MACHINE LEARNING AND AI FOR HEALTHCARE

By Arjun Panesar

CHAPTER 5

Evaluating Learning for Intelligence

“Intelligence is the ability to adapt to change”

—Stephen Hawking

- ✓ Typically, the most laborious tasks within a machine learning project are identifying the appropriate model and engineering features, which make a substantial difference to the output of the model.

- ✓ In fact, the features chosen can often have more impact on the quality of a model compared to the model choice itself.

- ✓ Therefore, it is important to evaluate the learning algorithm that will determine the model's intelligence to predict the output of an unknown sample.

Model Development and Workflow

To successfully deploy a machine learning model, there are several stages of development and evaluation that take place

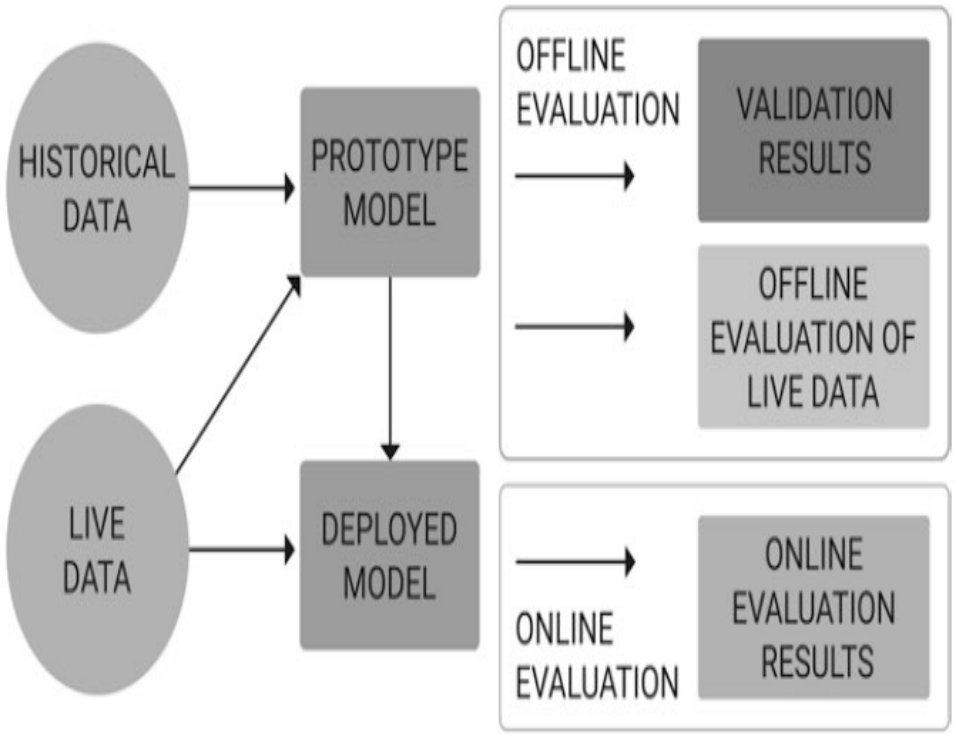


Figure 5-1. *Model development and work flow*

The first stage is the prototype phase.

During this phase, a prototype is created through testing various models on historical data to determine the best model.

Hyperparameter tuning, is a requirement of model training.

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.

Once the model has been successfully validated, it is deployed to production.

The model is then usually evaluated by one (or several) performance metrics.

 # Why Are There Two Approaches to Evaluating a Model?

- 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.

- e.g./ model that predicts the side effects of medications to patients based on their health profile.

- The distribution of relevant side effects based on patient data can vary quickly over time, and hence it is essential or a model to detect a shift in distribution and accordingly evolve the model.

- ✓ The model is evaluated based on live data (e.g., data that enters the system in real-time).
- ✓ This evaluation is conducted using validation metrics.
- ✓ The validation metrics were previously configured using historical data.
- ✓ 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.
- ✓ Degradation of model performance indicates that the model does not fit the data and requires retraining.

There are two ways of evaluating a machine learning model: offline evaluation and online (or live) evaluation.

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 s deployed.

The key takeaway is that these metrics may differ from the metrics used to evaluate performance when the model is deployed live.

For instance, a model that is learning on new pharmacological treatments may seek to be as precise as possible in training and validation; but when placed online, it may need to consider business goals such as budget or treatment value when 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.

human agent
automated through a contextually intelligent agent
users of the model

Validation:

- ✓ It is important that the evaluation of a machine learning model is based on a statistically independent dataset and not on the datasetit is trained on.
- ✓ By evaluating the model with previously unseen data, there is a better estimate of the generalization error.
- ✓ Methods such as n-fold cross-validation discussed are useful techniques for this purpose.
- ✓ Often the data used is more important than the algorithm choice; the better the features used, the greater the performance of the model.

- # Train test split

Some machine learning techniques may return one solution, whereas others may produce several. It is often a case of gathering their outputs(also referred to as hypotheses, or learned models) and evaluating their outputs.

The assessment of hypotheses is conducted through evaluating the predictive accuracy, comprehensibility, or utility.

Predictive accuracy refers to the accuracy at which theagent performs the task of classification.

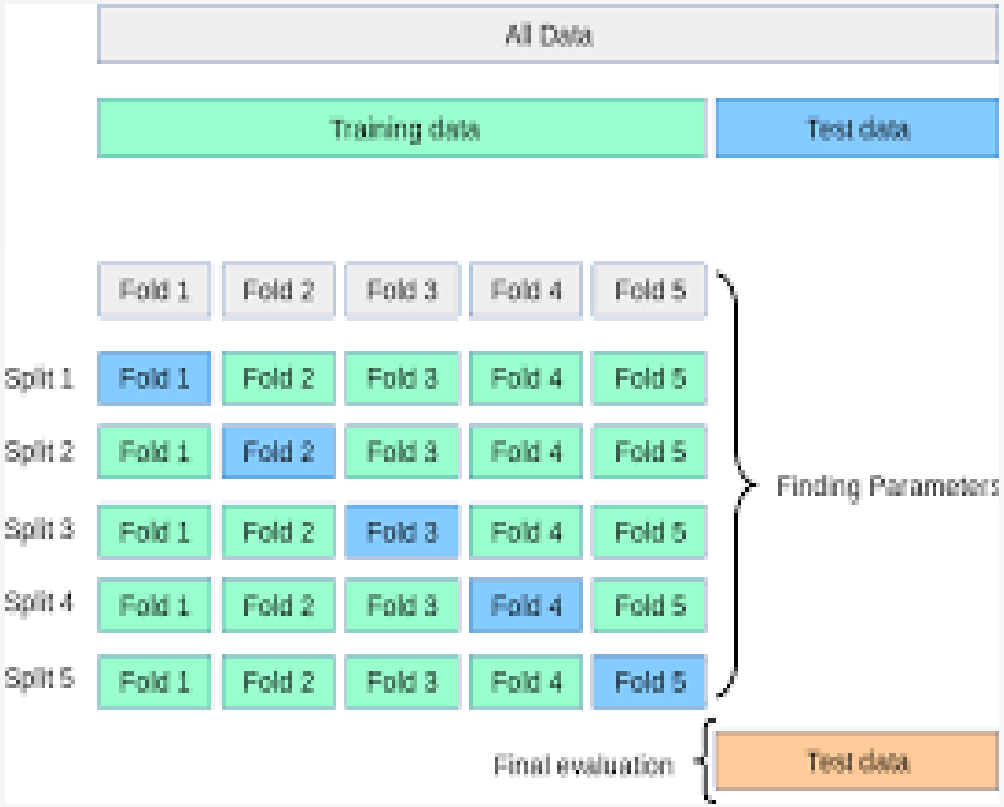
Comprehensibility refers to how well we as humanscan understand the output.

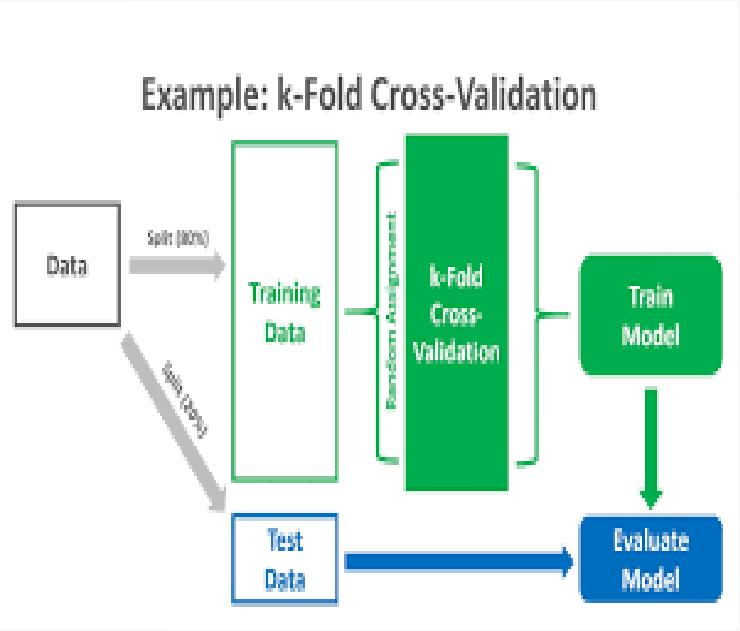
Utility refers to the problem-specific measure of worth.

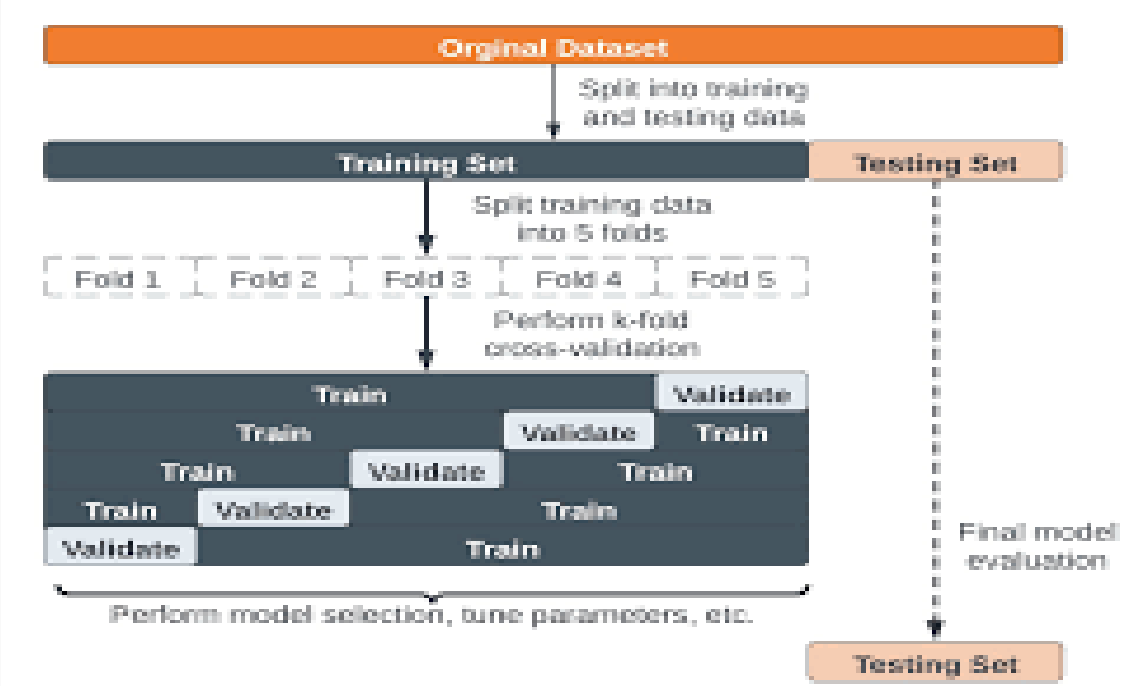
After the assessment is complete, a candidate hypothesis is chosen.

- **Hold back method**
- **N fold-cross validation**
- **Monte-Carlo Cross-Validation**
- ----------try many algorithms

N/K fold-cross validation







How can I evaluate my model?

- Aim of model
- Evaluation Metrics

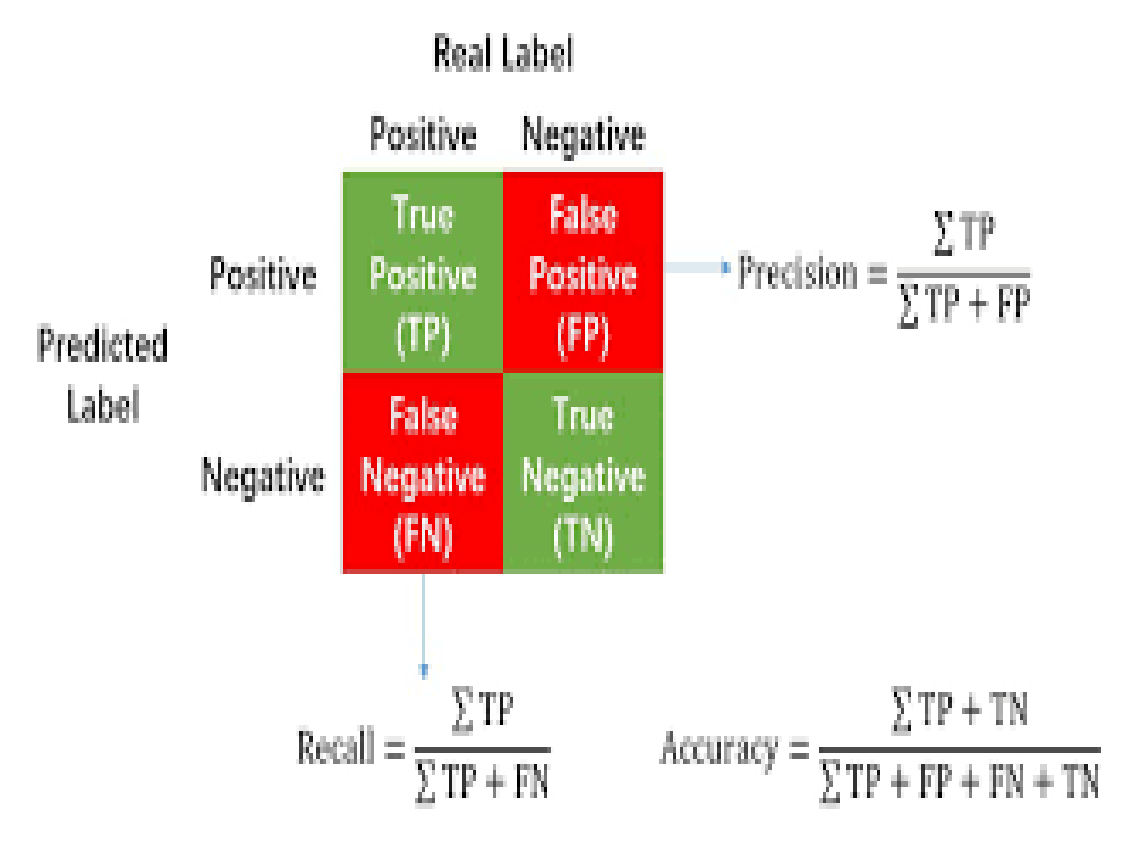
metrics package(R)

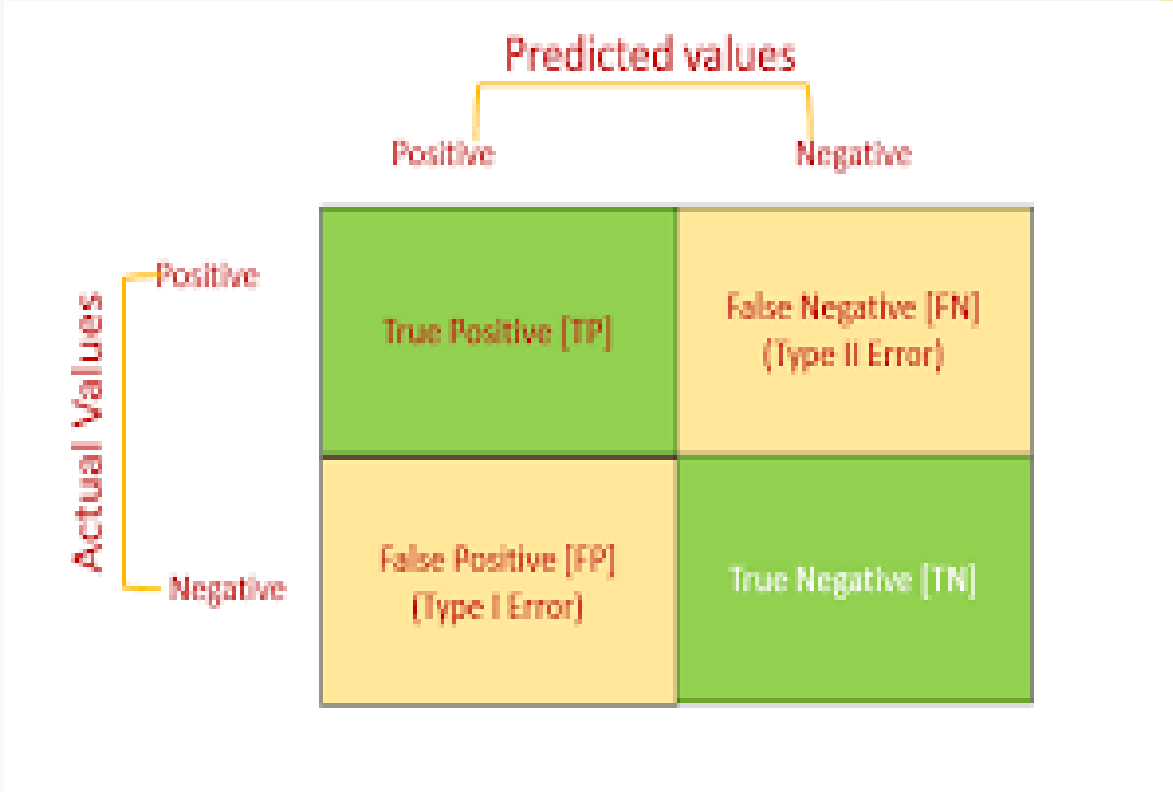
scikit-learn(Python)

Classification

- Accuracy
- Precision
- Recall
- F1-score
- Confusion matrix
- log-loss (logarithmic loss)
- AUC (area under the curve)

confusion matrix:





Accuracy

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.

For instance, a medical misdiagnosis that is a false positive (e.g., take a patient diagnosed with breast cancer when they do not have it) has substantially different consequences compared to a false negative, whereby a patient is told that they do not have breast cancer when in fact they do.

A confusion matrix breaks down the correct and incorrect classifications made by the model and attributes them to the appropriate label.

As a result, accuracy can be rewritten as the following:
Accuracy = (correctly predicted observation)/(total observation) =
(TP + TN)/(TP + TN + FP + FN)

Per-class accuracy is an extension of accuracy that takes into account the accuracy of each class. As a result, the preceding example has a per-class accuracy of (80% + 40%)/2 = 60%.

From the confusion matrix, it is determined that the positive class has greater accuracy than the negative class.
The accuracy of the positive classification is 20/25 = 80%.
The negative class has an accuracy of 10/25 = 40%.

Both metrics differ from the overall accuracy of the model, which would be determined as (20 + 10)/50 = 60%.

It is apparent how a confusion matrix adds more detail to the overall accuracy of a machine learning model.

Per-class accuracy 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.

Table 5-1. *Confusion matrix*

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

precision/recall?

Precision evaluates how many items are truly relevant compared to the total number of items correctly classified.

Recall evaluates how many items are predicted to be relevant by the model from the items that are relevant.

• Precision: (correctly predicted Positive)/(total predicted Positive) = TP/TP + FP

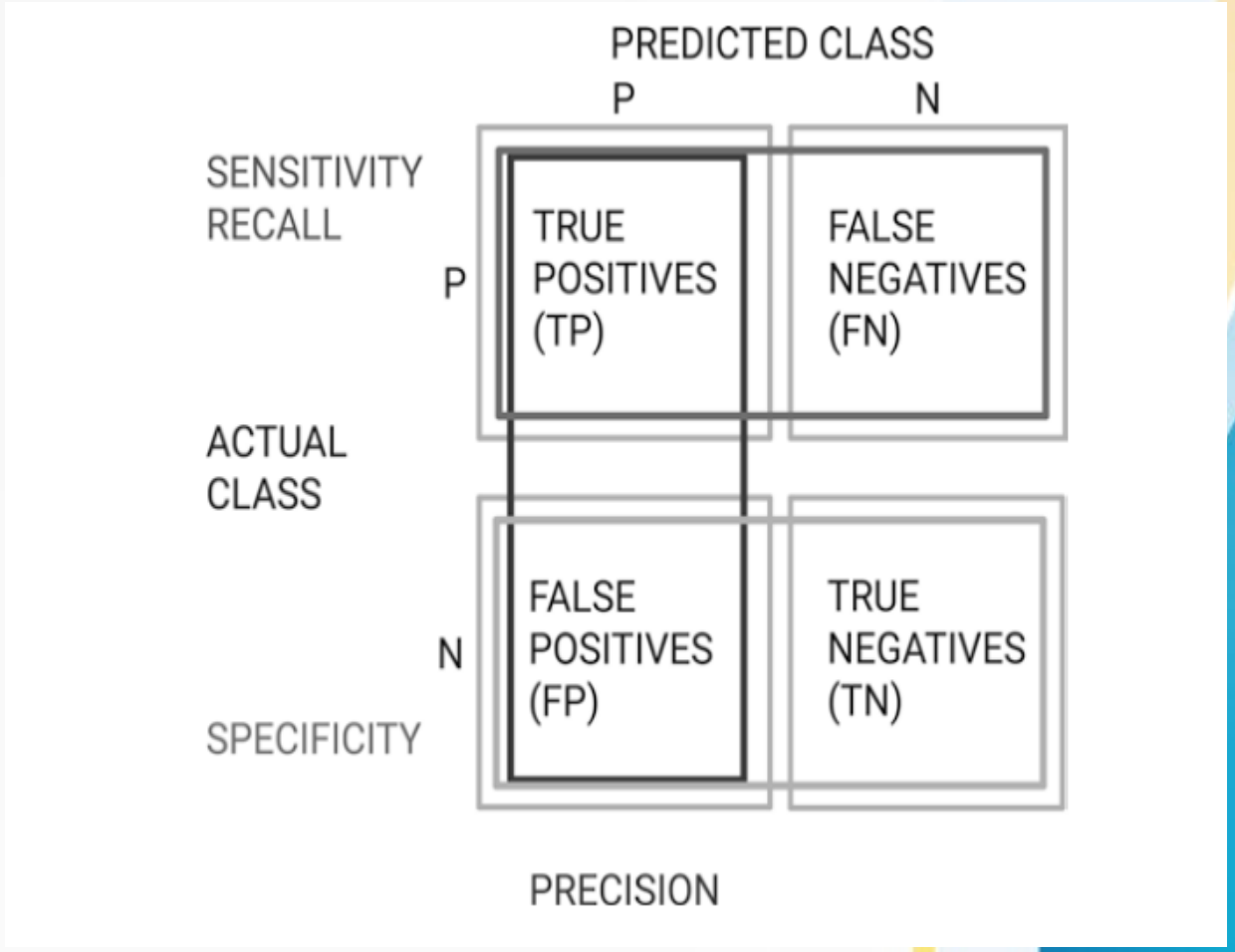
• Recall: (correctly predicted Positive)/(total correct Positive observation) = TP/TP + FN

Specificity refers to how well the model performs at returning incorrect

• Specificity: (correctly predicted Negative)/(total Negative observation) = TN/TN + FP

• Sensitivity?

- Figure 5-3. Specificity classification diagram



F-measure

- Imagine you have developed a disease detection model:
- - **Precision:** If the model predicts "this person is sick," how likely is it that they are truly sick?
- - **Recall (Sensitivity):** Out of all the actual sick individuals, how many has the model successfully identified?
- - **Specificity:** Out of all the healthy individuals, how many has the model correctly classified as healthy?

F-measure goes beyond the arithmetic mean and calculates the harmonic mean of precision and recall:

Where p denotes precision and r denotes recall.

F-1 Score= $$F = \frac{1}{\frac{1}{2}\left(\frac{1}{p}+\frac{1}{r}\right)} = \frac{2pr}{p+r}$$

Logarithmic loss

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 follows:

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

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

Assume your model predicts the probability of a dog being present in an image:

Image 1: True label $y = 1$ (dog is present), predicted probability $p = 0.9$

Logarithmic loss: $-\log(0.9) = 0.105$ (low loss, good prediction).

Image 2: True label $y = 1$ (dog is present), predicted probability $p = 0.3$

Logarithmic loss: $-\log(0.3) = 1.204$ (high loss, poor prediction).

Conclusion: The closer the predicted probability is to the true label, the lower the Log-Loss.

AUC or ROC?

- The AUC (area under curve)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.

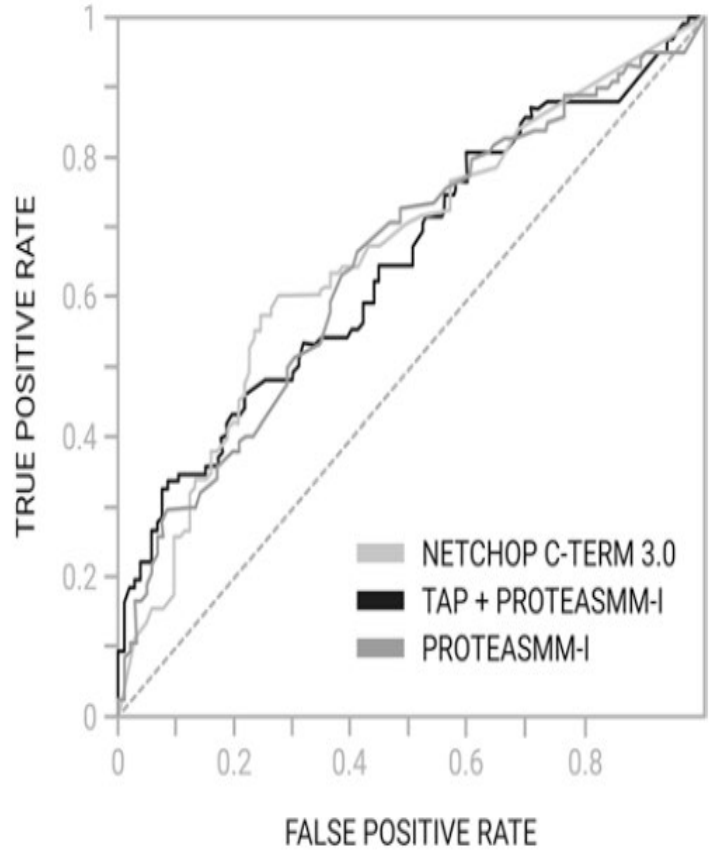
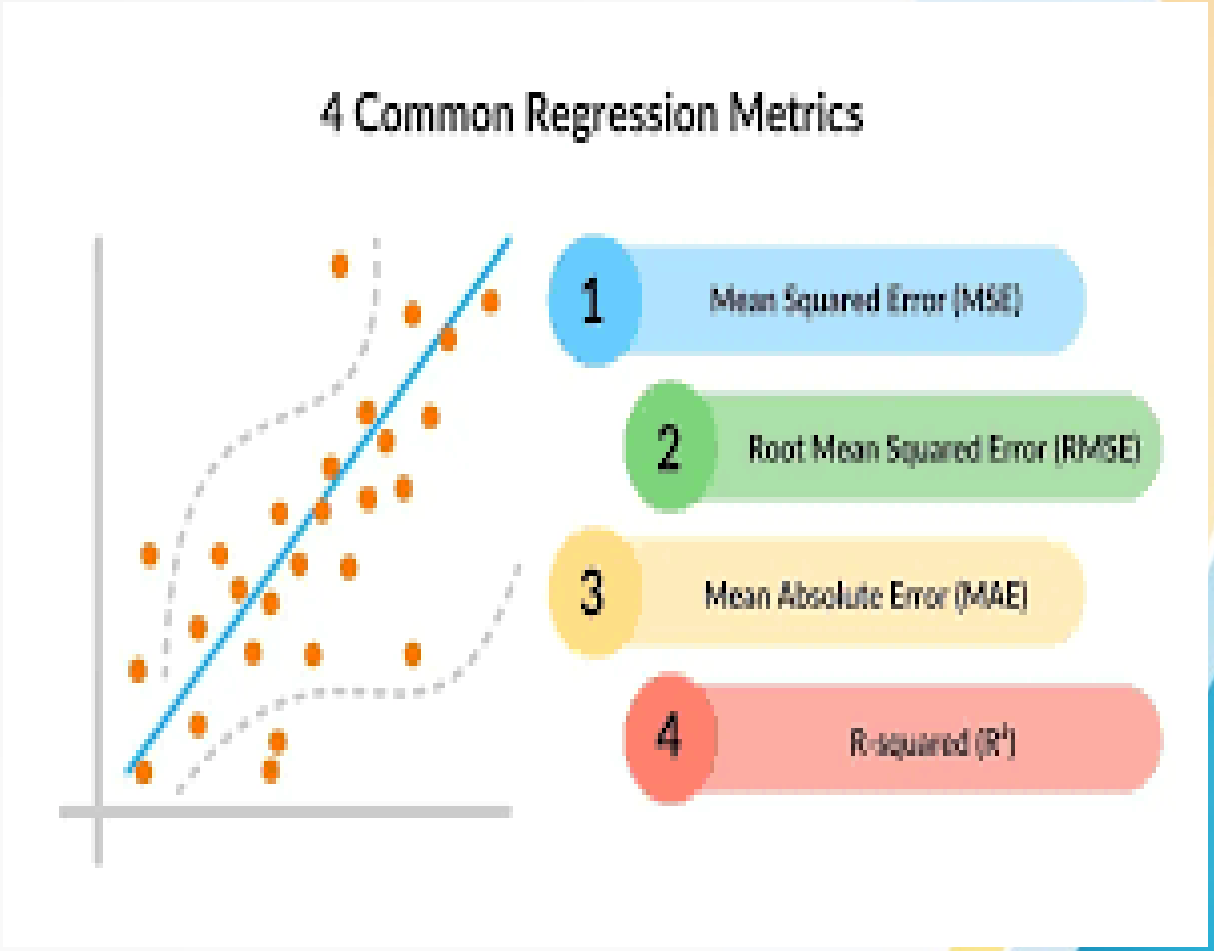


Figure 5-2. *ROC curve*

Regression

- root-mean-squared error (RMSE)
- mean-squared error (MSE)
- mean-absolute error (MAE)
- median absolute percentage error (MAPE)
- R2



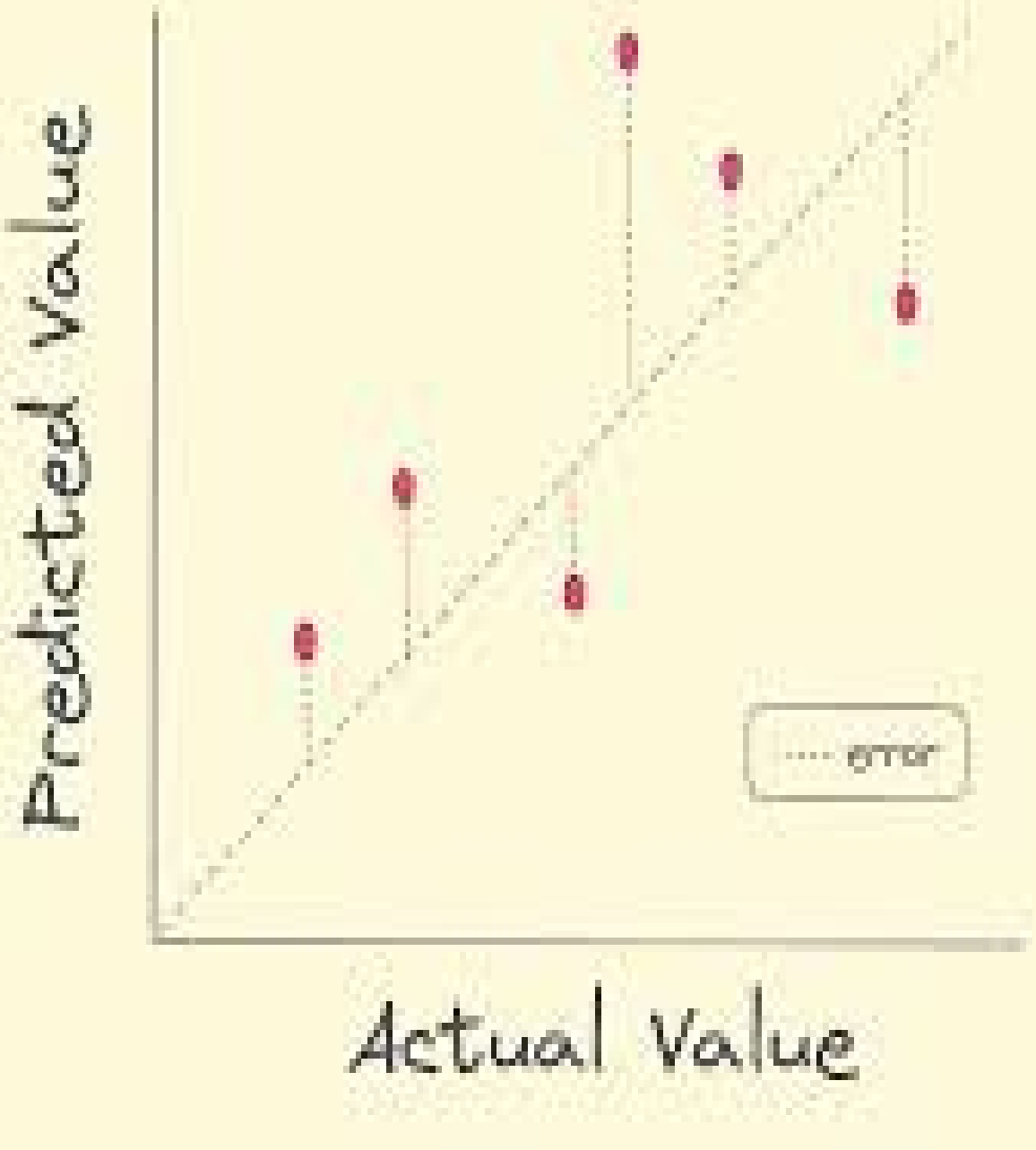
$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

$$R^2 = 1 - \frac{SSR}{SST} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$



- **RMSE**

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.

$$RMSE = \sqrt{(1/n) * \Sigma(yi - \hat{y}i)^2}$$

- **MAE**

$$MAE = (1/n) * \Sigma|yi - \hat{y}i|$$

- **MSE**

$$MSE = (1/n) * \Sigma(yi - \hat{y}i)^2$$

Percentiles of errors

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.

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

- $R^2 = 1 - (SSR/SST) = 1 - \Sigma(yi - \hat{y}i)^2 / \Sigma(yi - \bar{y})^2$

- SSR is the sum of squared residuals
- SST is the total sum of squares
- yi is the actual value
- $\hat{y}i$ is the predicted value
- $\bar{y}$ is the mean of the actual values

- Mean Absolute Percentage Error:

- $MAPE = (1/n) * \Sigma(|yi - \hat{y}i| / |yi|) * 100\%$

 # dealing with datasets

An experienced data scientist treats all data with suspicion.

- ✓ Skewed Datasets,Anomalies,Rare Data,inconsistent,imbalanced class examples, 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.
- ✓ The effect of large outliers can be mitigated using percentiles of error.
- ✓ good data cleansing, removal of outliers, and normalization of variables

Parameters and Hyperparameters tuning

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.

- ✓ Model hyperparameters are parameters to the model building process that are not learned during training.
- ✓ 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.

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

Tuning Hyperparameters

- Hyperparameter tuning or optimization is the task of selecting a set of optimal hyperparameters for a machine learning model.
- Optimized hyperparameters values maximize a model's predictive accuracy.
- Hyperparameters are optimized through running training a model, assessing the aggregate accuracy, and appropriately adjusting the hyperparameters.
- Through trialing a variety of hyperparameter values, the
- best hyperparameters for the problem are determined, which improves overall model accuracy.

Hyperparameter Tuning Algorithms

Hyperparameter tuning is like training a machine learning model.

The ask at hand is one of optimization.

Model parameters can be expressed as a loss function, whereas hyperparameters cannot be expressed as such, as it depends entirely on the model training process.

There are several approaches to hyperparameter tuning, with the most common being

grid search and random search.

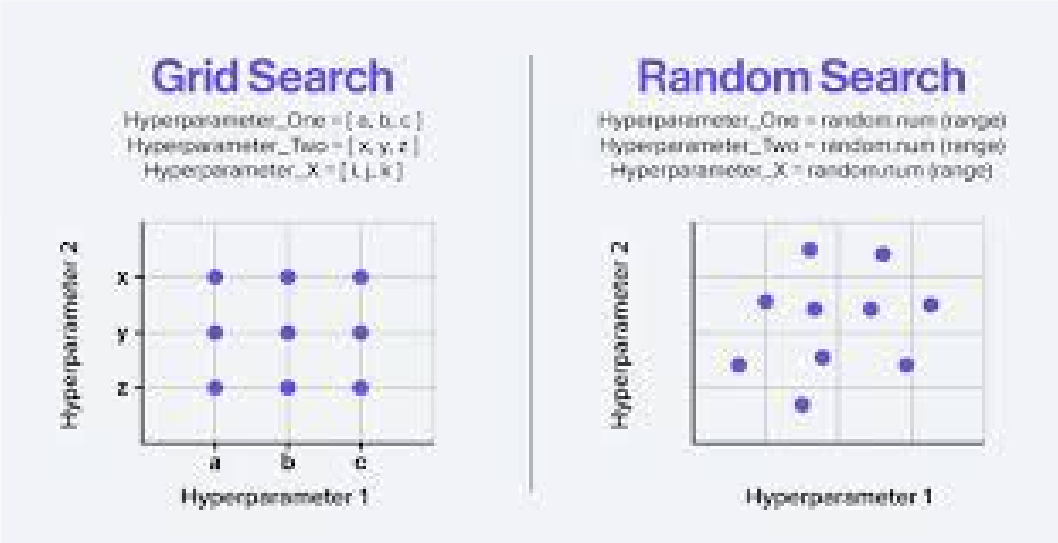
Grid Search

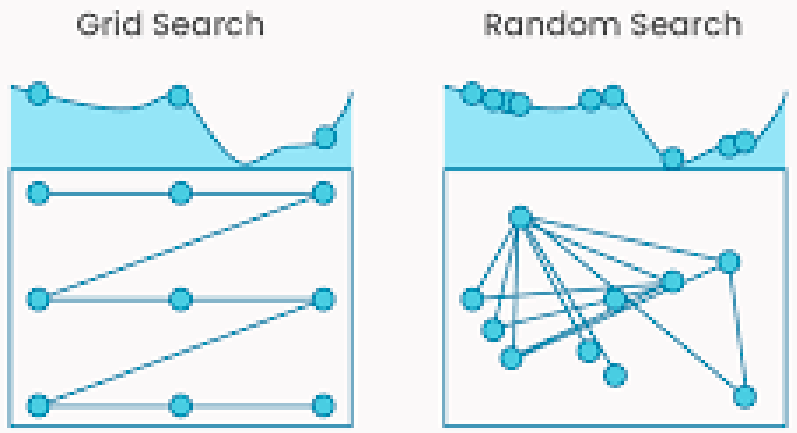
The grid search is a simple, effective, yet resource expensive hyperparameter optimization technique that evaluates a grid of hyperparameters. The method evaluates each hyperparameter and determines the winner. For example, if the hyperparameter were the number of leaves in a decision tree, which could be anywhere from n = 2 to 100, grid search would evaluate each value of n (i.e., points on the grid) to determine the most effective hyperparameter.
It is often a case of guessing where to start with hyperparameters, including minimum and maximum values. The approach is typical of trial and error, whereby if the optimal value lies toward either maximum or minimum, the grid would be expanded in the appropriate direction in an attempt to further optimize the model's hyperparameters.

Random Search

Random search is a variant of grid search that evaluates a random sample of grid points. Computationally, this is far less expensive than a standard grid search. Although at first glance it would appear that this is not as useful in finding optimal hyperparameters.The simplicity and better-than-expected performance of a random search means that it is often chosen over grid search. Both grid search and random search are parallelizable.
More intelligent hyperparameter tuning algorithms are available that are computationally expensive as the result of evaluating which samples to try next. These algorithms often have hyperparameters of their own.

Bayesian optimization, random forest smart tuning, and derivative-free optimization are three examples of such algorithms.





Statistics

Multivariate Testing

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?

Choosing the appropriate metric to evaluate your model depends on the use case.

Consider the impact of false positives, false negatives, and the consequences of such predictions.

Furthermore, if a model is attempting to predict an event that only happens 0.001% of the time, an accuracy of 99.999% can be reported but not confirmed.

Build the model to cater to the appropriate metrics.

One approach is to repeat the experiment, thus performing repeat evaluations.

Although not a fail-safe, this reduces the change of illusionary results.

If there is indeed change between the null and alternate hypothesis, the difference will be confirmed.

Correlation Does Not Equal Causation

The phrase correlation does not equal causation is used to stress that a correlation between two variables does not suggest that one causes the other.

Correlation refers to the size and direction of a relationship between two or more variables.

Causation, also known as cause and effect, emphasizes that the occurrence of one event is related to the presence of another event.

It may be tempting to assume that one variable causes the other; however, in models with several features, there may be hidden factors that cause both

variables to move in tandem.For instance, smoking tobacco is a cause that increases the risk of developing a variety of cancers.

However, it may be correlated with alcoholism, but it does not cause alcoholism

What Amount of Change Counts as Real Change?

Defining the amount of change required before the null hypothesis is rejected once again depends on the use case.

Specify a value at the beginning of the project that would be satisfactory and adhere to it.

Types of Tests, Statistical Power, and Effect Size

There are two main types of tests—one-tailed and two-tailed tests.

One tailed tests evaluate whether the new model is better than the original.
However, it does not specify whether the model is worse than the baseline.
One-tailed tests are thus inherently biased.

With two-tailed tests, the model is tested for the possibility of change in two directions—positive and negative.

Statistical power refers to the probability that the difference detected during the testing reflects a real-world difference.
Effect size determines the difference between two groups through evaluating the standardized mean difference between two sets.
Effect size is calculated as the following:
Effect size = ((mean of experiment group) – (mean of control group))/standard deviation

Checking the Distribution of Your Metric

Many multivariate tests use the t-test to analyze the statistical difference between means.
The t value evaluates the size of the difference relative to the variation in your sample data.

However, the t-test makes assumptions that are not necessarily satisfied by all metrics.
For instance, the t-test assumes both sets have a normal, or Gaussian, distribution.
If the distribution does not appear to be Gaussian, select a nonparametric test that does not make assumptions about a Gaussian distribution, such as the Wilcoxon–Mann–Whitney test.

Determining the Appropriate p Value

Statistically speaking, the p value is a calculation used in hypothesis testing that represents the strength of the evidence.

The p value measures the statistical significance, or probability, that a difference would arise by chance given there was no real difference between two populations.

It provides the evidence against the null hypothesis and is a useful metric for stakeholders to draw conclusions from.

A p value lies between 0 and 1, and is interpreted as follows:

- a p value of ≤ 0.05 indicates strong evidence against the null hypothesis, thus rejecting the null hypothesis
- a p value of > 0.05 indicates weak evidence against the null hypothesis, hence maintaining the null hypothesis
- a p value near 0.05 is considered marginal and could swing either way

The smaller the p value, the smaller the probability that the results are down to chance.

How Many Observations Are Required?

The quantity of observations required is determined by the statistical power demanded by the project. Ideally, this should be determined at the beginning of the project.

Data Variance

The control and experimentation sets could be biased as the result of not being split at random.
This may result in biases in the sample data.
If this is the case, other tests can be used, such as Welch's t-test, which does not assume equal variance.

How Long to Run a Multivariate Test?

The duration of time required for your multivariate testing is ideally the amount of time required to capture enough observations to meet the defined statistical power.

It is often useful to run tests over time to capture a representative, variable sample.

When determining the duration of your testing phase, consider the novelty effect, which describes how user reactions in the short term are not representative of the long-term reactions. For instance, whenever Facebook updates their news feed layout or design, there is an uproar.

However, this soon subsides once the novelty effect has worn off. Therefore, it is useful to run your experiment for long enough to overcome this bias.

Running multivariate tests for long periods of time are typically not a problem in model optimization.

Spotting Distribution Drift

It is key to measure ongoing performance of your machine learning model once deployed. Data drifts and system development require the model to be confirmed against the baseline.

Typically, this involves monitoring the offline performance, or validation metric, against data from the live, deployed model.

If there is a sizeable change in the validation metric, this highlights the need to revise the model through training on new data. This can be done manually or automated to ensure consistent reporting and confidence in the model.

Keep a Note of Model Changes

Keep a log of all changes to your machine learning model with notes on changes. Not only does this serve as a change log for stakeholders, it provides a physical record of how the system has changed over time.

The use of versioning software within a development enviornment (test/staging to live deployment) will enable software changes to automatically be noted. Versioning software provides a form of technical governance and can be used to deploy software with extensive rollback and backup facilities.

AIHRC

THANK YOU

By: Dr.Fatemeh Abedi