). In other situations, a Type II error might be more critical. For example, in fraud detection, missing a fraudulent transaction (Type II) can lead to significant financial losses for the company.

Evaluation Metrics: Sensitivity/Recall

Sensitivity/Recall is a measure of how well a machine learning model can detect positive instances. It is also known as the true positive rate (TPR) or recall.

Let's consider an example of a medical test for a rare disease to understand the concept of sensitivity. Suppose that the test has a sensitivity of 95%. This means that if 100 people who have the disease take the test, the test will correctly identify 95 of them as positive, but it will miss 5 of them (false negatives).

Sensitivity = (True Positive)/(True Positive + False Negative)
A high sensitivity means that the model is correctly identifying most of the positive results, while a low sensitivity means that the model is missing a lot of positive results.

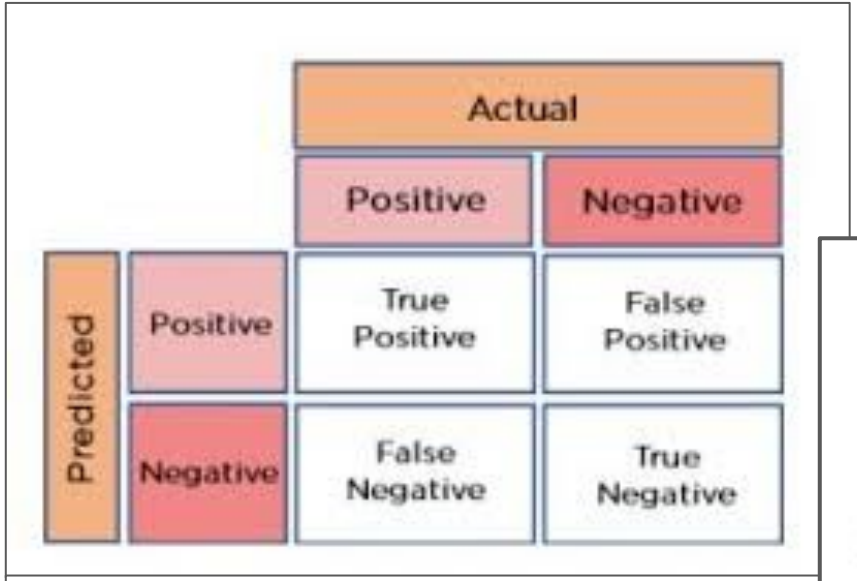
Evaluation Metrics:Specificity

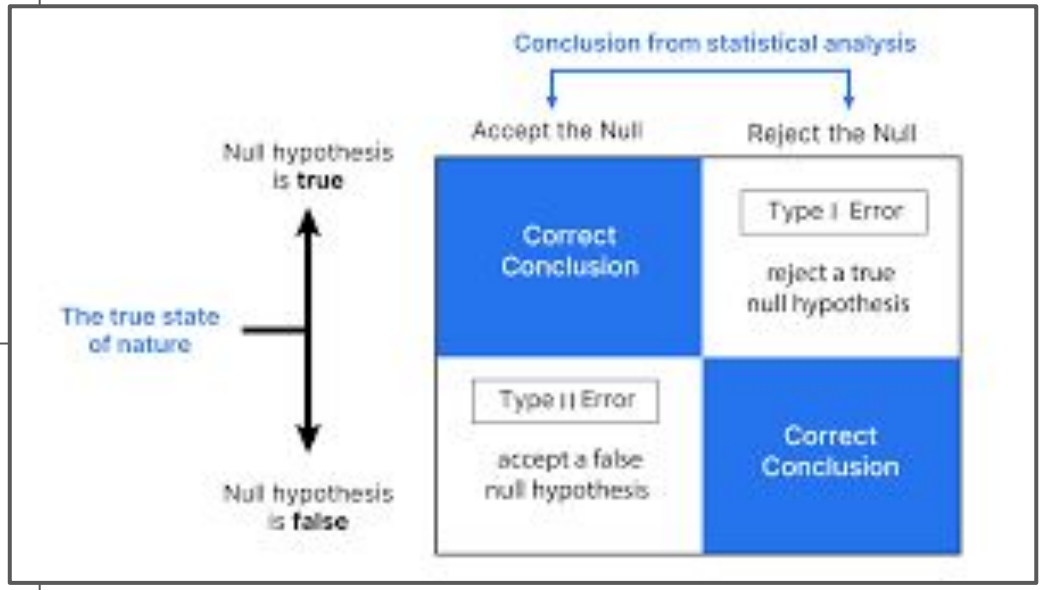
Specificity measures the proportion of true negatives that are correctly identified by the model. This implies that there will be another proportion of actual negative which got predicted as positive and could be termed as false positives.

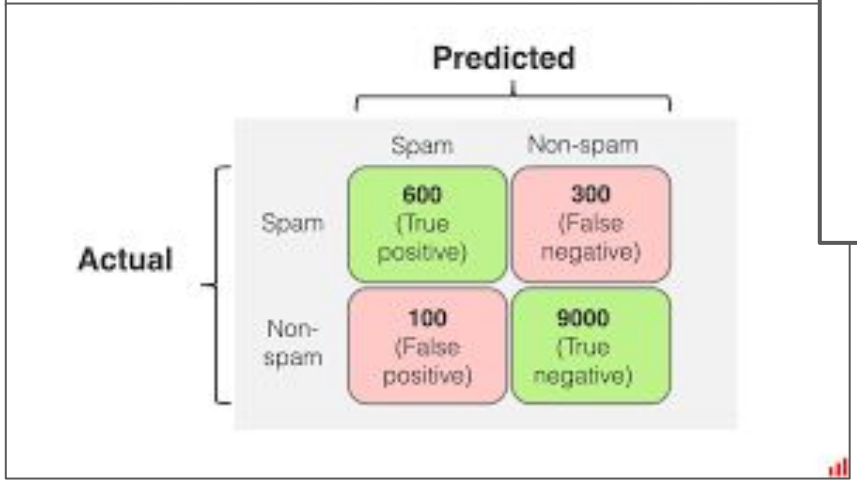
High specificity means that the model is correctly identifying most of the negative results, while a low specificity means that the model is mislabeling a lot of negative results as positive.

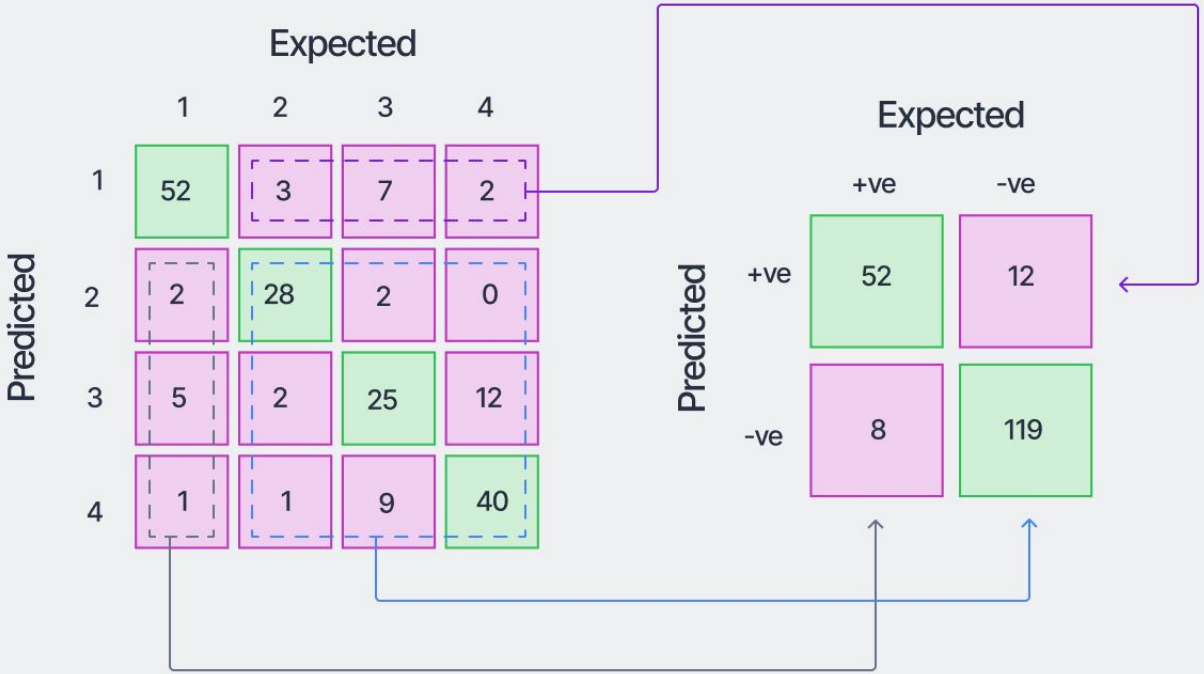
Let's consider an example of a medical test for a rare disease. Suppose that the test has a specificity of 95%. This means that if 100 people who do not have the disease take the test, the test will correctly identify 95 of them as negative, but it will incorrectly identify 5 of them as positive (false positives). Thus, the specificity, in this case, can be defined as a measure of the proportion of people not suffering from the disease who got predicted correctly as the ones who are not suffering from the disease. In other words, the proportion of person who is healthy actually got predicted as healthy is specificity.

Specificity = (True Negative)/(True Negative + False Positive)


Actual
Positive
Negative
Predicted
Positive
True Positive
False Positive
Negative
False Negative
True Negative


Conclusion from statistical analysis
Accept the Null
Reject the Null
Null hypothesis is true
The true state of nature
Null hypothesis is false
Correct Conclusion
Type I Error
reject a true null hypothesis
Type II Error
accept a false null hypothesis
Correct Conclusion


Predicted
Spam
Non-spam
Actual
Spam
600 (True positive)
300 (False negative)
Non-spam
100 (False positive)
9000 (True negative)



https://www.v7labs.com/blog/confusion-matrix-guide

Evaluation Metrics

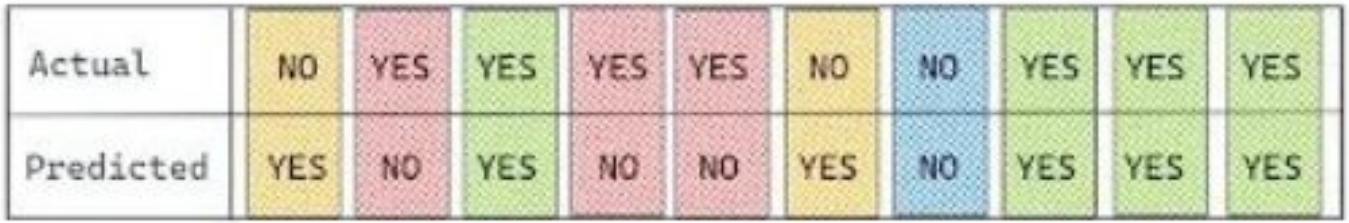
Per-class accuracy is an extension of accuracy that takes into account the accuracy of each class. **It is useful in distorted problems where there are a larger number of examples within one particular class compared to another.** The class with greater examples **dominates** the calculation, and therefore accuracy alone may not suffice for the nature of your model; thus it is useful to evaluate per-class accuracy also.

Logarithmic loss (or log-loss for short) is used for problems where a **continuous probability is predicted rather than a class label.** Log-loss provides a probabilistic measure of the confidence of the accuracy and considers the entropy between the distribution of true labels and predictions. For a binary classification problem, the logarithmic loss would be calculated as fo

$$\text{Log-loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log p_i + (1-y_i)\log(1-p_i)$$

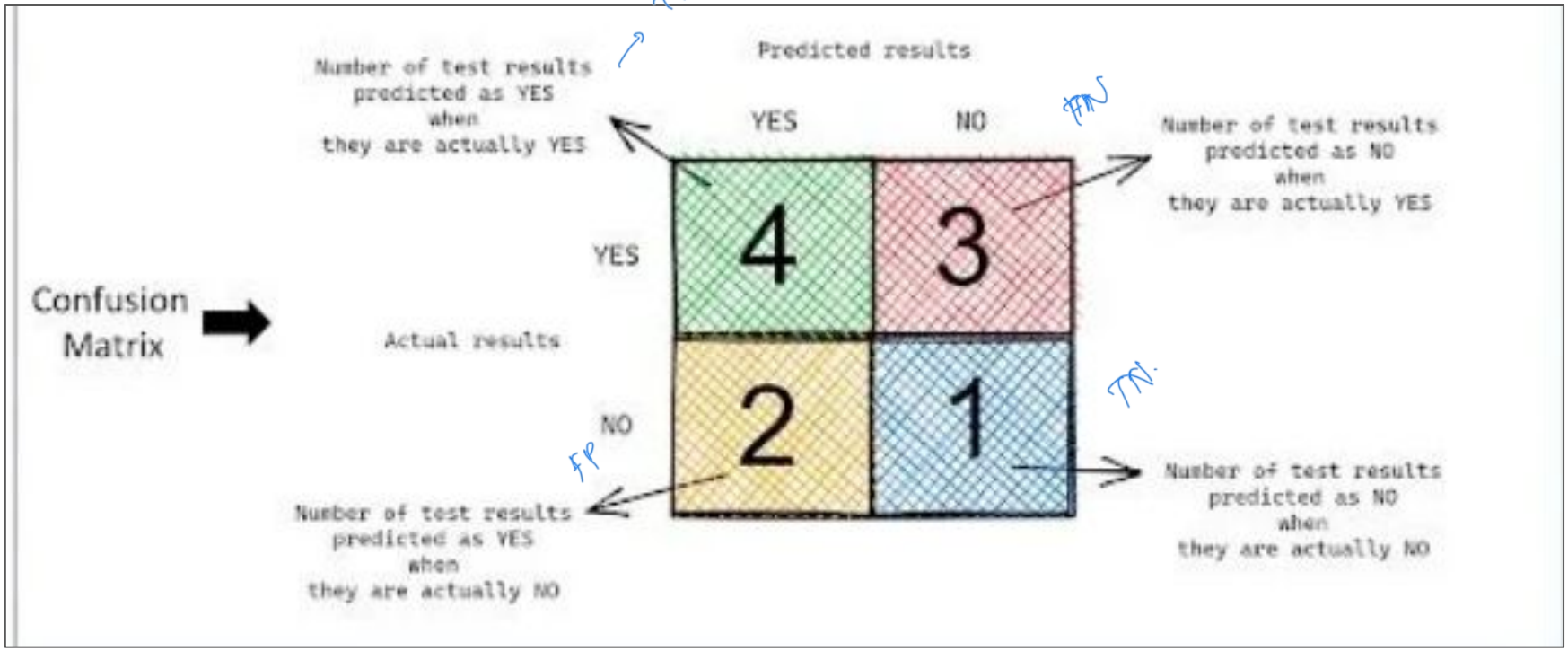
Where Pi is the probability of the ith data point belonging to a class and yi the true label (either 0 or 1).

Results



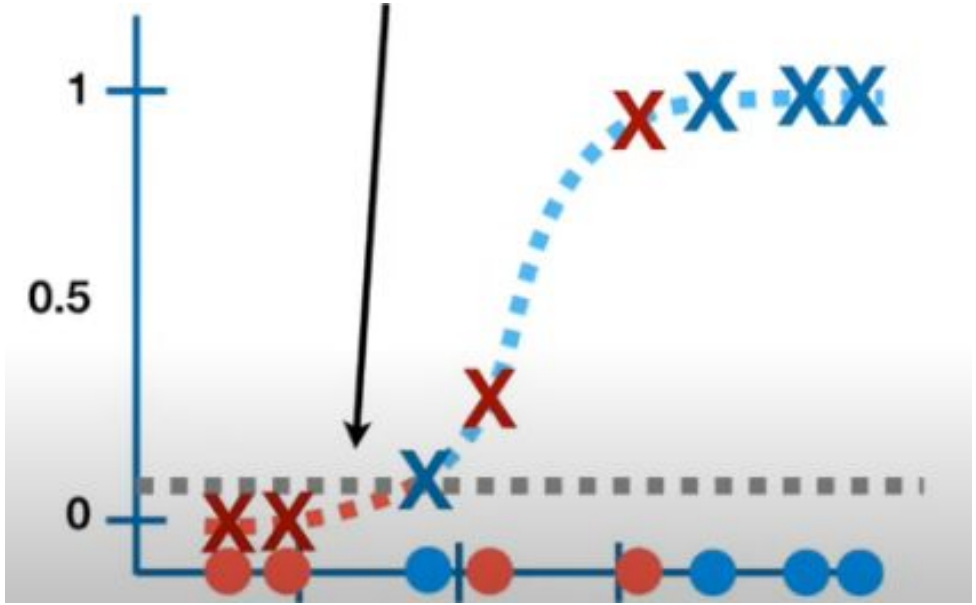
Actual	NO	YES	YES	YES	YES	NO	NO	YES	YES	YES
Predicted	YES	NO	YES	NO	NO	YES	NO	YES	YES	YES

Create confusion matrix


Confusion Matrix
Predicted results
YES
NO
Actual results
YES
NO
4
3
2
1
Number of test results predicted as YES when they are actually YES
Number of test results predicted as NO when they are actually YES
Number of test results predicted as YES when they are actually NO
Number of test results predicted as NO when they are actually NO
TP
FN
TN
FP

Not necessarily 0.5 threshold always gives a better accuracy, AUC gives the optimum threshold for decision making.

		Actual: Is Obese	Actual: Is Not Obese
Predicted	Is Obese	3	1
Predicted	Is Not Obese	1	3

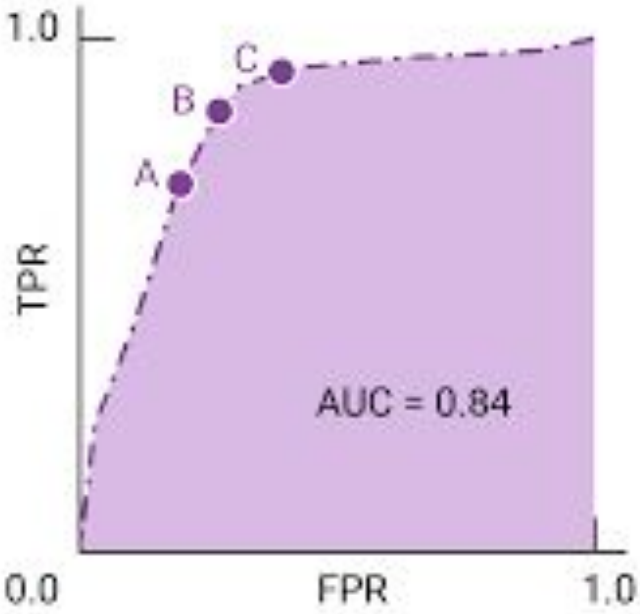


Evaluation Metrics: AUC

The Area Under the Curve (AUC) is a performance metric used in machine learning and other fields to evaluate **the ability of a model to distinguish between classes, particularly in binary classification**.

It represents the probability that a model will rank a randomly chosen positive instance higher than a randomly chosen negative instance. AUC values range from 0 to 1, with 1 indicating a perfect model and 0.5 indicating a model performing no better than random chance.

Evaluation Metrics: AUC



If false positives (false alarms) are highly costly, it may make sense to choose a threshold that gives a lower FPR, like the one at point A, even if TPR is reduced.

Conversely, if false positives are cheap and false negatives (missed true positives) highly costly, the threshold for point C, which maximizes TPR, may be preferable.

If the costs are roughly equivalent, point B may offer the best balance between TPR and FPR

Evaluation Metrics

Area Under the Curve (AUC)

The AUC **plots the rate of true positives to the rate of false positives.** The AUC enables the **visualization of the sensitivity and specificity of the classifier.** It highlights how many correct positive classifications can be gained allowing for false positives. **The curve is known as the receiver operating characteristic curve**, or **ROC** as shown in Figure 5-2.

A high AUC or greater space underneath the curve is good, and a smaller area under the curve (or less space under the curve) is undesirable. In Figure 5-2, test A has better AUC as compared to test B, as the AUC for test A is larger than for test B. The ROC visualizes the trade-off between specificity and sensitivity of the model.

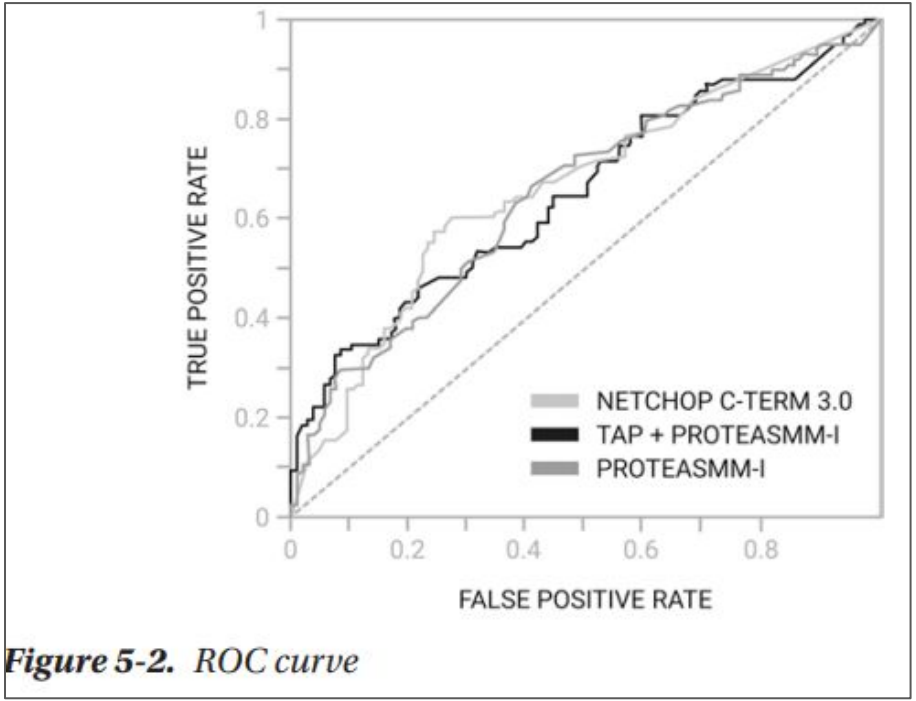


Figure 5-2. ROC curve

Evaluation Metrics

Regression machine learning models output continuous variables, and root-mean-squared error (RMSE) is the most commonly used evaluation metric for these problems
RMSE calculates the square root of the sum of the average distance between predicted and actual values. This can also be understood as the average Euclidean distance between the true value and predicted value vectors.
A criticism of RMSE is that it is sensitive to outliers, where yi denotes the actual value and yˆi denotes predicted value.

$$RMSE = \sqrt{\frac{\sum_i (y_i - \hat{y}_i)^2}{2}},$$

Evaluation Metrics

Percentiles (or quantiles) of error are more robust as a result of being less sensitive to outliers. **Real-world data is likely to contain outliers, and thus it is often useful to look at the median absolute percentage error (MAPE) rather than the mean,** where yi denotes the actual value and yˆi denotes predicted value. The MAPE is less affected by outliers by using the median of the dataset. A *threshold or percentage difference* for predictions can be set for a given problem to give an understanding of the precision of the regression estimate. The threshold depends on the nature of the problem

$$MAPE = median\left(\left|\left(y_i - \hat{y}_i\right) / \left(y_i\right)\right|\right),$$

Skewed Datasets, Anomalies, and Rare Data

An experienced data scientist treats all data with suspicion. Data can be inconsistent; and as a result, **skewed datasets, imbalanced class examples, and outliers can all significantly affect the performance of a model.**

- Having more examples within one class compared to another can lead to an underperforming model.
- Furthermore, outliers or data anomalies can further skew performance evaluation metrics.
- The effect of large outliers can be mitigated using percentiles of error.

In practice, good data cleansing, removal of outliers, and normalization of variables can reduce the sensitivity to outliers.

Parameters and Hyperparameters

Hyperparameters and parameters are often used interchangeably, yet there is a difference between the two. Machine learning models can be understood as mathematical models that represent the relationship between aspects of data.

Model parameters are properties of the training dataset that are learned and adjusted during training by the machine learning model.

Model parameters differ for each model, dataset properties, and the task at hand.

For instance, in the case of an NLP predictor that output the sophistication of a corpus of text, parameters such as word frequency, sentence length, and noun or verb distribution per sentence would be considered model parameters. Model hyperparameters are parameters to the model building process that are not learned during training.

Other Examples include the weights and biases in a neural network.

Parameters and Hyperparameters

- Hyperparameters can make a substantial difference to the performance of a machine learning model. Hyperparameters define the model architecture and effect the capacity of the model, influencing model flexibility.
- Hyperparameters can also be provided to loss optimization algorithms during the training process. Optimal setting of hyperparameters can have a significant effect on predictions and help prevent a model from overfitting.

Optimal hyperparameters often differ between datasets and models.

In the case of a neural network, for example, hyperparameters would include the number and size of hidden layers, weighting, learning rate, and so forth.

Decision trees hyperparameters would include the desired depth and number of leaves in the tree, the learning rate in gradient descent, the number of trees in a random forest, or the regularization strength.

Hyperparameters with a support vector machine would include a misclassification penalty term.

Parameters and Hyperparameters algorithms

Research paper, Alzheimer's, heart disease

1. Grid Search:
This method exhaustively searches through a predefined set of hyperparameter values. It's simple to implement but can be computationally expensive, especially when dealing with a large number of hyperparameters or a wide range of possible values.

2. Random Search:
Instead of evaluating all possible combinations, random search randomly samples hyperparameter values from the search space.

3. Bayesian Optimization:
This approach uses a probabilistic model to guide the search for optimal hyperparameters. It balances exploration (trying new hyperparameter combinations) and exploitation (focusing on promising areas) to find good solutions more efficiently than grid or random search

Multivariate Testing LINK

Multivariate testing is an extremely useful method of determining which model is best for the particular problem at hand.
Multivariate testing is known as **statistical hypothesis testing and determines the difference between a null hypothesis and alternative hypothesis**.
The null hypothesis is defined as the new model not affecting the average value of the performance metric; whereas the alternate hypothesis is that the new model does change the average value of the performance metric. **Multivariate testing compares similar models to understand which is performing best or compares a new model against an older, legacy model.**
The respective performance metrics are compared, and a decision is made on which model to proceed with.
The process of testing is as follows: 1. Split the population into randomized control and experimentation groups. 2. Record the behavior of the populations on the proposed hypotheses. 3. Compute the performance metrics and associated p-values. 4. Decide on which model to proceed with. Although the process seems relatively simple, there are a few key aspects for consideration.

Which Metric Should I Use for Evaluation?

Link

Ethics of Intelligence

"People worry that computers will get too smart and take over the world, but the real problem is that they're too stupid and they've already taken over the world."

—Pedro Domingos

Ethics of Intelligence

Material

https://www.frontiersin.org/articles/10.3389/fsurg.2022.862322/full

https://economictimes.indiatimes.com/news/india/icmr-comes-up-with-first-ethical-guidelines-for-application-of-ai-in-biomedical-research-healthcare/articleshow/99012557.cms?from=mdr

https://www.wionews.com/opinions-blogs/doctor-ai-will-see-you-now-navigating-the-ethics-and-regulations-of-smart-medicine-607812

https://news.mit.edu/2022/marzyeh-ghassemi-explores-downside-machine-learning-health-care-0201

3.1 Model Development & Workflow

- Prototype → Testing → Validation
- Data Splitting: Train | Validation | Test.
- Evaluation Types:
- Offline: Historical data (accuracy, precision-recall, cross-validation).
- Online: Deployed model, multivariate testing, feedback loops.
- Data Distribution Shift: Real-world data ≠ stationary → e.g., medication side effects vary by population.
- Independent Dataset: Must evaluate on unseen data (avoid optimistic bias).

🔑 Mnemonic: “P-DOI” → Prototype, Data split, Offline/Online, Independent dataset.

———

3.2 Evaluation Metrics

Classification Metrics

- Accuracy: % correct → limited in imbalanced healthcare data.
- Sensitivity/Recall (TPR): TP / (TP + FN). Detect sick correctly.
- Specificity: TN / (TN + FP). Detect healthy correctly.
- Precision: Predicted positives that are actually positive.
- F1-score: Harmonic mean of precision & recall.
- Type I Error: FP = false alarm.
- Type II Error: FN = missed diagnosis.
- Per-class Accuracy: Useful for imbalanced datasets.
- Log-loss: Confidence-based probability measure.
- AUC/ROC: Trade-off between sensitivity & specificity → Higher = better.

Regression Metrics

- RMSE: Square root of mean squared error → sensitive to outliers.
- Imbalanced classes → model bias.
- Outliers distort performance.
- Fix: Data cleansing, normalization, robust metrics.

🔑 Mnemonic: “ION” → Imbalance, Outliers, Normalization.

———

3.4 Parameters vs. Hyperparameters

- Parameters: Learned from data (e.g., weights, biases).
- Hyperparameters: Set before training (e.g., learning rate, depth, hid

capacity & flexibility.

🔑 Mnemonic: “PH” → Parameters = Learned, Hyperparameters = Set by Hand.

———

3.5 Hyperparameter Optimization

- Grid Search: Exhaustive search.
- Random Search: Random sampling.
- Bayesian Optimization: Probabilistic, balances exploration & exploitatio

🔑 Mnemonic: “GRB” → Grid, Random, Bayesian.

———

3.6 Multivariate Testing

- Hypothesis testing for best model.
- Steps:

1. Split population → control & experiment.
2. Record behavior.
3. Compute metrics + p-values.
4. Select model.

🔑 Mnemonic: “SRCS” → Split, Record, Compute, Select.

———

Master Mnemonic for Module 3

👉 “Patient Data Scientists Prefer Honest Great Models & Ethical AI.”

- Patient Data → P-DOI (Prototype, Data split, Offline/Online, Independent)
- Scientists → ASSP-FLAT RAM (Metrics)
- Prefer → ION (Imbalance, Outliers, Normalization)
- Honest → PH (Parameters vs Hyperparameters)
- Great → GRB (Grid, Random, Bayesian)
- Models → SRCS (Multivariate testing steps)