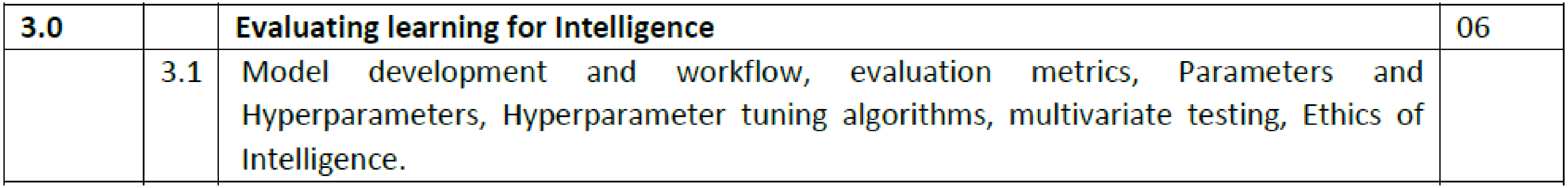
HAIMLC701 AI & ML in Healthcare



3.0		**Evaluating learning for Intelligence**	06
	3.1	Model development and workflow, evaluation metrics, Parameters and Hyperparameters, Hyperparameter tuning algorithms, multivariate testing, Ethics of Intelligence.	

Evaluating learning for Intelligence

- The most laborious tasks within a machine learning
    - identifying the appropriate model and engineering features, which make a substantial difference to the output of the model
- it is important to evaluate the learning algorithm that will determine the model's intelligence to predict the output of an unknown sample
    - This is usually done using various metrics

Model Development and Workflow

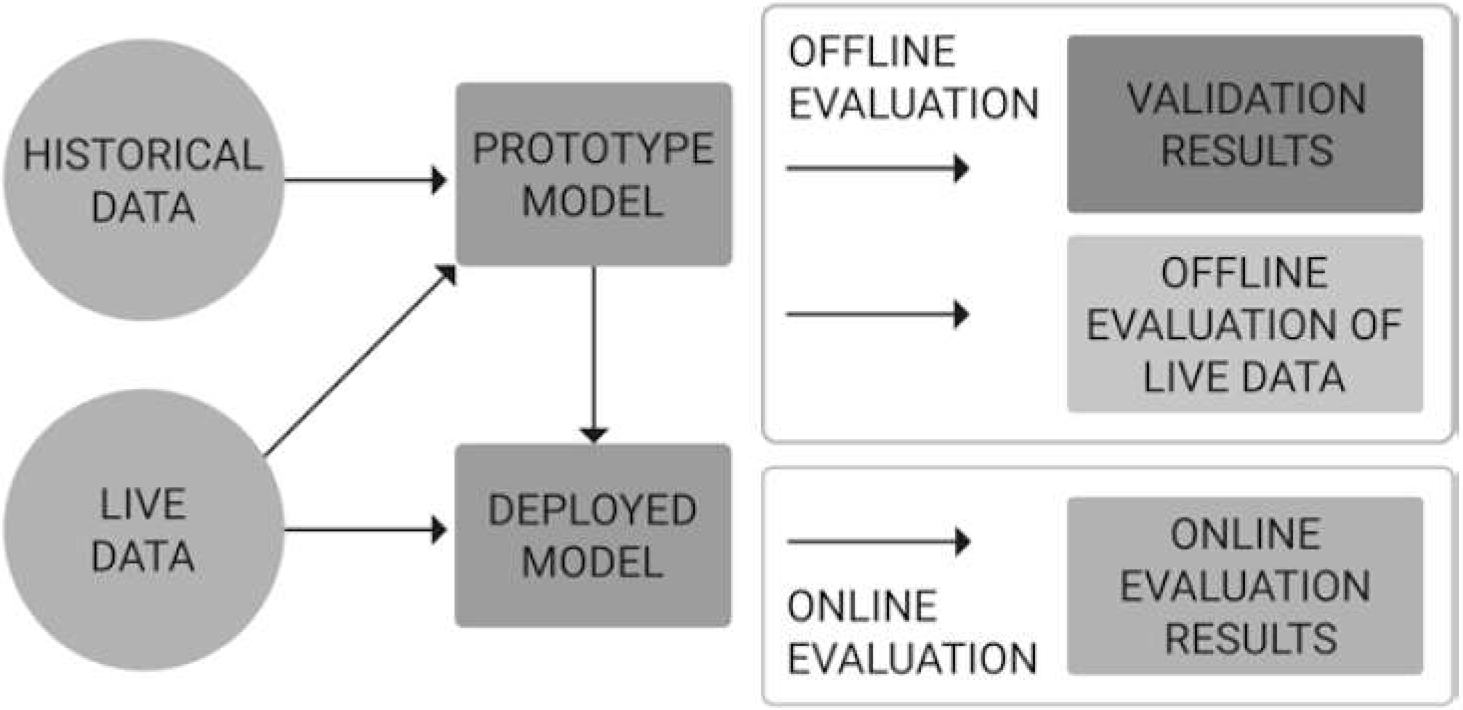


Figure 5-1. *Model development and work flow*

Model development and workflow

- Real time training examples
  - credit card fraud detection, recommendation systems, and loan application approvals
- Offline example applications
  - churn predictions, customer segmentation, and sales forecasting tools
- Online training involves adjusting model parameters as new data arrives, can be challenging to implement
  - involves downloading real-time logs to generate training data, performing offline training, and uploading updated model parameters to the online parameter service, all within minutes to hours

Model Development and Workflow

- Prototype phase
  - A prototype is created through testing various models on historical data to determine the best model
  - The best prototype model chosen is tested and validated
  - Validating a model requires splitting datasets into training, testing, and validation sets
  - The model is then usually evaluated by one (or several) performance metrics
- Two ways of evaluating a machine learning model:
  - Offline evaluation
    - accuracy and precision-recall hold-back method and n-fold cross-validation
  - online (or live) evaluation
    - Online evaluation refers to the evaluation of metrics once the model is deployed
    - 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

Evaluation Metrics

- Accuracy, precision-recall, confusion matrices, log-loss (logarithmic loss), and AUC (area under the curve)

Confusion Matrix:

	Predicted Class	
Actual Class	C_1	¬ C_1
C_1	**True Positives (TP)**	**False Negatives (FN)**
¬ C_1	**False Positives (FP)**	**True Negatives (TN)**

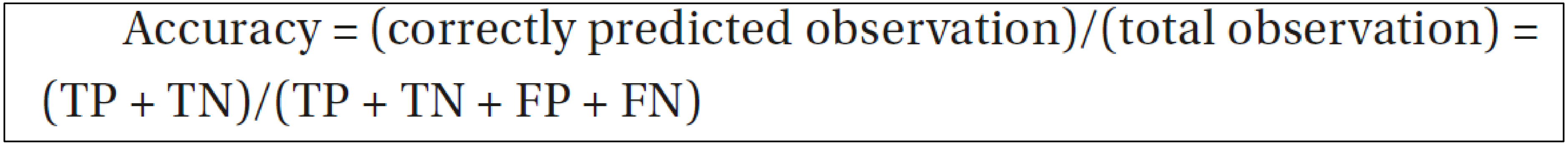
Example of Confusion Matrix:

Actual class	Predicted Class	
	Has_diabetes = yes	has_diabetes= no
has_diabetes = yes	**6954**	**46**
has_diabetes = no	**412**	**2588**

Evaluation Metrics

- Confusion matrix
- A confusion matrix breaks down the correct and incorrect classifications made by the model and attributes them to the appropriate label.
- True positive: Where the actual class is yes and the predicted class is yes.
- False positive: Actual class is no, and predicted class is yes
- True negative: The value of the actual class and the predicted class is no
- False negative: When the actual class value is yes, but predicted class is no

- 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 = (correctly predicted observation)/(total observation) = (TP + TN)/(TP + TN + FP + FN)

Evaluation Metrics

- Accuracy = (correctly predicted observation)/(total observation) =
  - (TP + TN)/(TP + TN + FP + FN)
- 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%.

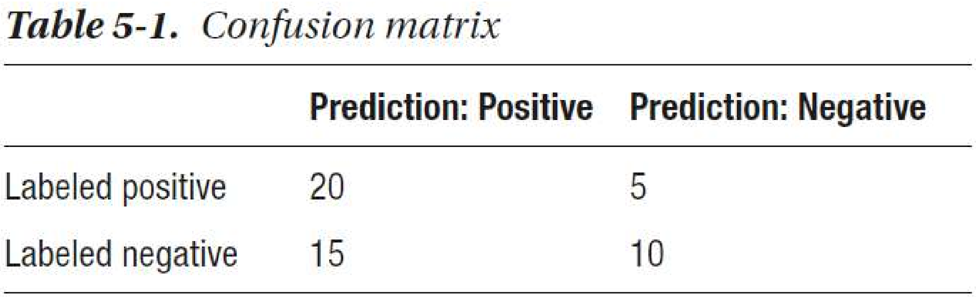


Table 5-1. Confusion matrix

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

Per-class accuracy

extension of accuracy that takes into account the accuracy of each class.
per-class accuracy : (80% + 40%)/2 = 60%.
Per-class accuracy is useful in distorted problems where there are a larger number of examples within one particular class compared to another

Classifier Evaluation Metrics

Actual class\Predicted class	Has_diabetes = yes	has_diabetes= no
has_diabetes = yes	**6954**	**46**
has_diabetes = no	**412**	**2588**

- **Class Imbalance Problem**:
  - One class may be *rare*, e.g. fraud, or HIV-positive
  - Significant *majority of the negative class* and minority of the positive class

- **Specificity**:
- True Negative recognition rate

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

=2588/3000

=86.26

Classifier Evaluation Metrics

Actual class\Predicted class	Has_diabetes = yes	has_diabetes= no
has_diabetes = yes	**6954**	**46**
has_diabetes = no	**412**	**2588**

- **Sensitivity**: (RECALL)
- Sensitivity measures the total number of actual positive cases that were correctly predicted.
- True Positive recognition rate

Sensitivity =6954/(6954+46)

=99.34%

- **Class Imbalance Problem**:
  - One class may be *rare*, e.g. fraud, or HIV-positive
  - Significant *majority of the negative class* and minority of the positive class

$$\text{Sensitivity} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative (FN)}}$$

Recall: completeness – what % of positive tuples did the classifier label as positive?

$$\boxed{recall = \frac{TP}{TP + FN} = \frac{TP}{P}}$$

Classifier Evaluation Metrics: Precision and Recall, and F-measures

Precision: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$\boxed{precision = \frac{TP}{TP + FP}}$$

Precision is the likelihood that a given positive prediction is actually positive, specificity is the likelihood that a given negative prediction is actually negative.

***F* measure (*F_1* or *F*-score)**: harmonic mean of precision and recall

$$\boxed{F = \frac{2 \times precision \times recall}{precision + recall}}$$

Evaluation Metrics

- Log loss, also known as logarithmic loss or cross-entropy loss, is a common evaluation metric for binary classification models
- It measures the performance of a model by quantifying the difference between predicted probabilities and actual values
- Log-loss is indicative of how close the prediction probability is to the corresponding actual/true value (0 or 1 in case of binary classification), penalizing inaccurate predictions with higher values
- Lower log-loss indicates better model performance

Evaluation Metrics

- Logarithmic loss
  - used for problems where a continuous probability is predicted rather than a class label
  - provides a probabilistic measure of the confidence of the accuracy considers the entropy between the distribution of true labels and predictions
  - For a binary classification problem, the logarithmic loss would be calculated as follows:

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

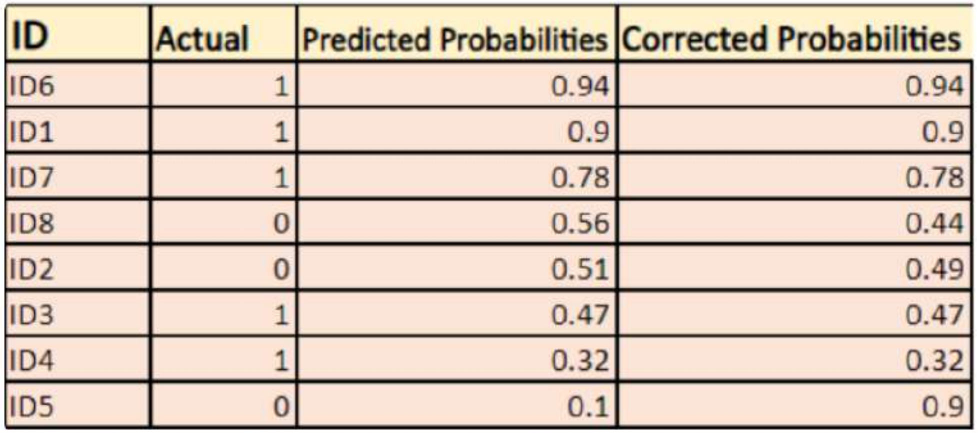
Where Pi is the probability of the *i*th data point belonging to a class and

- yi the true label (either 0 or 1)
- Here Yi represents the actual class and log(p(yi)is the probability of that class
- p(yi) is the probability of 1.
- 1-p(yi) is the probability of 0.

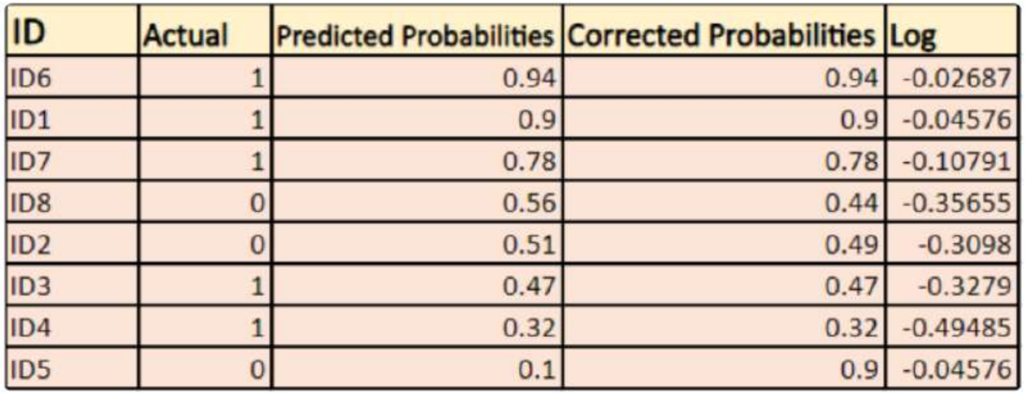
Logarithmic loss

There are three steps to find Log Loss:

1. To find corrected probabilities.
2. Take a log of corrected probabilities.
3. Take the negative average of the values we get in the 2nd step.



ID	Actual	Predicted Probabilities	Corrected Probabilities
ID6	1	0.94	0.94
ID1	1	0.9	0.9
ID7	1	0.78	0.78
ID8	0	0.56	0.44
ID2	0	0.51	0.49
ID3	1	0.47	0.47
ID4	1	0.32	0.32
ID5	0	0.1	0.9



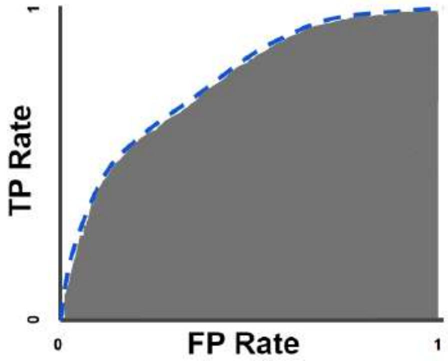
ID	Actual	Predicted Probabilities	Corrected Probabilities	Log
ID6	1	0.94	0.94	-0.02687
ID1	1	0.9	0.9	-0.04576
ID7	1	0.78	0.78	-0.10791
ID8	0	0.56	0.44	-0.35655
ID2	0	0.51	0.49	-0.3098
ID3	1	0.47	0.47	-0.3279
ID4	1	0.32	0.32	-0.49485
ID5	0	0.1	0.9	-0.04576

As you can see these log values are negative. To deal with the negative sign, we take the **negative average of these values**, to maintain a common convention that lower loss scores are better.

$$log\ loss = \ -1/N \sum_{i=1}^{N} (log\,(Pi\,))$$

ROC(Receiver Operating Characteristics)

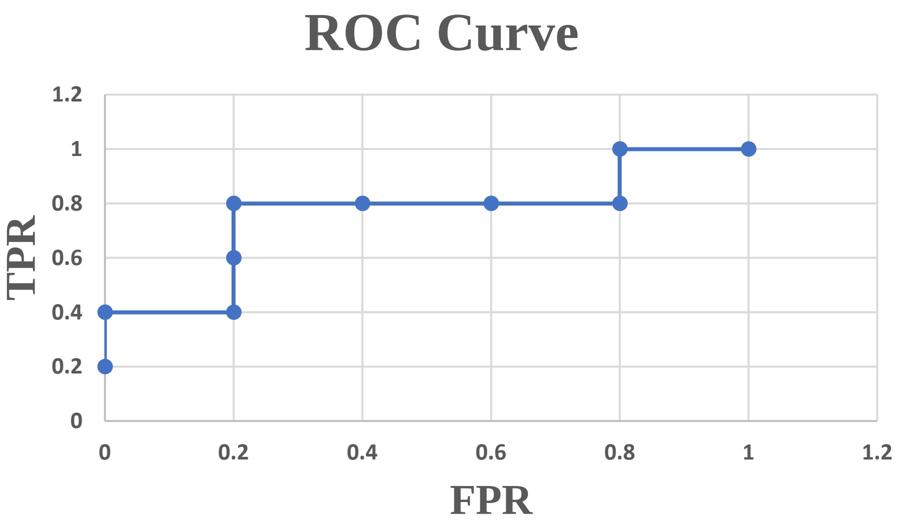
- It is an evaluation metric used for binary classification problems
- A probability curve that depicts the TPR (rate of true positives) on the y-axis against the FPR (rate of false positives) on the x-axis
- An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds.
- This curve plots two parameters:
  - True Positive Rate
  - False Positive Rate
- **True Positive Rate** (**TPR**) is a synonym for recall and is therefore defined as follows:
  - TPR=TP/(TP+FN)
  - TPR=TP/P
- **False Positive Rate** (**FPR**) is defined as follows:
  - FPR=FP/(FP+TN)
  - FPR=FP/N



ROC curve

Sample	Class	Probability	True Positive	False Positive	TPR	FPR
1	P	0.9	1	0	0.2	0
2	P	0.8	2	0	0.4	0
3	N	0.7	2	1	0.4	0.2
4	P	0.6	3	1	0.6	0.2
5	P	0.55	4	1	0.8	0.2
6	N	0.54	4	2	0.8	0.4
7	N	0.53	4	3	0.8	0.6
8	N	0.51	4	4	0.8	0.8
9	P	0.5	5	4	1	0.8
10	N	0.4	5	5	1	1

No. of positive samples=5
No. of negative samples=5
Assume class probability as classification threshold.

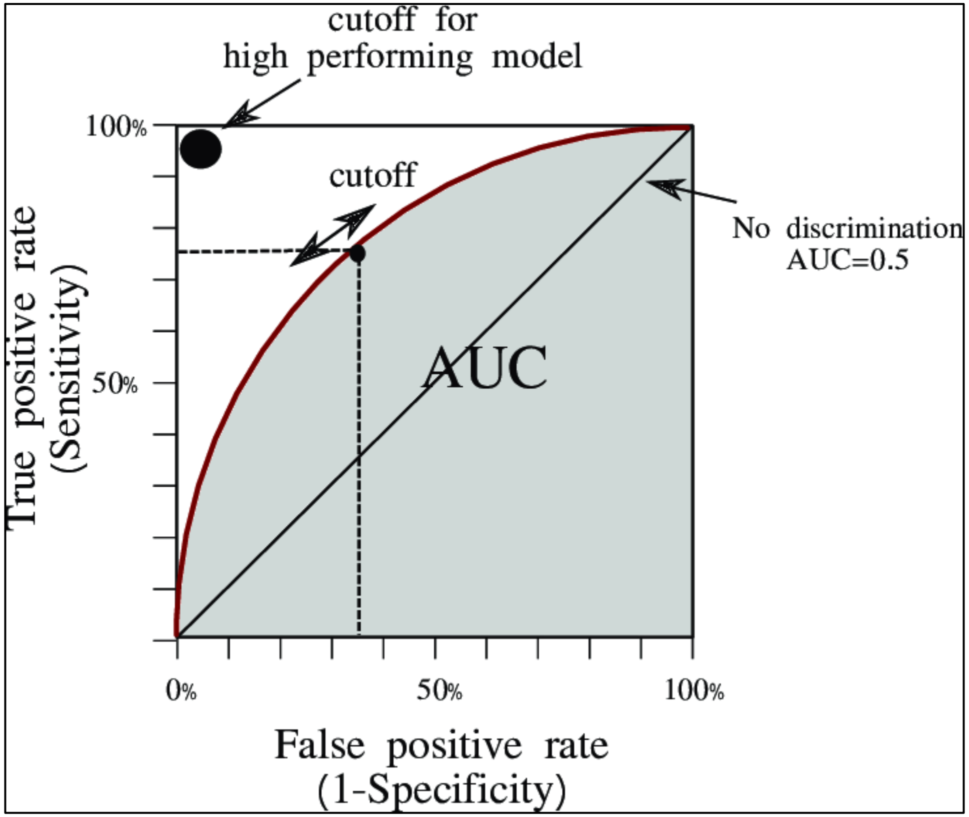


- AUC ranges in value from 0 to 1.
- A model whose predictions are 100% wrong has an AUC of 0.0
- one whose predictions are 100% correct has an AUC of 1.0

AUC (Area Under the Curve)

- 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
- 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.
- Measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).



Evaluation Metrics - 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

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

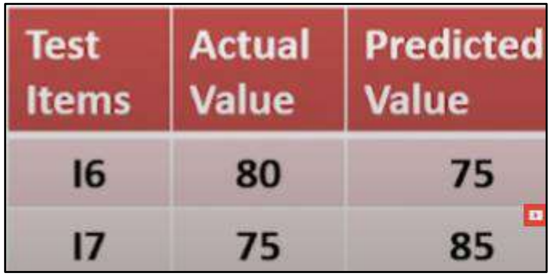
- where *yi* denotes the actual value and ˆ*yi* denotes predicted value.

Evaluation Metrics – MAE, MSE, RMSE

MAE –Mean Absolute Error

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=0}^{n-1}\left|y_i - \hat{y}_i\right|$$

$$\mathrm{MAE} = \frac{1}{2}\times\left(\left|80-75\right| + \left|75-85\right|\right) = \frac{15}{2} = 7.5$$



Test Items	Actual Value	Predicted Value
I6	80	75
I7	75	85

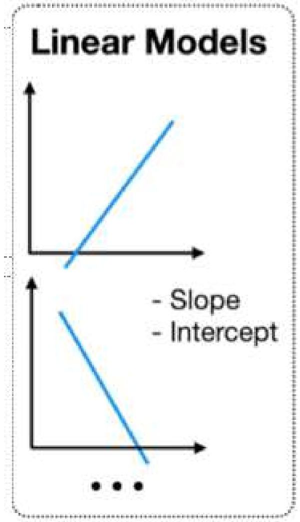
MSE –Mean Squared Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2$$

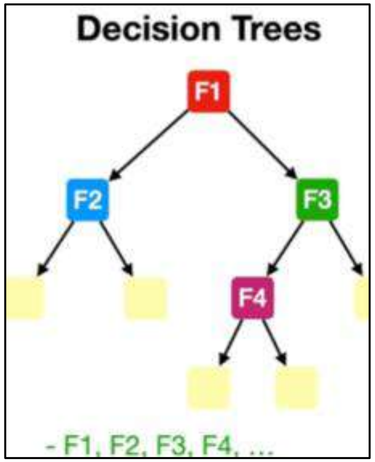
$$\mathrm{MSE} = \frac{1}{2}\times\left(\left|80-75\right|^2 + \left|75-85\right|^2\right) = \frac{125}{2} = 62.5$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} = \sqrt{62.5} = 7.91$$

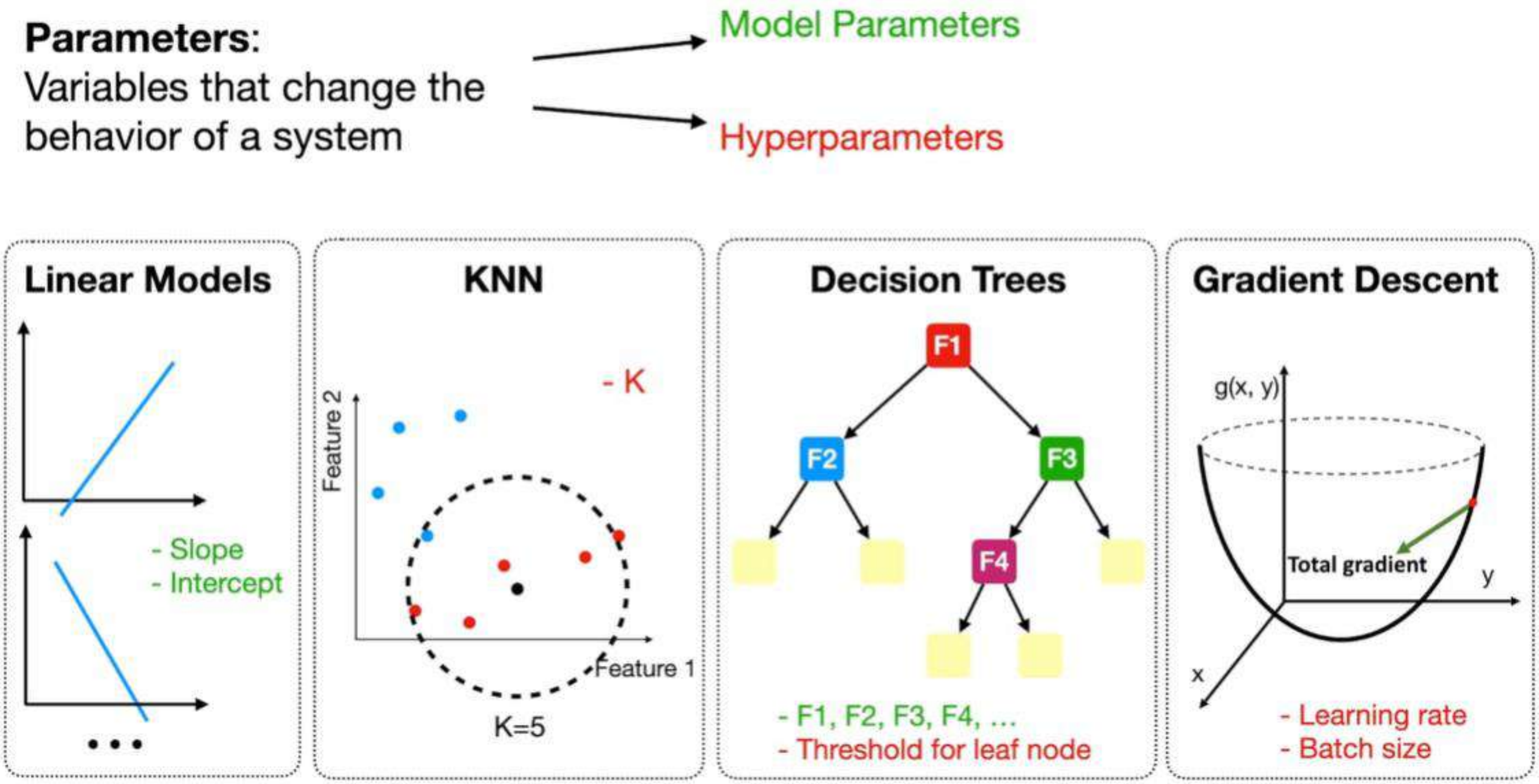
Parameters and Hyperparameters



- Model parameters
  - configuration variables that are internal to the model, and a model learns them on its own
  - A parameter is a variable that is learned from the data during the training process
  - It is used to represent the underlying relationships in the data and is used to make predictions on new data.
  - Example:
    - W Weights or Coefficients of independent variables in the Linear regression model
    - Weights or Coefficients of independent variables SVM
    - weight, and biases of a neural network



Model and Hyper parameters



Parameters:
Variables that change the behavior of a system

Model Parameters

Hyperparameters

Linear Models

- Slope
- Intercept

KNN

- K

Decision Trees

- F1, F2, F3, F4, …
- Threshold for leaf node

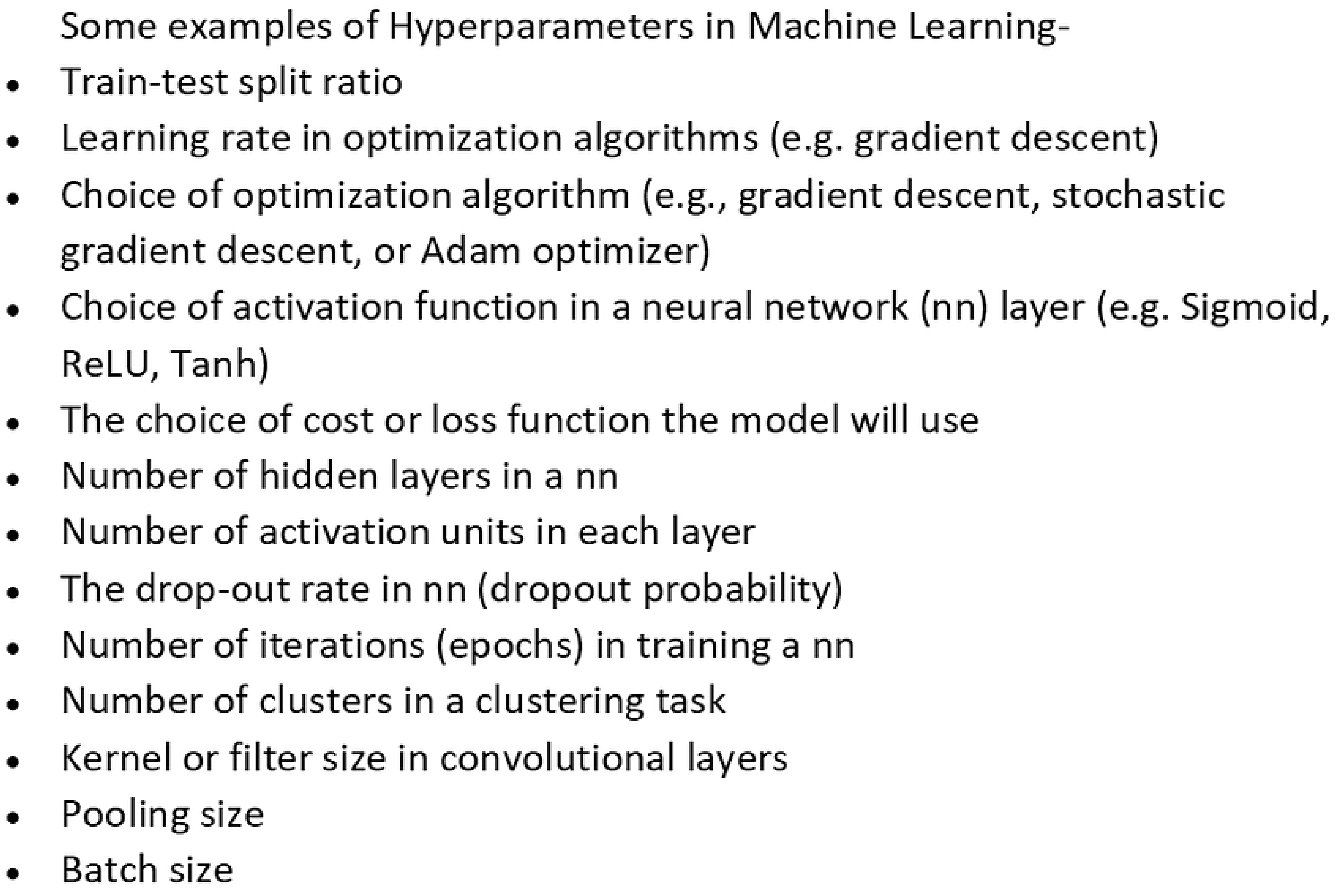
Gradient Descent

- Learning rate
- Batch size

Hyper parameter

- Hyperparameters are often used to tune the performance of a model, and they can have a significant impact on the model's accuracy, generalization, and other metrics.
- parameters that are explicitly defined by the user to control the learning process
- The best value can be determined for the given problem either by the rule of thumb or by trial and error
- not learned from the data but are instead set by the user or determined through a process known as hyperparameter optimization
- Examples:
  - Decision trees hyperparameters would include the desired depth and number of leaves in the tree
  - support vector machine would include a misclassification penalty term
  - In a neural network
    - Learning rate in gradient descent
    - Number of iterations in gradient descent
    - Number of layers in a Neural Network
    - Number of neurons per layer in a Neural Network
    - Number of clusters(k) in k means clustering

Hyper parameters



Some examples of Hyperparameters in Machine Learning-

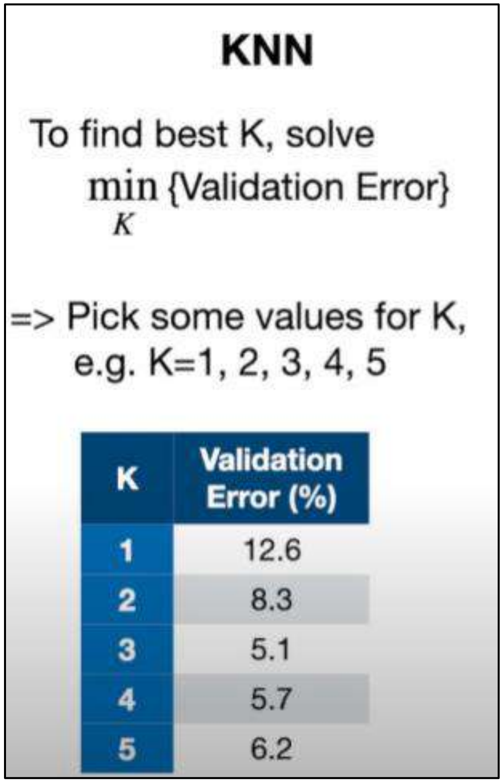
- Train-test split ratio
- Learning rate in optimization algorithms (e.g. gradient descent)
- Choice of optimization algorithm (e.g., gradient descent, stochastic gradient descent, or Adam optimizer)
- Choice of activation function in a neural network (nn) layer (e.g. Sigmoid, ReLU, Tanh)
- The choice of cost or loss function the model will use
- Number of hidden layers in a nn
- Number of activation units in each layer
- The drop-out rate in nn (dropout probability)
- Number of iterations (epochs) in training a nn
- Number of clusters in a clustering task
- Kernel or filter size in convolutional layers
- Pooling size
- Batch size

Hyper parameter Tuning

- Hyperparameter tuning is the process of selecting the optimal values for a machine learning model's hyperparameters.
- **Learning rate:**
  - controls the step size taken by the optimizer during each iteration of training
  - Too small a learning rate can result in slow convergence, while too large a learning rate can lead to instability and divergence
- **Epochs:**
  - This hyperparameter represents the number of times the entire training dataset is passed through the model during training
  - Increasing the number of epochs can improve the model's performance but may lead to overfitting if not done carefully
- **Number of layers:**
  - This determines the depth of the model, which can have a significant impact on its complexity and learning ability.
- **Number of nodes per layer:**
  - determines the width of the model, influencing its capacity to represent complex relationships in the data
- **Architecture:**
  - determines the overall structure of the neural network, including the number of layers, the number of neurons per layer, and the connections between layers
  - The optimal architecture depends on the complexity of the task and the size of the dataset
- **Activation function:**
  - introduces non-linearity into the model, allowing it to learn complex decision boundaries
  - Common activation functions include sigmoid, tanh, and Rectified Linear Unit (ReLU).

Hyper parameter Tuning

- 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 trialling a variety of hyperparameter values, the best hyperparameters for the problem are determined, which improves overall model accuracy



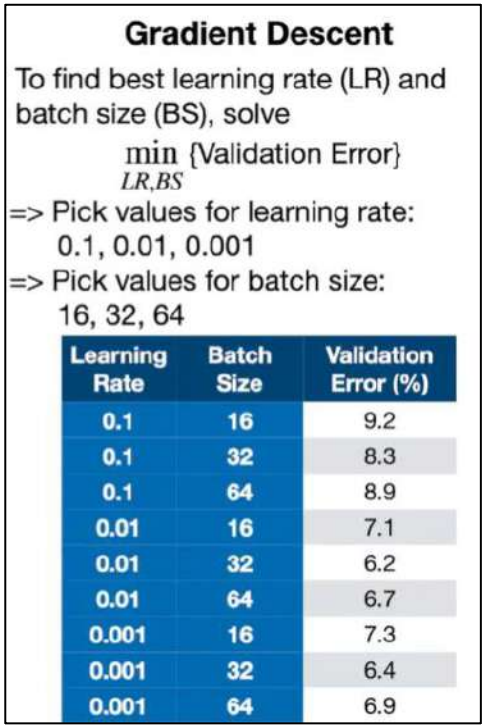
KNN

To find best K, solve

$$\min_{K} \{\text{Validation Error}\}$$

=> Pick some values for K, e.g. K=1, 2, 3, 4, 5

K	Validation Error (%)
1	12.6
2	8.3
3	5.1
4	5.7
5	6.2



Gradient Descent

To find best learning rate (LR) and batch size (BS), solve

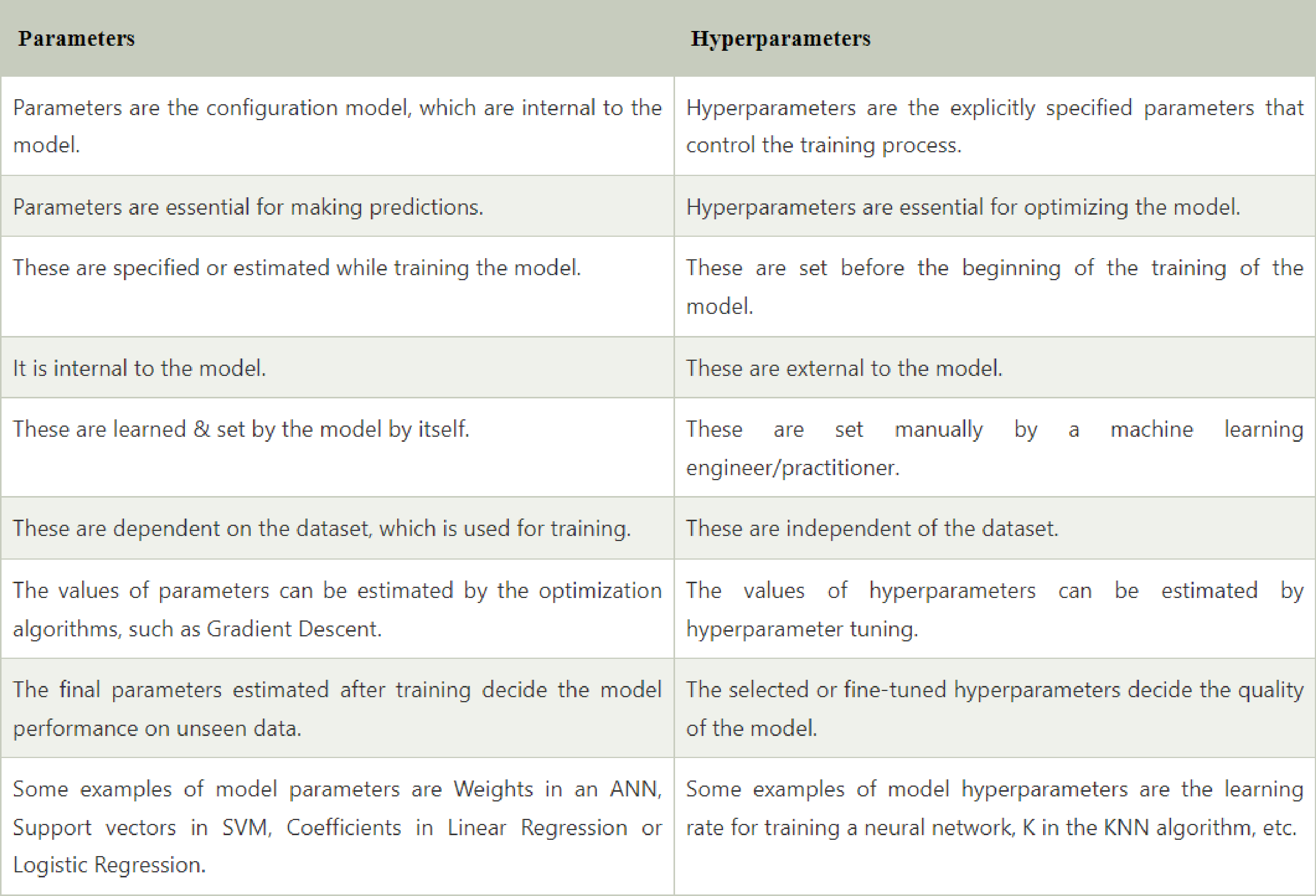
$$\min_{LR,BS} \{\text{Validation Error}\}$$

=> Pick values for learning rate: 0.1, 0.01, 0.001

=> Pick values for batch size: 16, 32, 64

Learning Rate	Batch Size	Validation Error (%)
0.1	16	9.2
0.1	32	8.3
0.1	64	8.9
0.01	16	7.1
0.01	32	6.2
0.01	64	6.7
0.001	16	7.3
0.001	32	6.4
0.001	64	6.9

Model parameters Vs Hyper parameters



Parameters	Hyperparameters
Parameters are the configuration model, which are internal to the model.	Hyperparameters are the explicitly specified parameters that control the training process.
Parameters are essential for making predictions.	Hyperparameters are essential for optimizing the model.
These are specified or estimated while training the model.	These are set before the beginning of the training of the model.
It is internal to the model.	These are external to the model.
These are learned & set by the model by itself.	These are set manually by a machine learning engineer/practitioner.
These are dependent on the dataset, which is used for training.	These are independent of the dataset.
The values of parameters can be estimated by the optimization algorithms, such as Gradient Descent.	The values of hyperparameters can be estimated by hyperparameter tuning.
The final parameters estimated after training decide the model performance on unseen data.	The selected or fine-tuned hyperparameters decide the quality of the model.
Some examples of model parameters are Weights in an ANN, Support vectors in SVM, Coefficients in Linear Regression or Logistic Regression.	Some examples of model hyperparameters are the learning rate for training a neural network, K in the KNN algorithm, etc.

Hyperparameter Tuning Algorithms

Grid Search

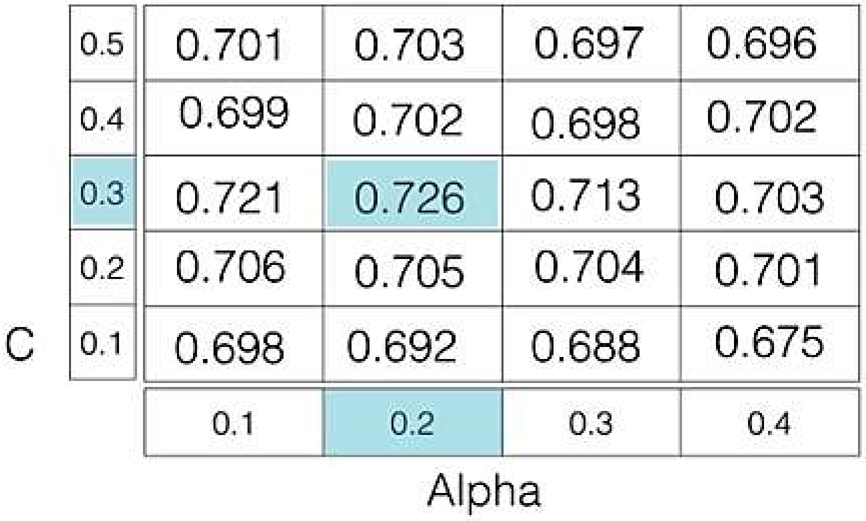
- simple, effective, yet resource expensive hyperparameter optimization technique that evaluates a grid of hyperparameters
- The method evaluates each hyperparameter and determines the winner
- For example, if the hyperparameter were the number of leaves in a decision tree, which could be anywhere from 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

Grid Search

- a “brute force” approach to hyperparameter optimization
- We fit the model using all possible combinations after creating a grid of potential discrete hyperparameter values
- We log each set’s model performance and then choose the combination that produces the best results
- This approach is called GridSearch, because it searches for the best set of hyperparameters from a grid of hyperparameters values
- An exhaustive approach that can identify the ideal hyperparameter combination is grid search
- But the slowness is a disadvantage
- It often takes a lot of processing power and time to fit the model with every potential combination, which might not be available

Example

- **For example:**
- if we want to set two hyperparameters C and Alpha of the Logistic Regression Classifier model, with different sets of values.
- The grid search technique will construct many versions of the model with all possible combinations of hyperparameters and will return the best one
- As in the image, for C = [0.1, 0.2, 0.3, 0.4, 0.5] and Alpha = [0.1, 0.2, 0.3, 0.4]
- For a combination of ***C=0.3 and Alpha=0.2***, the performance score comes out to be **0.726(Highest)**, therefore it is selected



C				
0.5	0.701	0.703	0.697	0.696
0.4	0.699	0.702	0.698	0.702
0.3	0.721	0.726	0.713	0.703
0.2	0.706	0.705	0.704	0.701
0.1	0.698	0.692	0.688	0.675
	0.1	0.2	0.3	0.4
	Alpha			

Drawback: GridSearchCV will go through all the intermediate combinations of hyperparameters which makes grid search computationally very expensive.

Hyperparameter Tuning Algorithms

- Random Search
- 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, Bergstra et al. demonstrated that in a surprising number of instances, a random search performed roughly as well as grid search.
- 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

Random search

- As the name suggests, the random search method selects values at random as opposed to the grid search method's use of a predetermined set of numbers
- Every iteration, random search attempts a different set of hyperparameters and logs the model's performance
- It returns the combination that provided the best outcome after several iterations
- This approach reduces unnecessary computation
- solves the drawbacks of GridSearch as it goes through only a fixed number of hyperparameter settings
- It moves within the grid in a random fashion to find the best set of hyperparameters
- The advantage is that, in most cases, a random search will produce a comparable result faster than a grid search

Advantages & Disadvantages

- **Advantages of Hyperparameter tuning:**
- Improved model performance
- Reduced overfitting and underfitting
- Enhanced model generalizability
- Optimized resource utilization
- Improved model interpretability
- **Disadvantages of Hyperparameter tuning:**
- Computational cost
- Time-consuming process
- Risk of overfitting
- No guarantee of optimal performance
- Requires expertise

Multivariate Testing

- Extremely useful method of determining which model is best for the particular problem at hand
- Also known as statistical hypothesis testing
- 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:
  - Split the population into randomized control and experimentation groups
  - Record the behavior of the populations on the proposed hypotheses
  - Compute the performance metrics and associated p-values
  - 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

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

How Long to Run a Multivariate Test?

- 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

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

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 environment (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

Ethics of Intelligence

- Novel solutions to problems developed through machine learning are themselves leading to questions of morality and ethics
- Currently, governance is moving at the pace of the industry itself
- There are many scenarios within AI for which there are no precedents, regulations, or laws
- As a result, it is paramount to consider the ethical and moral implications of creating intelligent systems

Ethics of Intelligence

- As AI penetrates humanity’s day to-day activities, philosophical, moral, ethical, and legal questions are raised.
- What Is Ethics?
- Ethics or moral philosophy refers to the moral codes of conduct (or set of moral principles) that shape the decisions people make and their conduct
- Morality refers to the principles that distinguish between good/right or bad/ wrong behavior
- Ethics in the workplace, for example, is often conveyed through professional codes of conduct for which employees must abide

What Is Data Science Ethics?

- Data science ethics is a branch of ethics that is concerned with privacy, decision-making, and data sharing
- Data science ethics comprises three main strands:
  - Ethics of data: focuses on the generation, collection, use, ownership, security, and transfer of data
  - Ethics of intelligence : covers the output or outcomes from predictive analytics that data is used to develop
  - Ethics of practices: proposed by Floridi and Taddeo, referring to the morality of innovation and systems to guide emerging concerns

Data Ethics

- More smartphones exist in the world than people—and phones, tablets, and digital devices alongside apps, wearables, and sensors are creating millions of data points a day
- There are over 7.2 billion phones in use, 112 million wearable devices sold annually, and over 100,000 healthcare apps available to download on your mobile phone
- IBM reports more than 2.5 quintillion bytes (2.5 × 1018) of data are created daily
- Data is everywhere Moreover, it’s valuable
- The topic of data ethics has been thrust into the public spotlight through high profile fiascos such as the Facebook Cambridge Analytica scandal
- Facebook, one of the world’s largest and most trusted data collection organizations, had user data harvested through a quiz hosted on its platform
- Behavioral and demographic data of the 1.5 million completers of the quiz was sold to Cambridge Analytica
- The data is largely considered to have been used to target and influence the outcome of the United States 2017 elections
- What’s more concerning is that this breach of security was reported over 2 years after the initial data leak
- We are in a time where fake news can travel quicker than the truth
- Society is at a critical point in its evolution, where the use, acceptance, and reliance on data must be addressed collectively to develop conversations and guiding principles on how to handle data ethically

Informed Consent

- Informed consent refers to the user (or patient) being aware of what their data will be used for
- Informed consent refers to an individual being legally able to give consent
- Typically this requires an individual to be over 18, of sound mind, and able to exercise choice
- Consent should ideally be voluntary
- Scenario A demonstrates how useful data can be given a particular context (or use case) and demonstrates the many intricacies of informed consent.

Scenario A

- John, type 2 diabetic, aged 30, has a severe hypoglycemic episode and is rushed to the hospital
- John has been taken to the hospital unconscious for treatment
- John, a truck driver, was prescribed insulin to treat his type 2 diabetes by his doctor’s advice, which is most likely the cause of his hypoglycemia
- Lots of data is generated in the process—both in the hospital, by healthcare professionals, and also on John’s Apple Watch—his heart rate, heart rate variability, activity details, and blood oxygen saturation to look out for signs of a diabetic coma
- John’s blood sample is also taken, and his genome identified
- Let’s suppose all this data is used, and it’s useful. John’s Apple Watch was used to monitor his heart on the way to the emergency room, through which it was suspected that he has an irregular heartbeat, later confirmed with the hospital’s medical equipment. Upon waking from his hypoglycemic episode, John is pleased to learn that genetic testing does not always give bad news, as his risks of contracting prostate cancer is reduced because he carries low-risk variants of the several genes known in 2018 to contribute to these illnesses. However, John is told of his increased risks of developing Alzheimer’s disease, colon cancer, and stroke from his genetic analysis.

Freedom of Choice

- Freedom of choice refers to the autonomy to decide whether your data is shared and with whom.
- This refers to the active decision to share your data with any third party

Should a Person’s Data Consent Ever Be Overturned?

- In an ideal world, an individual’s decision to share their data should be respected
- However, as an absolute concept, this is neither realistic nor plausible
- For instance, let’s assume John was to decline consent to use his data. On the assumption that the emergency response team’s duty is to safeguard the health of its patients, John’s data helped to monitor his vital signs, which contributed to his survival.
- Arguably the emergency response team would be acting unethically if they were not to use the full variety of data available to them at that given moment. Perhaps even John himself would overrule his consent if it meant increasing his chances of survival.

Who Owns the Data?

- People generate thousands of data points daily
- Nearly every transaction and behavior creates a data trail left through devices such as smartphones, televisions, smartwatches, mobile apps, health devices, contactless cards, cars, and even fridges. However, who owns the data?
- The topic of data ownership is an example of a complex first-time problem that has facilitated the development of international policy and governance.
- Historically, user data is owned by companies rather than the individual

How Will Data Affect the Future?

- Prioritizing Treatments
- Determining New Treatments and Management Pathways
- More real-world evidence
- Enhancements in Pharmacology
- Optimizing Pathways Through Connectivity—Is There a Limit?
- Security

Ethics of Artificial Intelligence and Machine Learning

- Machine Bias
- Data Bias
- Human Bias
- Intelligence Bias
- Bias Correction
- Is Bias a Bad Thing?

Prediction Ethics

- Explaining Predictions
- Protecting Against Mistakes
- Validity
- Preventing Algorithms from Becoming Immoral
- Unintended Consequences
- How Does Humanity Stay in Control of a Complex and Intelligent System?

How Do Machines Affect Our Behavior and Interaction

- Humanity
- Behavior and Addictions
- Economy and Employment
- Affecting the future
- Playing God
- Overhype and Scaremongering
- Stakeholder Buy-In and Alignment
- Policy, Law, and Regulation
- Data and Information Governance
- Is There Such a Thing as Too Much Policy?