

Introduction to Machine Learning

Shadi Khalifa, PhD

Senior Analytics Developer
Centre for Advanced Computing
Queen's University
Khalifa.s@queensu.ca



Agenda



- Introduction to Machine Learning Theory
- Personal Income Prediction Hands-on

Tools required for the tutorial:

- Laptop/Tablet (with Internet Access)
- Access to <https://jupyter.org/try>



What is the difference between **AI** and **Machine Learning**?

- AI is the top-level container for all components and layers related to creating a system that represents a synthetic “mind” that can solve problems in a heuristic manner.
- Machine Learning is a subfield of AI that gives machines the ability to learn and act without being explicitly programmed.
- Machine learning is **the core** of any AI system.

Machine Learning is the systematic computational analysis of data.

Aka “Data Mining”, “Data Analytics” and “Data Science”

Machine Learning is the multidisciplinary art and science of estimating functions (models) based on datasets. These functions predict the likelihood of future events.

- e.g , $Y = b_0 + b_1 * X$
- The training process is done using the historical data to learn the model parameters, b_0 and b_1 .
- Then when we have new input data X , the trained model can predict the value of Y .

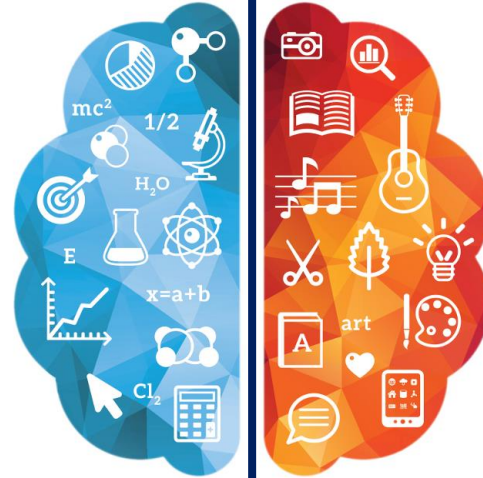
When to use Machine Learning?

Deduction is used in data-poor sciences, such as physics or mathematics, when reasoning in the presence of certainty.

All men are mortal.

Socrates was a man,

Therefore Socrates was mortal.

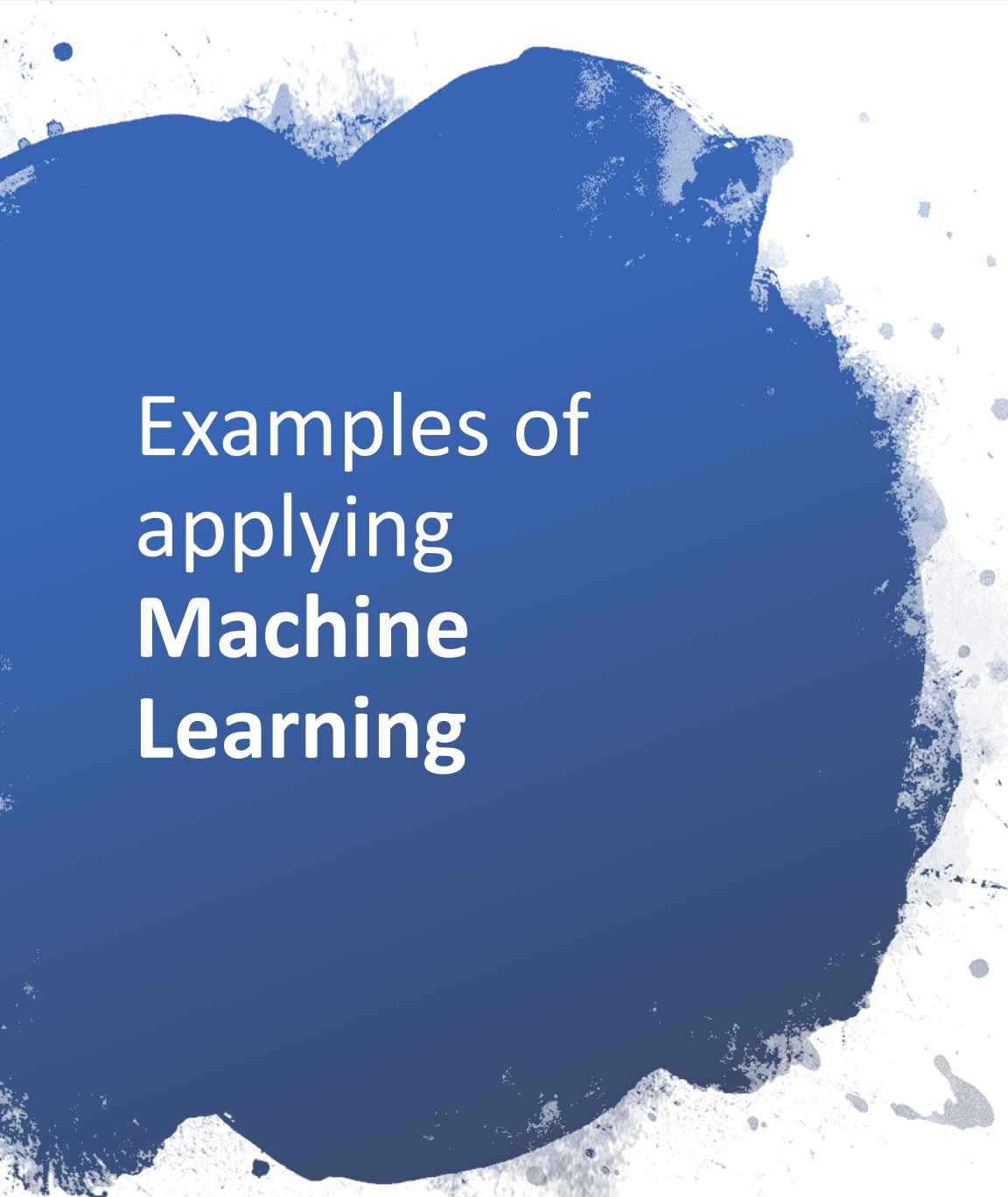


ML (Induction) is used in data-rich highly-dimensional problems, when reasoning in the presence of uncertainty.

Socrates was Greek.

Most Greeks eat fish.

Therefore Socrates ate fish
with **some confidence**.



Examples of applying Machine Learning

- Human expertise is absent (e.g. Navigating on Mars)
- Humans are unable to explain their expertise (e.g. Speech recognition, vision, language)
- Solution changes with time (e.g. Tracking, temperature control, preferences)
- Solution needs to be adapted to particular cases (e.g. Biometrics, personalization)
- The problem size is too vast for our limited reasoning capabilities (e.g. Calculating webpage ranks, matching ads to Facebook pages)

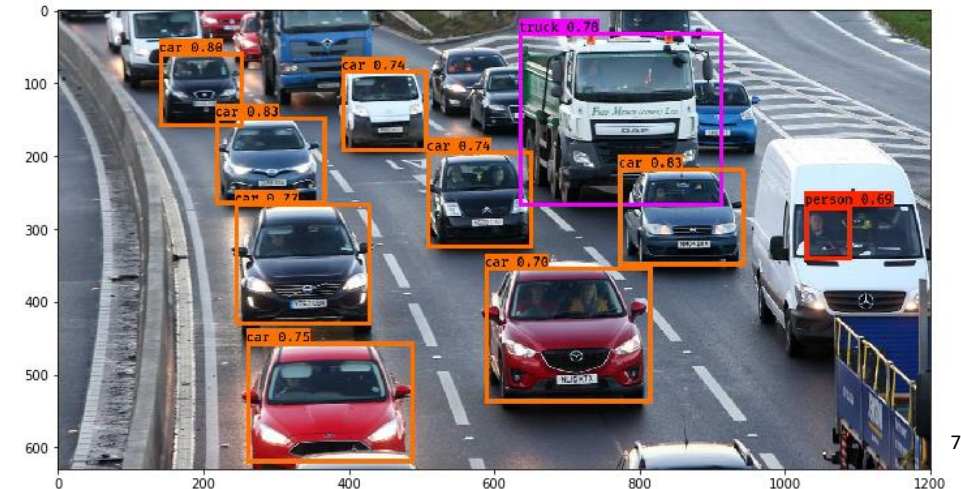
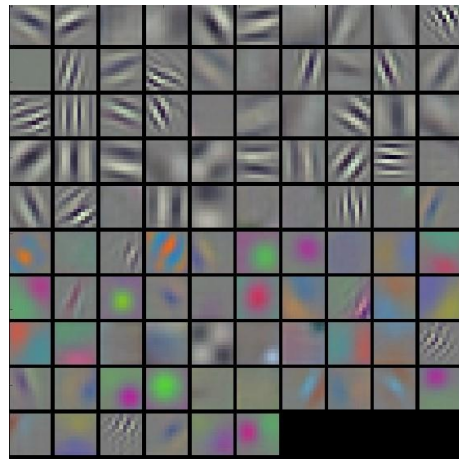
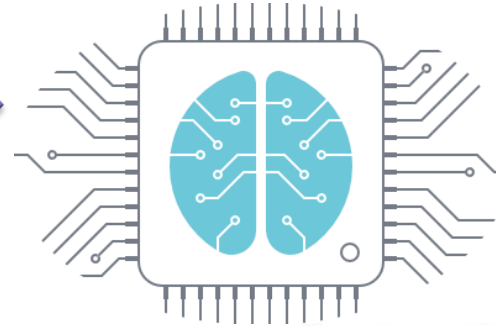
How Machine Learning works?

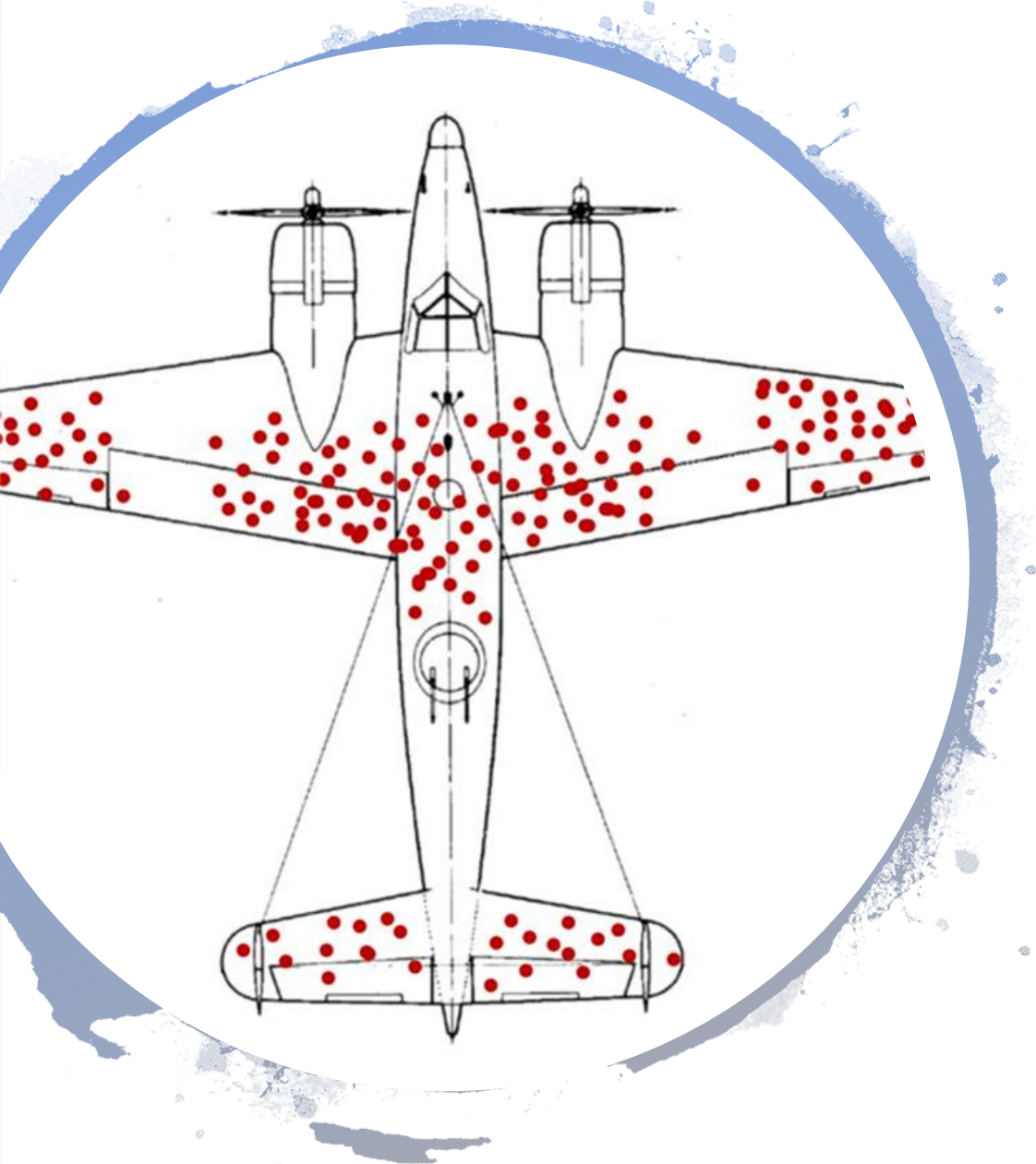
Observations

Model
Generation

Pattern
Extraction

Confidence
Calculation





Machine Learning Limitations

Since AI systems depend on data, then humans have two advantages:

- Humans have world knowledge, hence awareness of self in relation to the environment.
- Humans are better at decision making with little data or in rare circumstances.

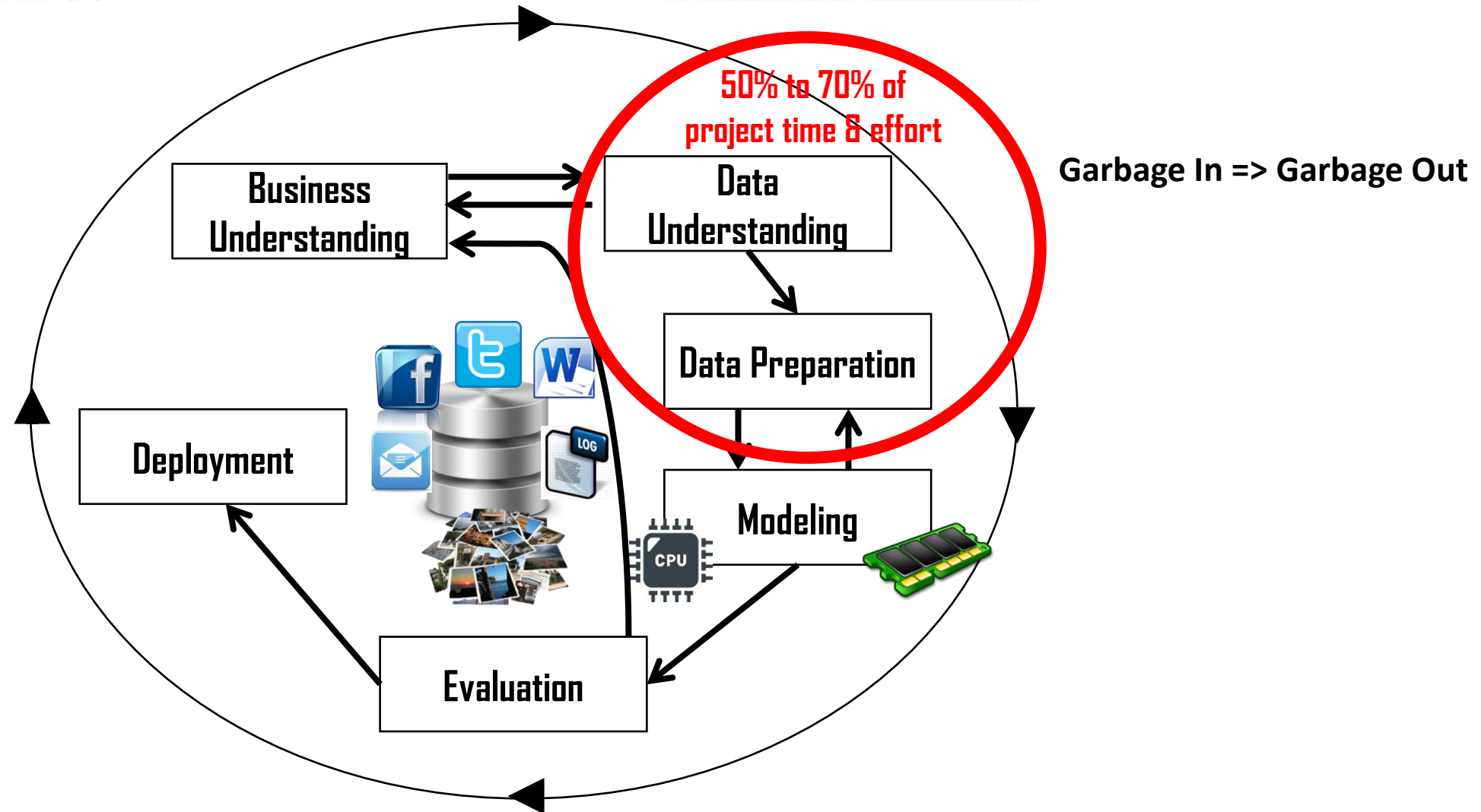
Humans have three types of data that machines don't:

- Data from our senses, smell, taste, touch, hearing and intuition.
- We are the ultimate arbitrators of our own preferences.
- Privacy concerns restrict the data available to machines.

ML Considerations

Human biases may propagate to AI systems via **training data** or **algorithmic models**

The Machine Learning Process



Cross Industry Standard Process for Data Mining (CRISP-DM)

Business Understanding

- **Acquire domain knowledge:** Learn the domain terminologies and workflow.
- **Identify the questions** that you need to answer.
- **Identify the Analytics type:**
 - **Descriptive (Insight into the past):** “What has happened?”
 - **Predictive (Understanding the future):** “What could happen?”
 - **Prescriptive (Advise on possible outcomes):** “How do future actions affect us?”



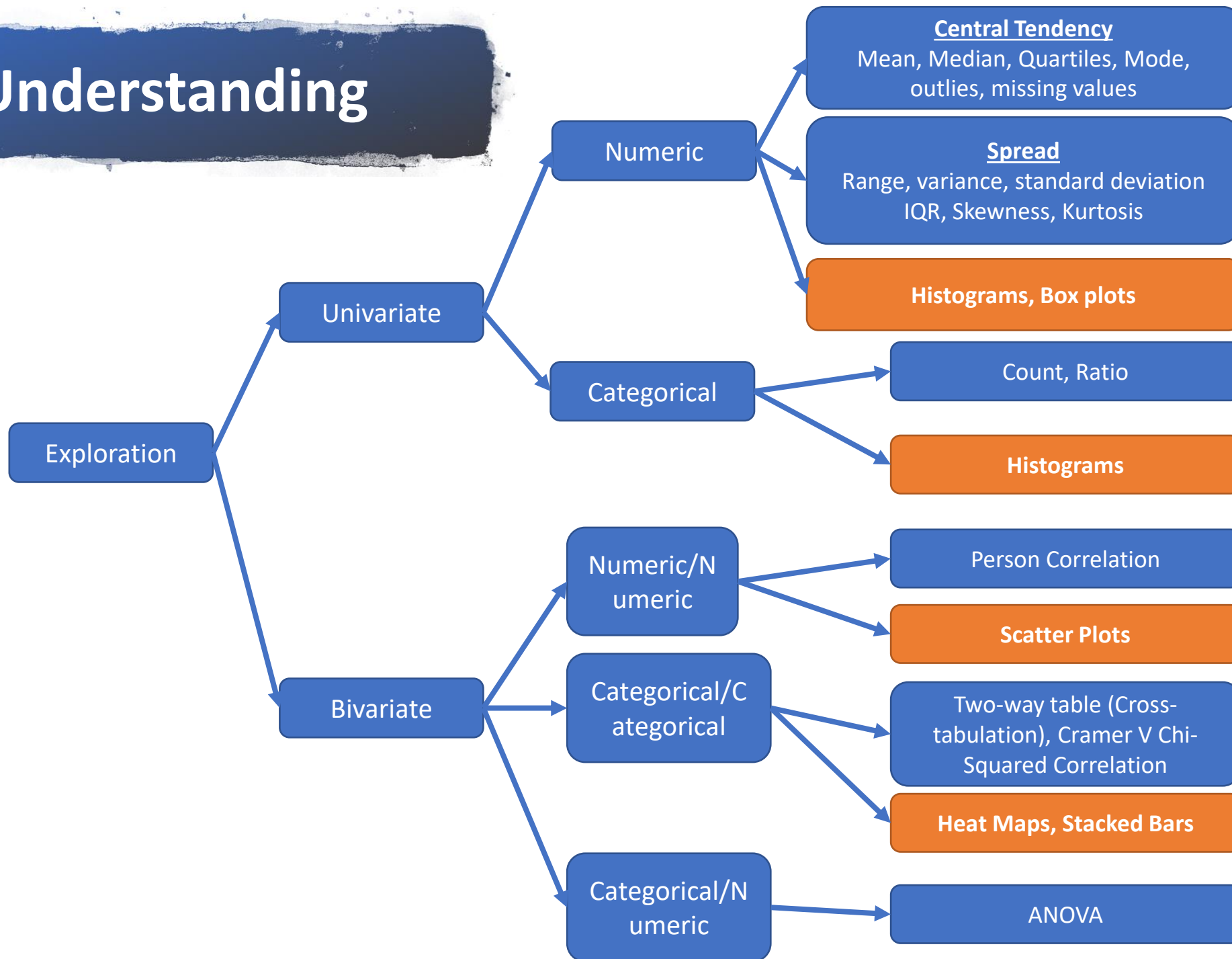
What do we mean by Data in this session?



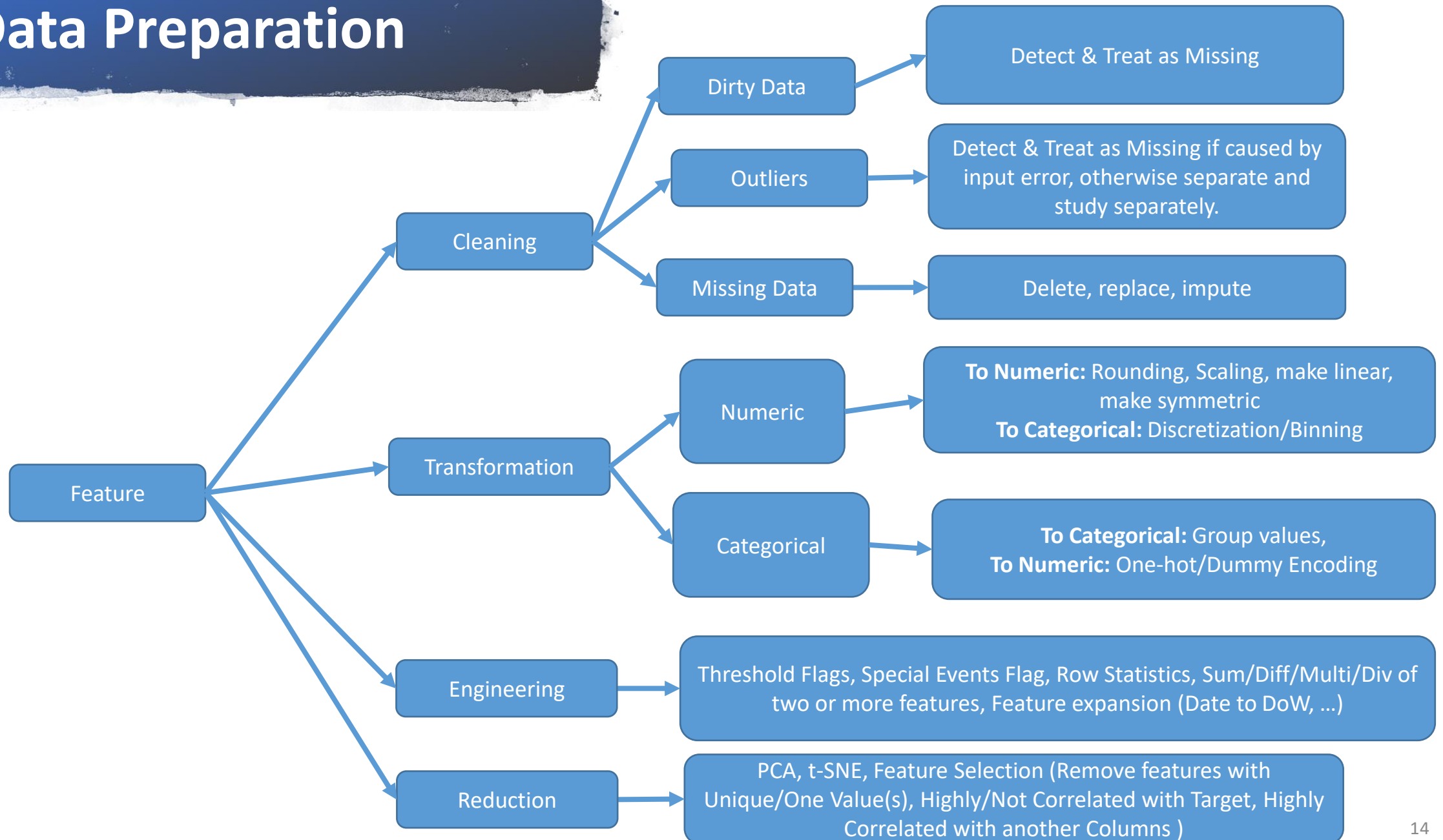
- To run analytics, Data needs to be represented or abstracted as an $n \times d$ *data matrix*, with n rows and d columns.
- Rows correspond to entities/records in the dataset.
 - The number of instances n is referred to as the size of the data.
- Columns represent attributes or properties of interest.
 - The number of attributes d is called the **dimensionality** of the data.
- Analytics run row-by-row considering all the row's columns.

$$\mathbf{D} = \begin{pmatrix} & X_1 & X_2 & \cdots & X_d \\ \mathbf{x}_1 & x_{11} & x_{12} & \cdots & x_{1d} \\ \mathbf{x}_2 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_n & x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

Data Understanding



Data Preparation



Modelling

- Machine Learning Models are **computational programs** to “**learn**” **directly from data** without relying on a predetermined equation(s).
- The **Models adaptively improve** their performance as **the number of samples available for learning increases**.

Learning can be

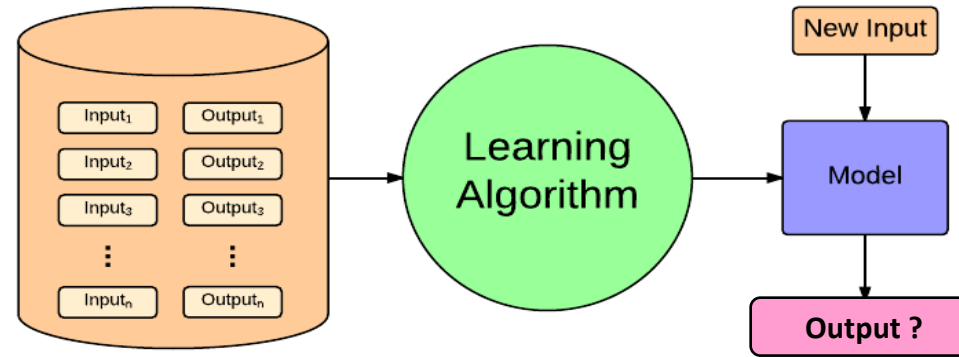
Unsupervised Learning

A model searches for **hidden groupings in the input data** when output values are not available.

Supervised Learning

A model is trained on **known input** and **output** data so that it can **predict future output values** for unseen input values.

Supervised Learning



- **Classification techniques** to predict **discrete** target values.
 - **Binary:** For example, whether a customer will respond to a marketing campaign or not.
 - **Multiclass:** For example, classify App users' reviews to complaining from "UI Design", "Bugs", "Ads" or "Battery Usage".
- **Regression techniques** to predict **continuous** target values.
 - For example, predicting how many units will sell in the next year.

Predicting grades using Machine Learning

What is the problem type?

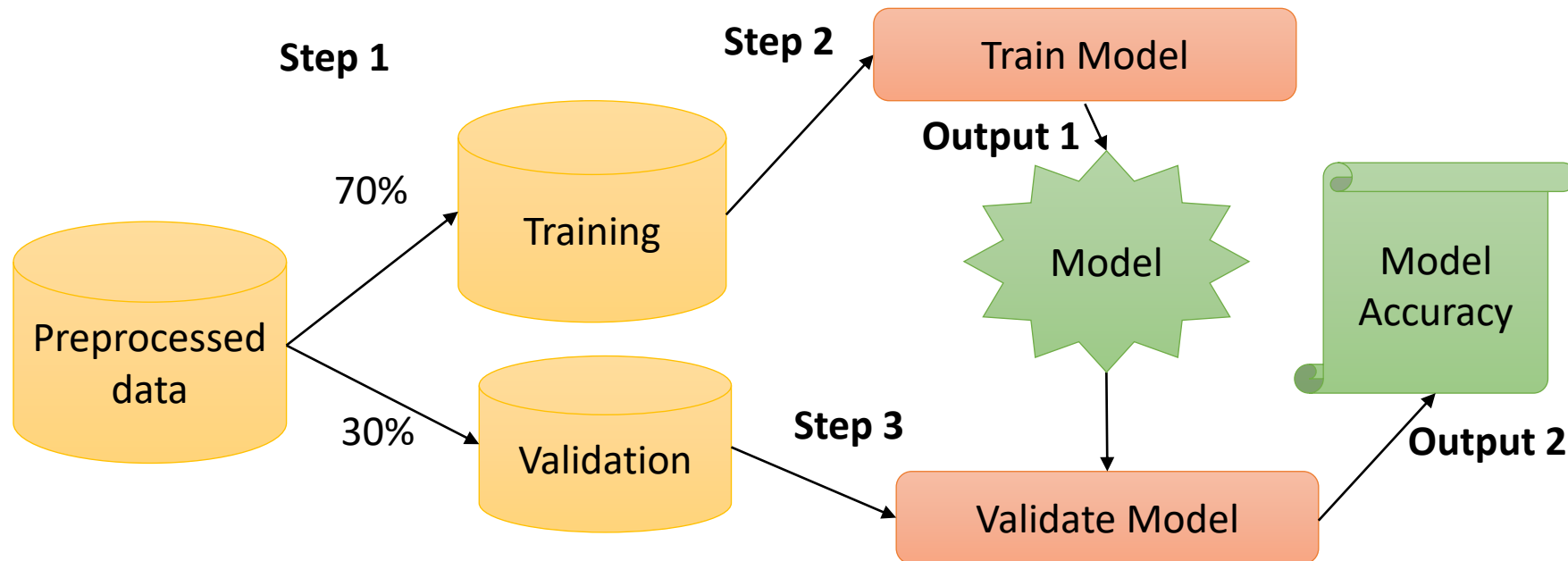
X		Y
hrs study	hrs sleep	Grade
7	5	78
6	8	93
8	2	67
5	5	? \hat{Y}

Supervised learning

This is a regression problem

Training & Validation Sets

- Split your dataset into **TWO representative** sub-sets (have records belonging to all Classes) :
 - Training**: used to **train/teach your Model** (usually 70% of your records)
 - Validation**: used to **calculate the accuracy** of your **trained Model** (usually 30% of your records)

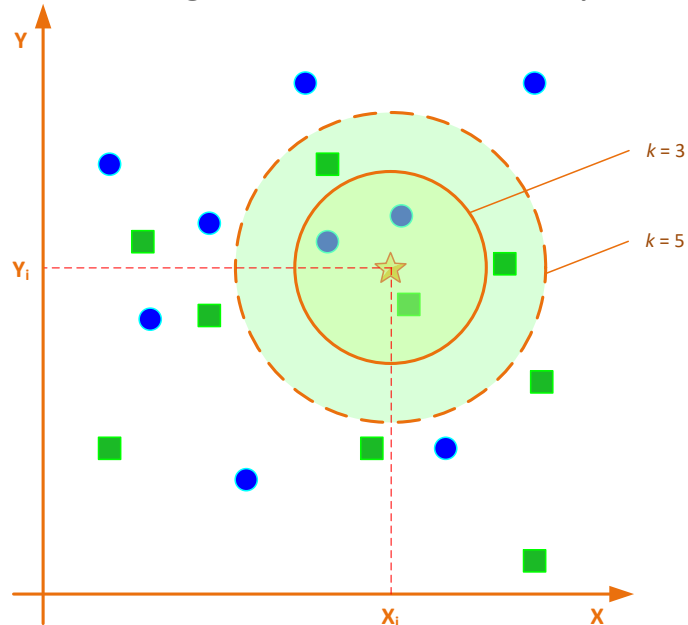


K-Cross Validation

- You can split your data to 70% training and 30% validation. But how can you be sure that the accuracy you are getting is not because of luck due to this specific training/validation split?
- Use **k-fold Cross-Validation**:
 - i. Split your data to k-folds of equal size (same number of records).
 - ii. Use $k-1$ folds to train your classifier and the remaining one for validation and calculating the model accuracy θ_i .
 - iii. Repeat step ii k -times leaving out a different fold each time.
 - iv. Calculate the mean of the model accuracy $\hat{\mu}_\theta = \frac{1}{k} \sum_{i=1}^k \theta_i$
 - v. Calculate the variance of the model accuracy $\hat{\sigma}_\theta^2 = \frac{1}{k} \sum_{i=1}^k (\theta_i - \hat{\mu}_\theta)^2$
- Note that the k-folds can be created using bootstrap resampling (random sampling with replacement) so that a record can exist in zero or more folds.
- Usually K is set to 10 (The 10-fold cross validation).

K Nearest Neighbor (k-NN)

- k-NN is a type of **instance-based learning** (or lazy learning)
 - Most of the work takes place at the time of scoring (not at modeling)
 - Data are kept after modeling (not only the model).
- **No actual training happens.** The model just saves all training data.
- **Scoring** works by calculating the distance (usually Euclidian distance) between the new record and every training data point.
 - The new record is predicted to be similar to those training data points of **shortest distance**.
- k is the number of neighbors used to classify the new record.



- Fast Training
- Consumes lots of memory and disk space to save all training data
- Slow scoring since it calculates the distance with every training data point

Iris Data

- One of the most famous datasets used in Analytics.
- The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.



Predict the type for: 4.5, 3.2, 1, 0.1 ?

Answer: Iris-setosa (most similar values for X_1, X_2, X_3, X_4)

	Sepal length X_1	Sepal width X_2	Petal length X_3	Petal width X_4	Class X_5
x_1	5.9	3.0	4.2	1.5	Iris-versicolor
x_2	6.9	3.1	4.9	1.5	Iris-versicolor
x_3	6.6	2.9	4.6	1.3	Iris-versicolor
x_4	4.6	3.2	1.4	0.2	Iris-setosa
x_5	6.0	2.2	4.0	1.0	Iris-versicolor
x_6	4.7	3.2	1.3	0.2	Iris-setosa
x_7	6.5	3.0	5.8	2.2	Iris-virginica
x_8	5.8	2.7	5.1	1.9	Iris-virginica
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
x_{149}	7.7	3.8	6.7	2.2	Iris-virginica
x_{150}	5.1	3.4	1.5	0.2	Iris-setosa

Bayesian Networks and Naïve Bayes

- Probability-based classifier
- Bayesian networks (aka belief networks) derive a predictive model from data based on Bayes theorem:

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

- For each of k possible outcomes or classes C_k

$$p(C_k | \mathbf{x}) = \frac{p(C_k) p(\mathbf{x} | C_k)}{p(\mathbf{x})}$$

- **Naïve Bayes** is a Bayesian network with the **hypothesis of independence between features**, which may require feature selection before building the model.
- This hypothesis allows us to do the following:

$$p(C_k | x_1 x_2 \dots x_n) = \frac{p(C_k) p(x_1 | C_k) p(x_2 | C_k) \dots p(x_n | C_k)}{p(x_1) p(x_2) \dots p(x_n)}$$

Practice

Record number	SYMPTOM	OCCUPATION	Diagnoses
1	Sneezing	Nurse	Flu
2	Sneezing	Nurse	Hayfever
3	Sneezing	Nurse	Flu
4	Vomiting	Nurse	Hayfever
5	Vomiting	Nurse	Flu
6	Vomiting	Farmer	Hayfever
7	Vomiting	Farmer	Flu
8	Vomiting	Farmer	Hayfever
9	Vomiting	Farmer	Hayfever
10	Sneezing	Nurse	Flu

Training

$$P(\text{Sneezing}) = 4/10$$

$$P(\text{Vomiting}) = 6/10$$

$$P(\text{Nurse}) = 6/10$$

$$P(\text{Farmer}) = 4/10$$

$$P(\text{Flu}) = 5/10$$

$$P(\text{Hayfever}) = 5/10$$

$$P(\text{Sneezing} | \text{Flu}) = 3/5$$

$$P(\text{Sneezing} | \text{Hayfever}) = 1/5$$

$$P(\text{Vomiting} | \text{Flu}) = 2/5$$

$$P(\text{Vomiting} | \text{Hayfever}) = 4/5$$

$$P(\text{Nurse} | \text{Flu}) = 4/5$$

$$P(\text{Nurse} | \text{Hayfever}) = 2/5$$

$$P(\text{Farmer} | \text{Flu}) = 1/5$$

$$P(\text{Farmer} | \text{Hayfever}) = 3/5$$

Scoring: Vomiting, Nurse?

$$P(\text{Flu} | \text{Vomiting, Nurse}) = P(\text{Flu}) P(\text{Vomiting} | \text{Flu}) P(\text{Nurse} | \text{Flu}) / P(\text{Vomiting}) P(\text{Nurse})$$

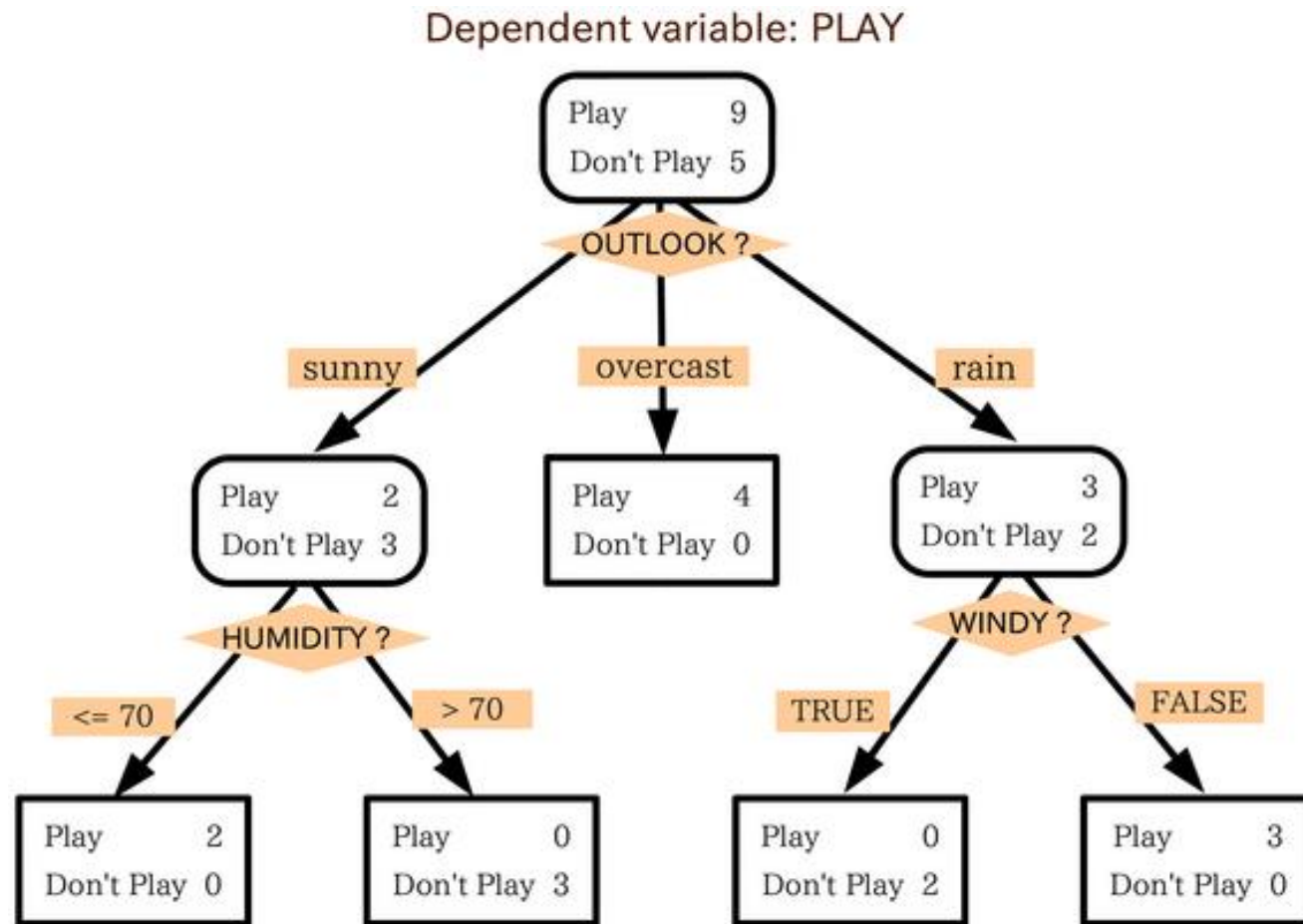
$$= 1/5 * 2/5 * 4/5 / 6/10 * 6/10 = 0.064 / 0.36 = 0.178$$

$$P(\text{Hayfever} | \text{Vomiting, Nurse}) = P(\text{Hayfever}) P(\text{Vomiting} | \text{Hayfever}) P(\text{Nurse} | \text{Hayfever}) / P(\text{Vomiting}) P(\text{Nurse}) =$$

$$5/10 * 4/5 * 2/5 / 6/10 * 6/10 = 0.16 / 0.36 = \mathbf{0.44}$$

Decision Tree and Random Forest

- Predict the class for this new data point (Outlook = Sunny and Humidity = 80)

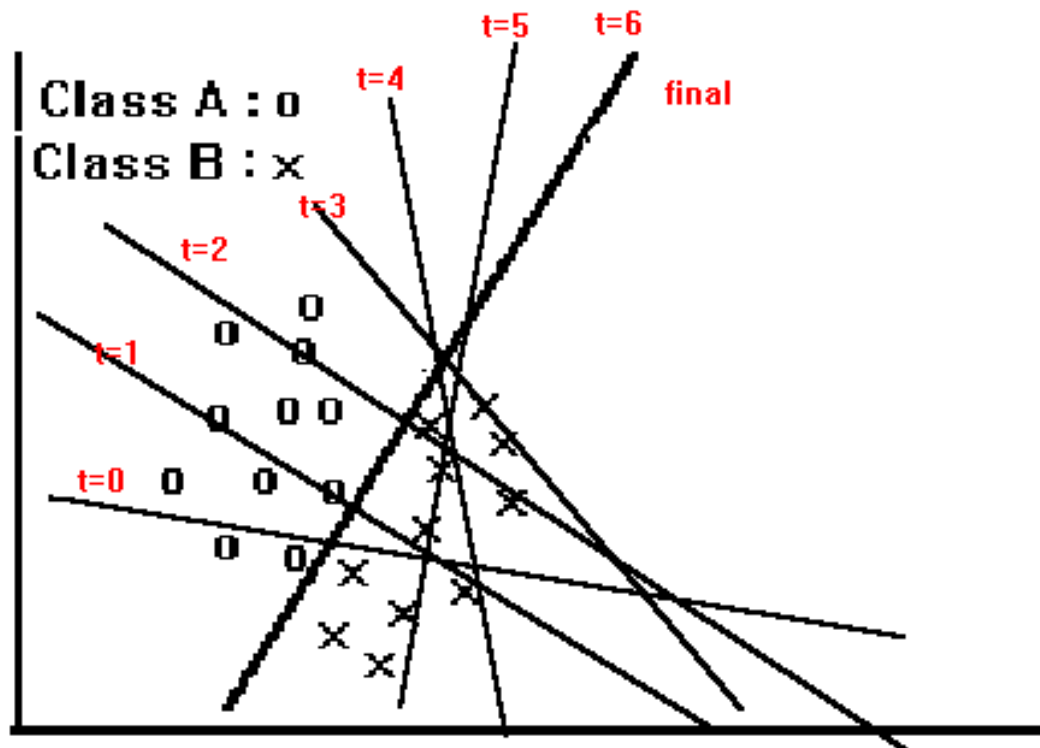


Pros: Easily understandable and can work with polynomial attributes.

Cons: Needs entire data to fit in memory.

Iterative Convergence

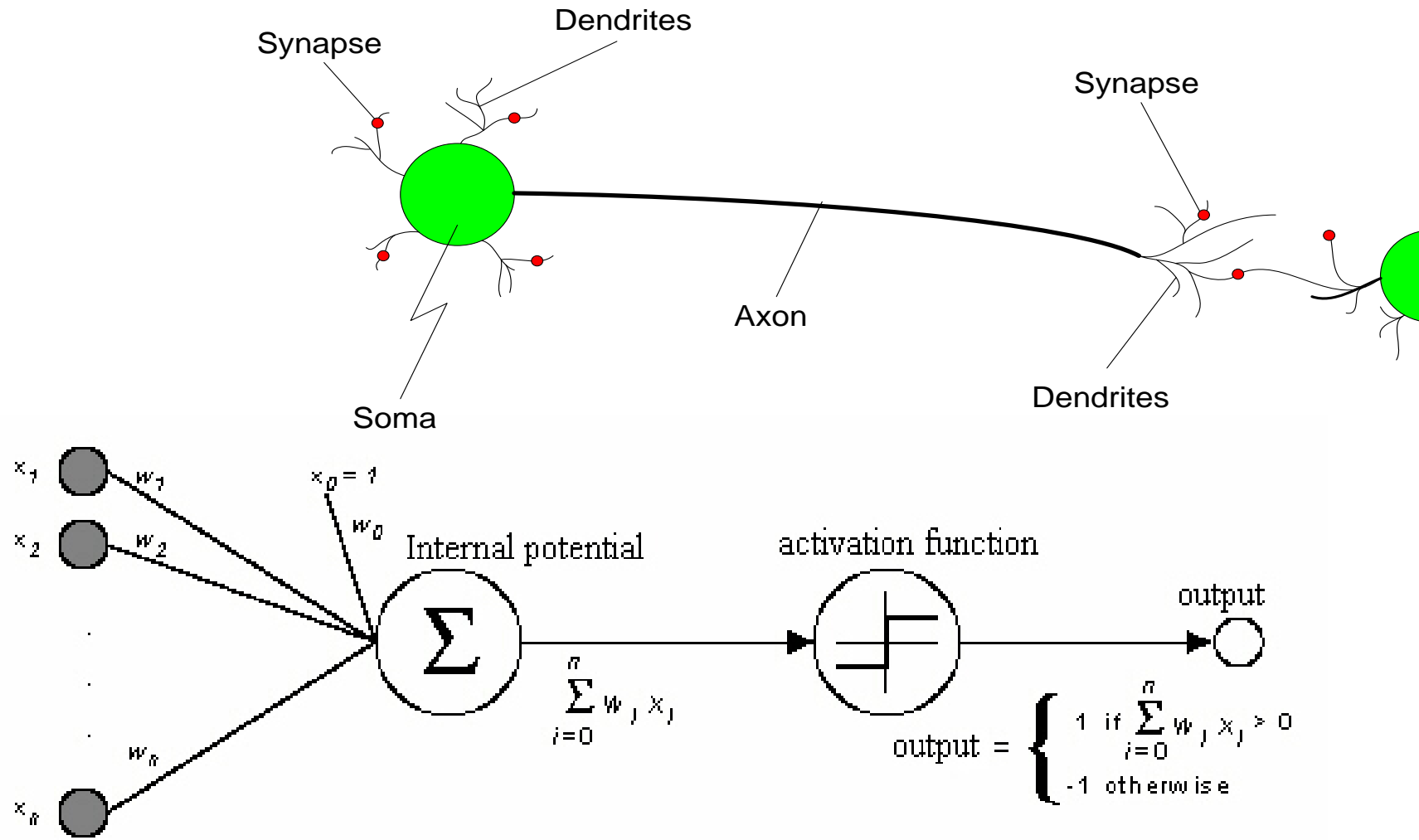
Linear Regression
Logistic Regression
Support Vector Machines (SVM)
Neural Networks



The Artificial Neural Network Learning Algorithms



A Single Artificial Neuron



Single Perceptron

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

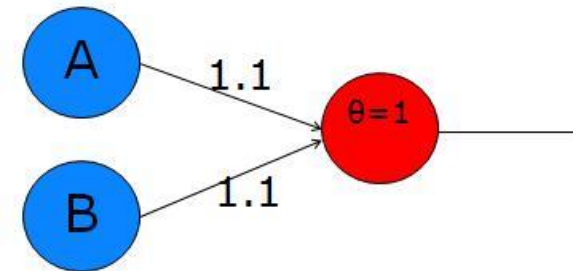
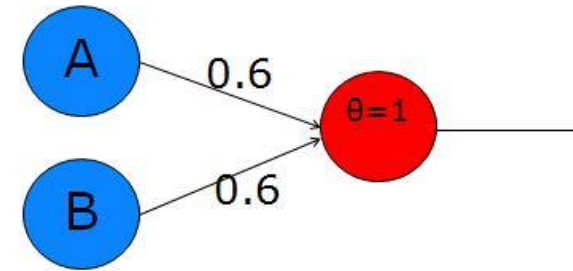
A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

A	B	$A \text{ xor } B$
0	0	0
0	1	1
1	0	1
1	1	0

Neuron Activation Threshold $\Theta = 1$

Output = 1 iff $\underline{W.X} > 1$

Otherwise output = 0



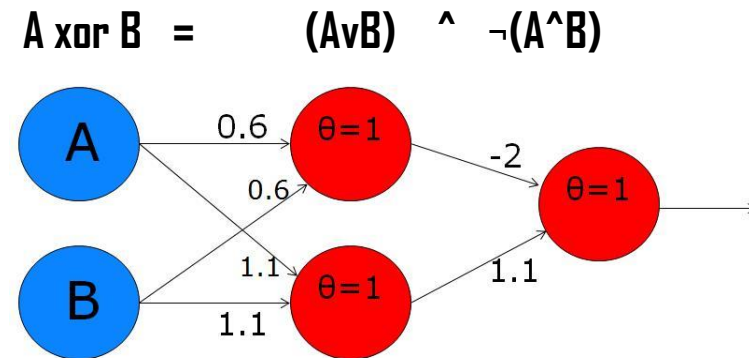
FAIL

$$A \text{ xor } B = (A \vee B) \wedge \neg(A \wedge B)$$

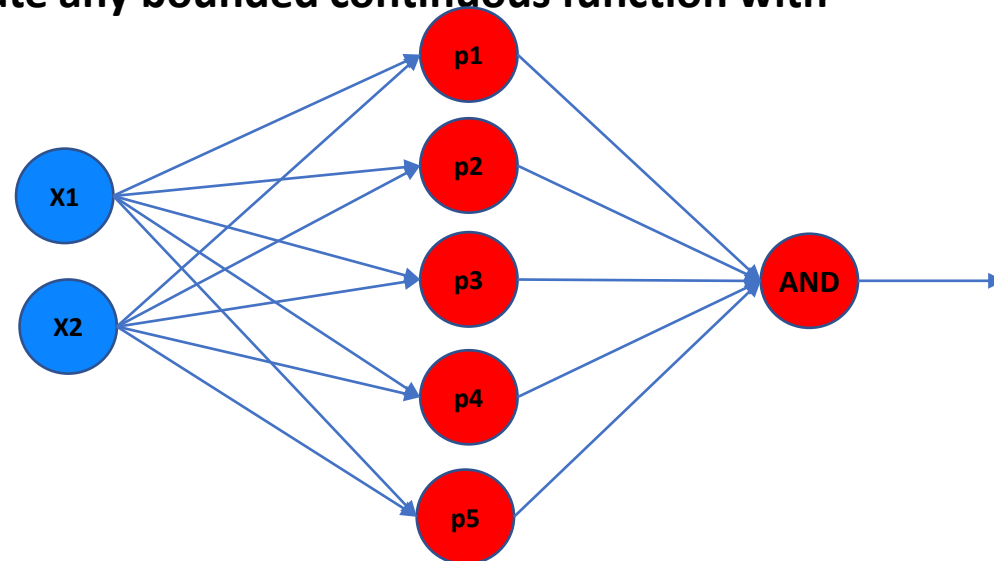
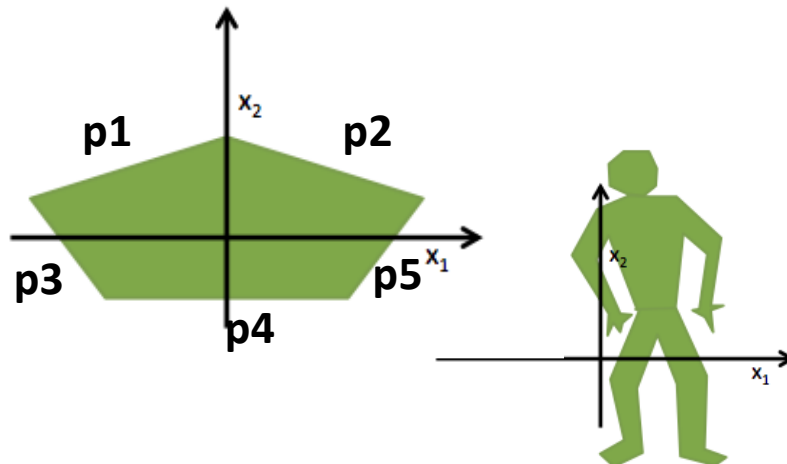
One-Layer Perceptron

- A One-Layer Perceptron of one hidden layer can represent any Boolean function exactly.

	A	B	AxorB
	0	0	0
	0	1	1
	1	0	1
	1	1	0



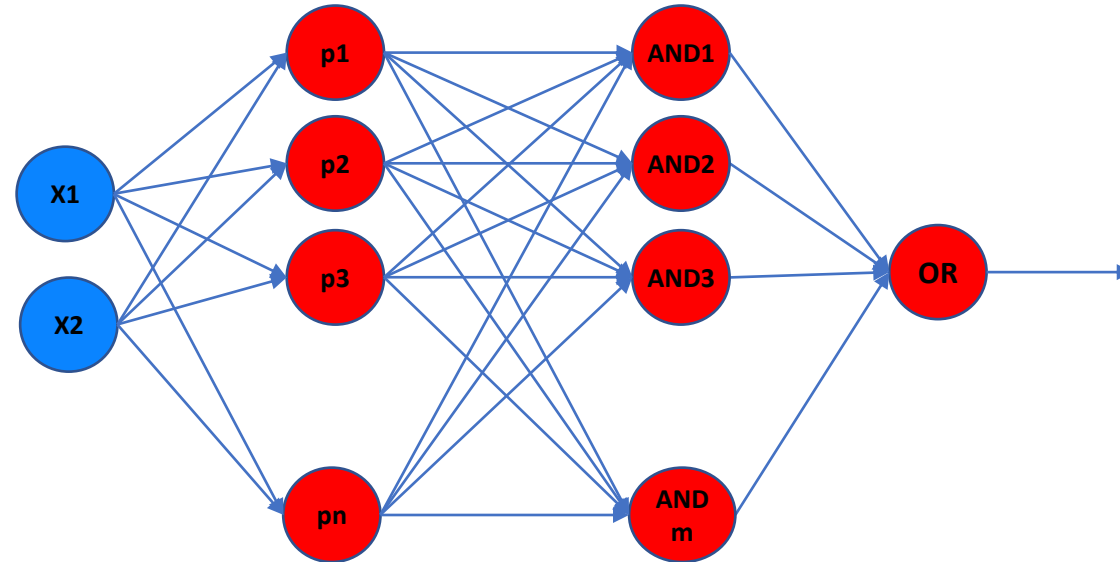
- A One-Layer Perceptron can approximate any bounded continuous function with arbitrary accuracy.



Having a large number of Perceptrons, a One-Layer Perceptron creates a circle region.

Multi-Layer Perceptron (MLP) / Neural Network

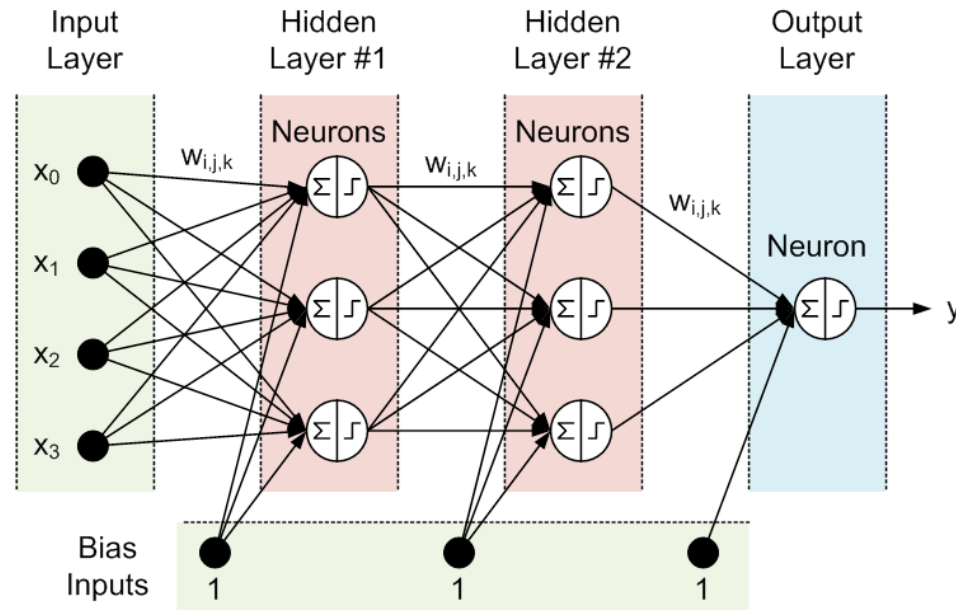
- A MLP of two hidden layers can approximate any function with arbitrary accuracy.



- A MLP with one hidden layer can represent a circle.
- A second hidden layer combines the outputs of many different circles.
- Each circle requires a large number of neurons, consequently, the entire function requires even more.
- Despite the exponentially many neurons needed, one will never need a third hidden layer to approximate any function.

Multi-Layer Perceptron (MLP)

- A MLP is a **feedforward** artificial neural network model that maps sets of input data onto a set of appropriate outputs.
- A MLP consists of multiple layers of nodes in a directed graph, with **each layer fully connected to the next one**.
- Except for the input nodes, each node is a neuron with a **Transfer/Activation function**.
- MLP utilizes a supervised learning technique called **backpropagation for training** the network (finding the synapses weights).
- MLP is just one type of ANN, other types are Recurrent, self-organizing feature maps, Hopfield networks, Deep Learning, ...

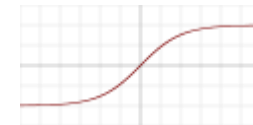
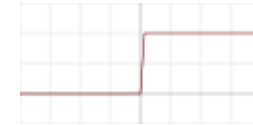
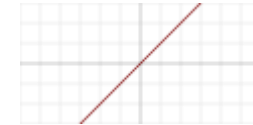


Number of Neurons per Layer

- **Input layer:** One Neuron per input (feature), these are not typical neurons but simply pass the data through to the next layer
- **Hidden layer(s):** You can have as many hidden layers as you want and each can consist of as many Neurons as you want.
- **Output layer:** Each output Neuron represents a classifier.
 - **Binary classification/Regression** - Single neuron with an activation function
 - **Multi-class classification** - Multiple neurons, one for each class (Target value), and a max function to output the final result which is the class belonging to the output Neuron of the highest value.
 - **Multi-class classification** – $(\log_2 m)$ neurons where m is the number of classes (Target values). Neurons work together to identify the winning class using binary representation. (Example: Using 2 Neurons for 4 classes: Class A \rightarrow 00, Class B \rightarrow 01 , Class C \rightarrow 10 , Class D \rightarrow 11)

Transfer/Activation Function

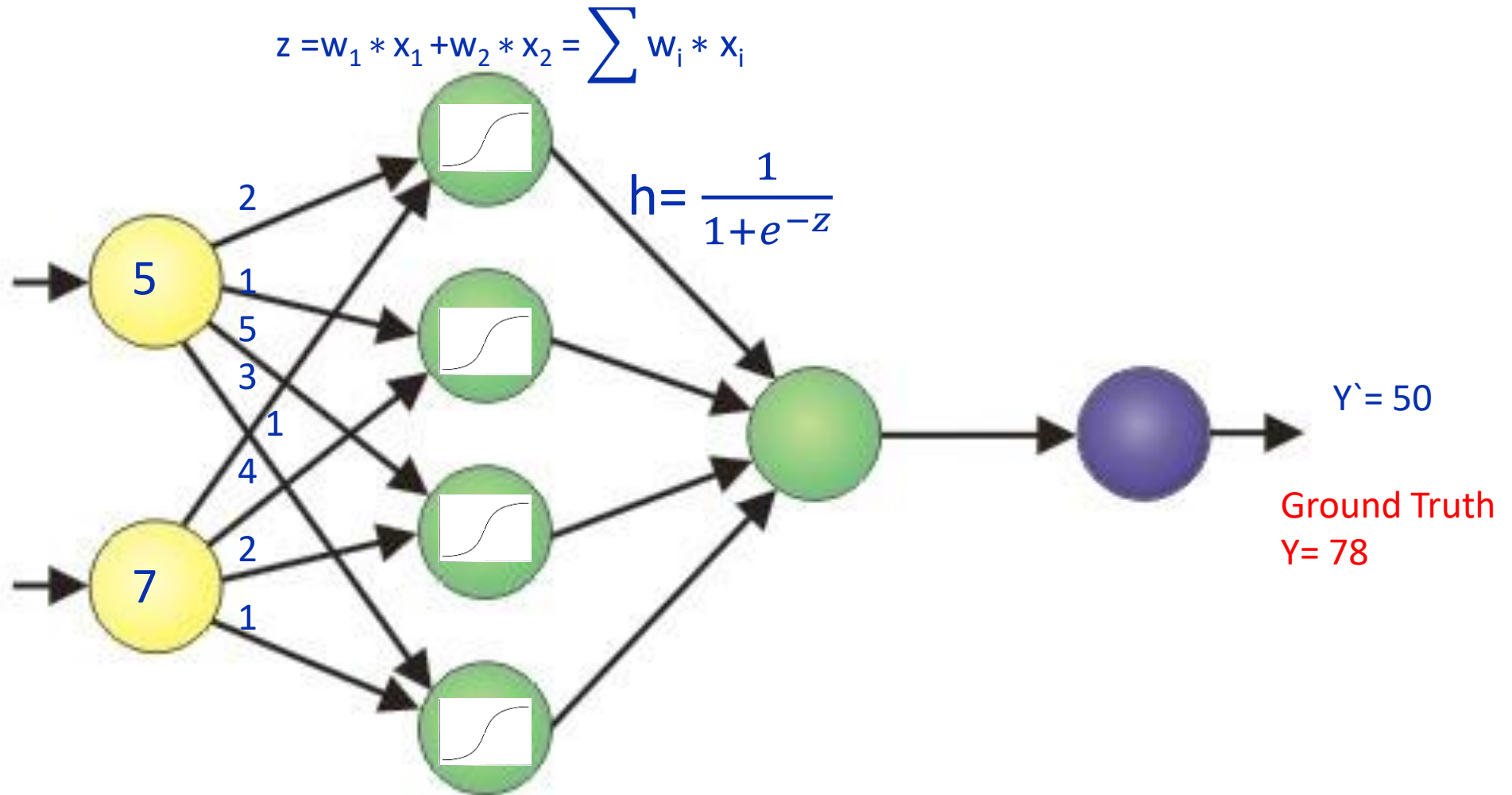
- It defines the relationship between the Neuron input and output. (i.e defines the range of the output)
- **Identity Function** $f(x) = x \rightarrow (-\infty, \infty)$
- **Binary Function** $f(x) = 1 \text{ if } x \geq 0 \text{ and } 0 \text{ otherwise} \rightarrow \{0,1\}$
- **Logistic (Sigmoidal) Function** $f(x) = 1/(1+e^{(-x)}) \rightarrow (0,1)$
- **Tanh Function** $f(x) = \tanh(x) \rightarrow (-1,1)$
- *And others*



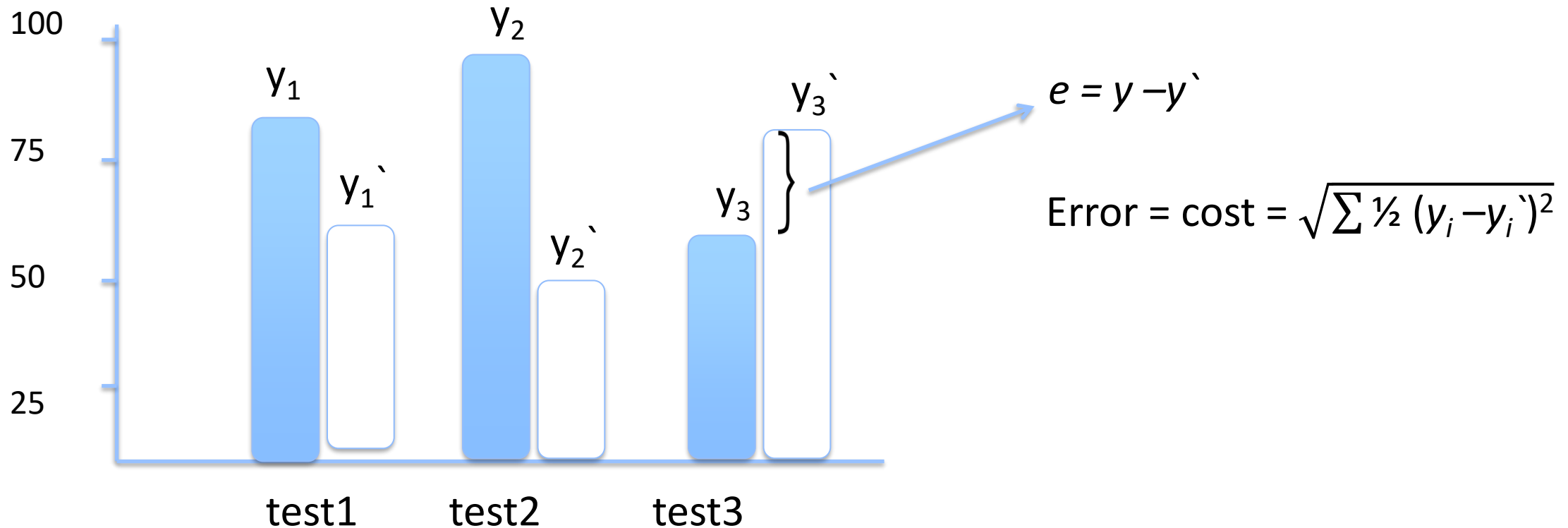
Predicting grades using Neural Network

X		Y
hrs study	hrs sleep	Grade
7	5	78
6	8	93
8	2	67
5	5	? \hat{Y}

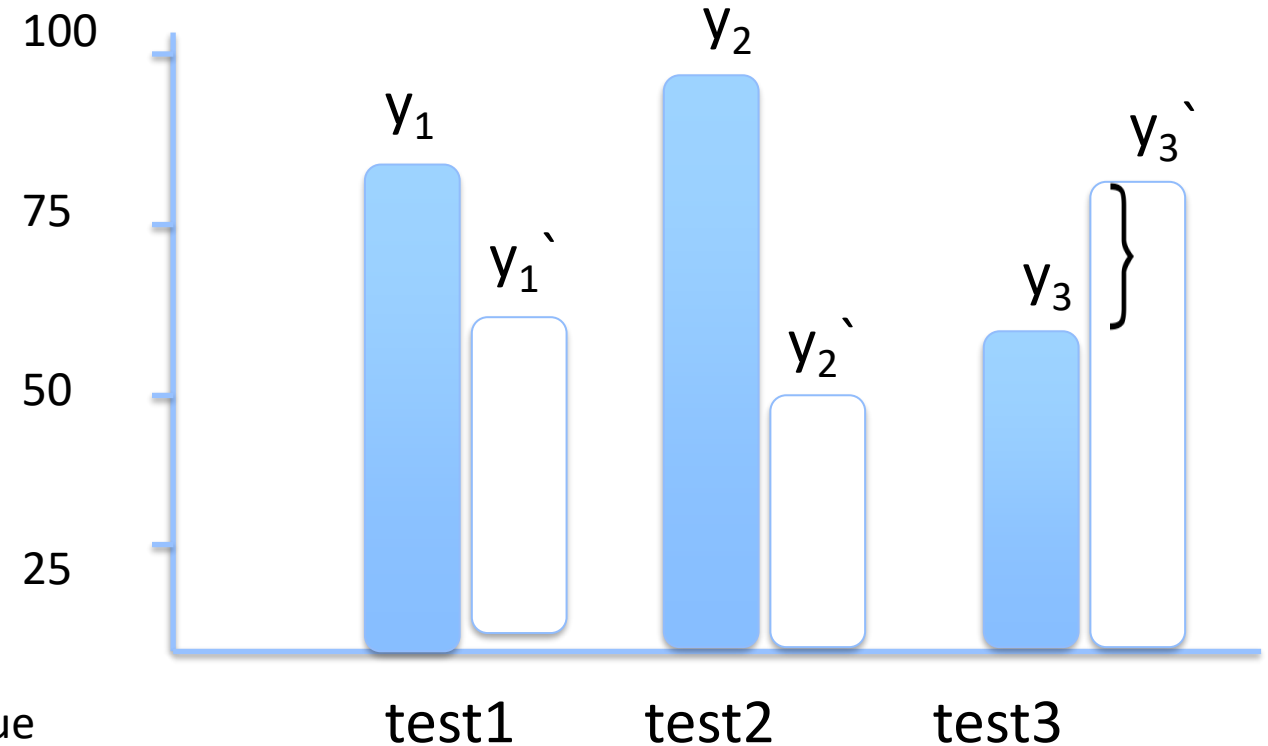
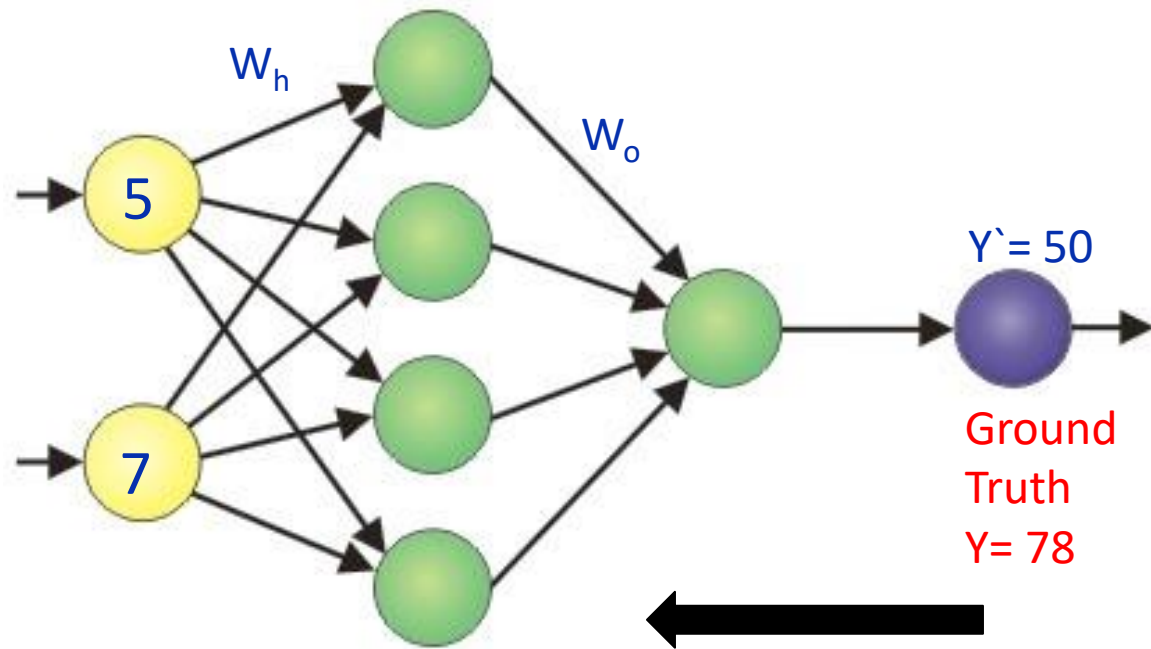
Forward Pass: Apply each input from the Training dataset and compute the actual output.



Compare actual output with desired output (Target)



Backpropagation: Update weights in proportion to the amount they affect the error.



To decrease the error, we subtract the error contribution value from the current weight (optionally multiplied by some constant learning rate α):

$$w_i^{(n+1)} = w_i^{(n)} - \alpha \frac{\partial \text{Error}}{\partial w_i}$$

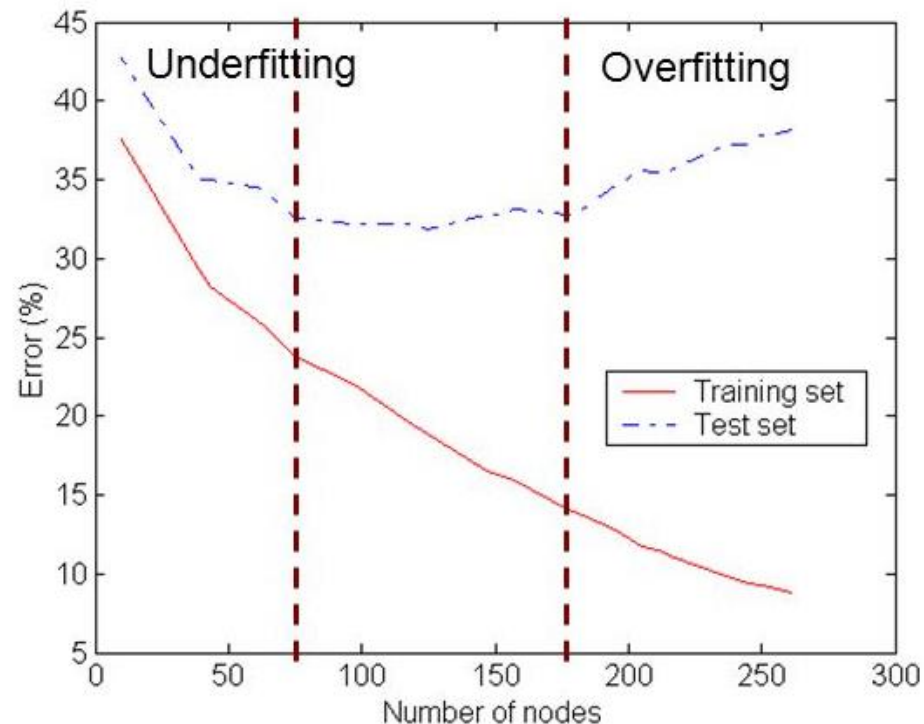
Repeat with the other input records

Model Evaluation



Underfitting and Overfitting

- **Underfitting** is when the model is too simple and both the training and scoring errors are large.
- **Overfitting** is when the model is too complex and it models the details of the training set and fails to generalize on the test set (small error on training set and large error on test set).



Confusion Matrix (Binary Classification)

- A **Confusion Matrix** is a table that compares the classifications made by the model with the actual class labels that we created.

		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive Count (TP)	Type II Error False Positive Count (FP)
	Negative	Type I Error False Negative Count (FN)	True Negative Count (TN)

Accuracy is the overall accuracy (how many the model got right).

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

Sensitivity (aka **Recall**) is the ability of the test to correctly identify the Positive class.

$$True\ Positive\ Rate = Sensitivity = \frac{TP}{TP + FN}$$

Specificity is the ability of the test to correctly identify the Negative class.

$$True\ Negative\ Rate = Specificity = \frac{TN}{TN + FP}$$

Precision measures the misclassifications.

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

Which **error type** and measure are the most important if we are predicting **CanPayLoanBack** and we can't tolerate predicting YES while it is NO?

F-Score

- The F-Score **combines the Precision and Recall** into a single metric.
 - The Precision is the number of correct positive results divided by the number of all positive predictions,
 - The Recall is the number of correct positive results divided by the number of true positives.
- The F-score can be interpreted as a **weighted average of the precision and recall**, where an F-score reaches its **best value at 1** and worst at 0.

$$FScore = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- Note that the F-measures **do not take the true negatives into account**.

Matthews Correlation Coefficient (MCC)

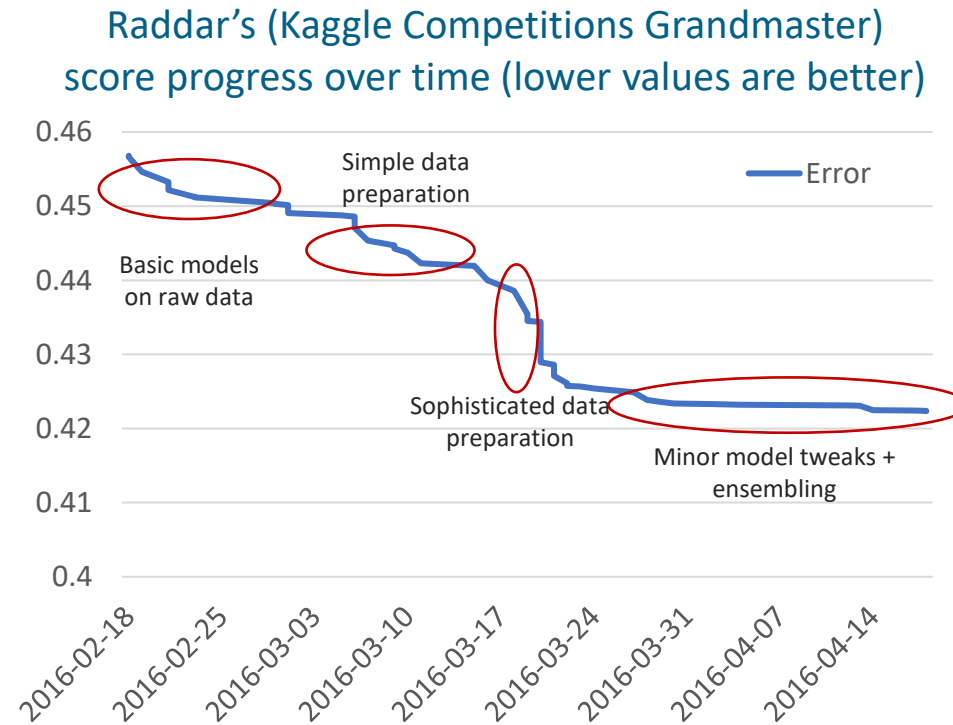
- The MCC is in essence a correlation coefficient between the true and predicted binary classifications.
- It returns a value between -1 and $+1$.
 - A coefficient of **+1 represents a perfect prediction**
 - A coefficient of 0 no better than random prediction
 - A coefficient of -1 indicates total disagreement between prediction and observation.
- The MCC **takes into account true and false positives and negatives** and so it is regarded as a balanced measure.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive Count (TP)	False Positive Count (FP)
	Negative	False Negative Count (FN)	True Negative Count (TN)

Rule of thumb: Data first, Modeling later

1. Create few simple models first to have a baseline.
2. Data exploration and visualization might be boring and frustrating but pays off well.
3. Create **smart** features.
4. Fine tune the model.



Back at ??:??

HANDS-ON

1. Go to <https://gke.mybinder.org/>
2. Use this link in the box saying Github:
[https://github.com/skhalifa/CAC ML S2020.git](https://github.com/skhalifa/CAC_ML_S2020.git)
3. Click Launch
4. Click MachineLearningHandsOn.ipynb



Shadi Khalifa, PhD

Senior Analytics Developer

Centre for Advanced Computing

Queen's University

Khalifa.s@queensu.ca