UNIVERSITY
OF TRENTO - Italy

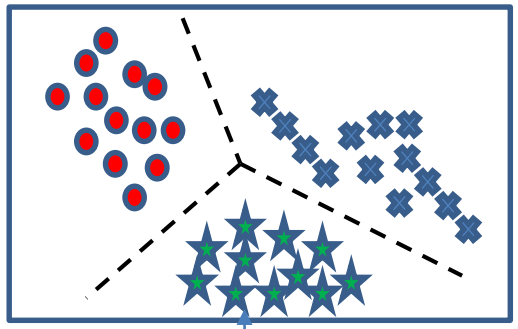
Classification using Machine Learning & Introduction to Classifiers

Sandeep Gupta

Advisor: Prof. Bruno Crispo

27-10-2019

Types to Machine Learning



Machine Learning

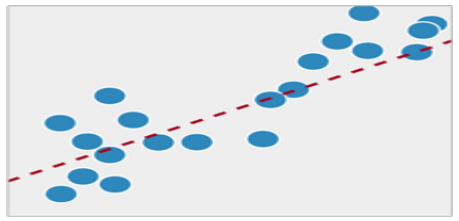
Supervised Learning

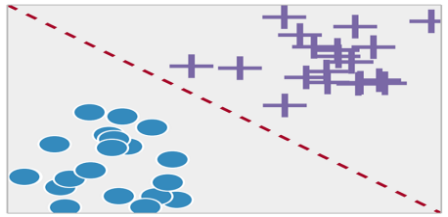
Unsupervised Learning

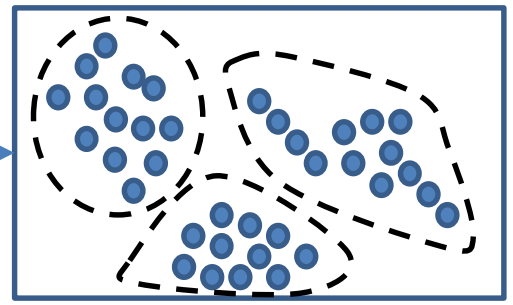
Reinforcement Learning

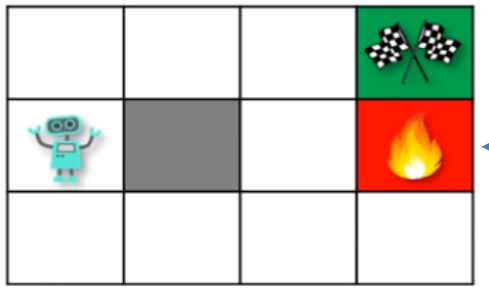
Regression

Classification









Classification

- Classification is a type of **supervised learning**.
- It is a systematic approach to build **classification models** from the labelled data set (training data) to predict class labels of new data.
- To build an efficient classification model:
  - **Data Pre-processing** (*transforming raw data into an understandable format*)
  - **Feature Extraction** (*Linear or nonlinear mappings*)
  - **Feature Fusion** (*Combining features of different modalities*)
  - **Feature Selection** (*Finding most productive feature-set*)
  - **Normalization of Data** (*Scaling*)
  - **Training and Testing Methods** (*to avoid overfitting or underfitting*)
- Classification models types:
  - Binary, Multiclass, One-class

Data Preprocessing

Data cleaning	*Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies*
Data integration	*Integration of multiple databases, data cubes, or files*
Data transformation	*Normalization and aggregation*
Data reduction	*Obtains reduced representation in volume but produces the same or similar analytical results*
Data discretization	*Part of data reduction but with particular importance, especially for numerical data*

Feature Extraction

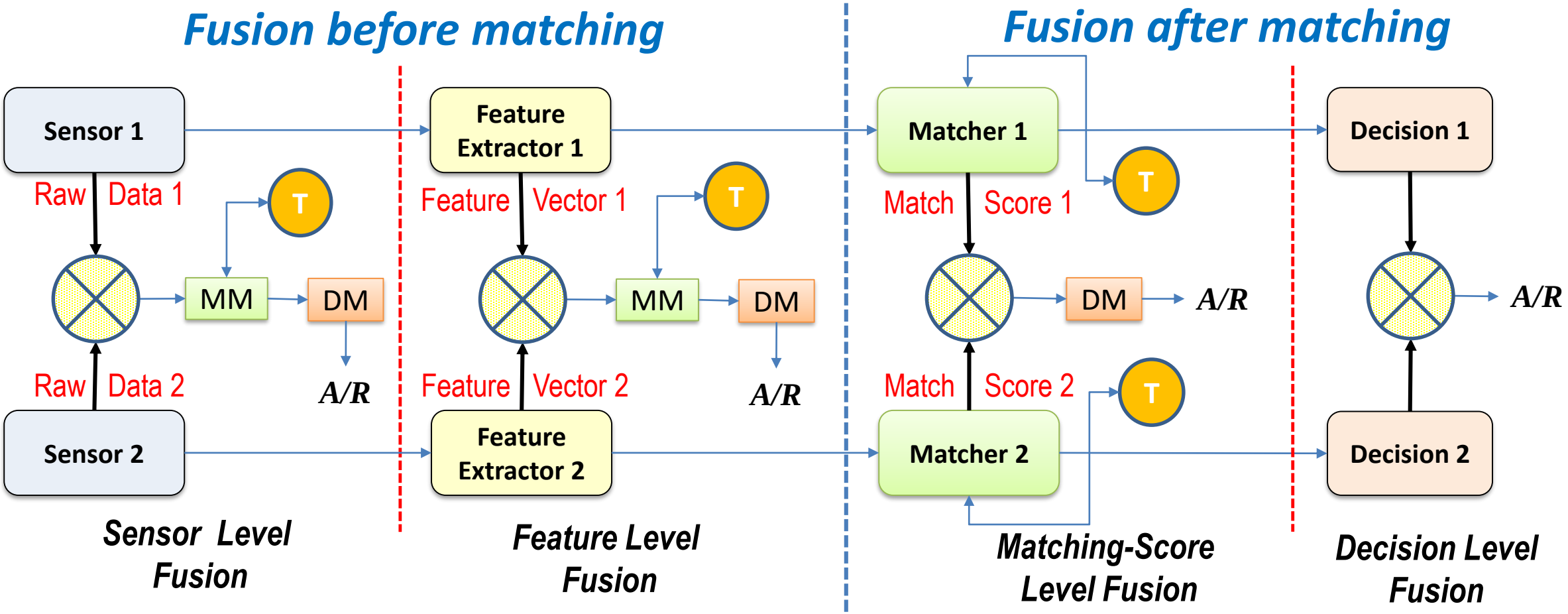
- Feature extraction is an attribute reduction process.
  - By mapping of the original *high-dimensional data (n)* onto a *lower-dimensional space (d)* $such\ that\ \boldsymbol{d \ll n}$

- Feature Reduction Algorithms

Linear	• Principal Component Analysis (PCA) • Linear Discriminant Analysis (LDA) • Latent Semantic Indexing (LSI) • Canonical Correlation Analysis (CCA) • Partial Least Squares (PLS)
Non-linear	• Nonlinear feature reduction using kernels • Manifold learning

2.5 Feature Fusion

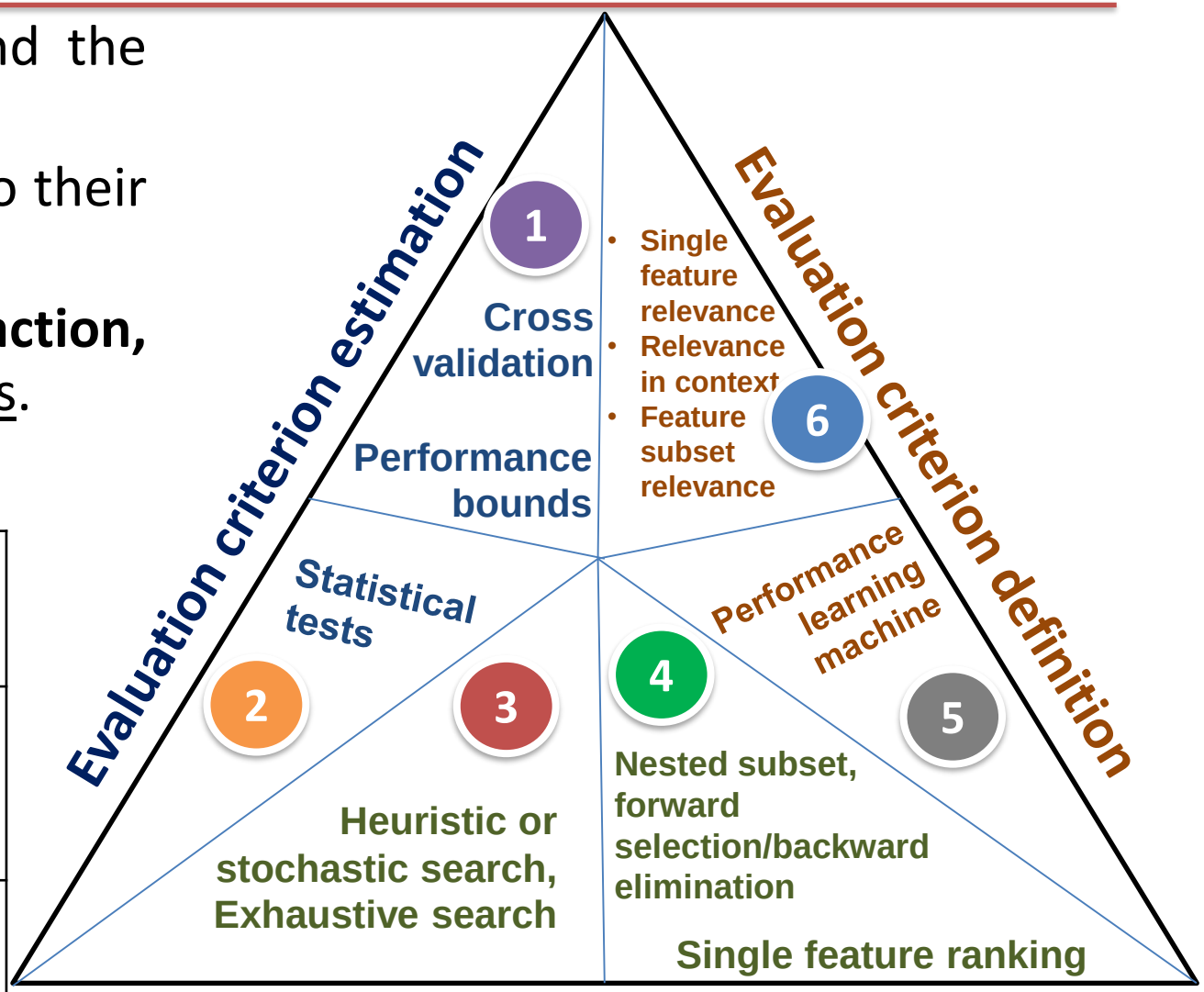
Feature fusion is the process of combining two or more **feature** vectors to obtain a single **feature** vector



2.6 Feature Selection

- Feature selection is a process to find the most productive features subset.
  - Attributes are ranked according to their predictive significance.
  - It is different from **feature extraction,** in which attributes are transforms.
- Common features selection methods:

Filter 2 3 4 6	Correlation-based, Pearson, Mutual Information, Relieff
Wrapper 1 3 4 5	Sequential (Forward or Backward) feature selection, Recursive feature elimination
Embedded 1 4 5	Decision trees, Artificial neural networks



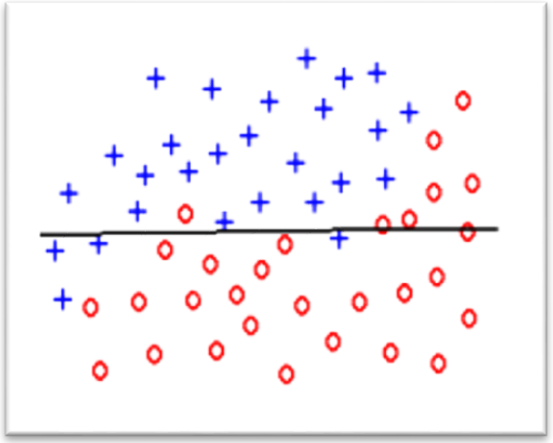
2.7 Data Normalization

- **Min-Max Normalization:** $\boldsymbol{n = [\frac{s - \min(S)}{\max(S) - \min(S)}]}$
  - Maps the raw scores to the [0, 1] range.
  - $\mathrm{Max}(S)$ and $\mathrm{Min}(S)$ specify the end points of the score range.
- **Z-score Normalization:** $\boldsymbol{n = [\frac{s - \mathrm{mean}(S)}{std(S)}]}$
  - Transforms the scores to a distribution with mean of 0 and standard deviation of 1
- **Tanh Normalization:** $\boldsymbol{n = [\frac{1}{2} * \{tanh(0.01 * (\frac{s - \mathrm{mean}(S)}{std(S)})) + 1\}]}$
  - Maps the scores to the (0, 1) range

2.8 Data Underfit Vs. Overfit

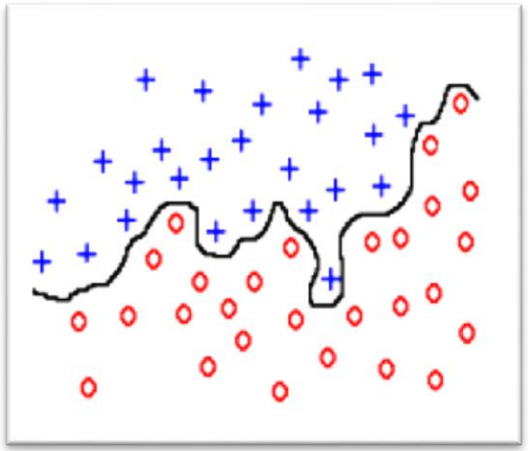
- Classifier has to have the ability to generalize.
- Learning the training data too precisely usually leads to poor classification results on new data.

Underfit



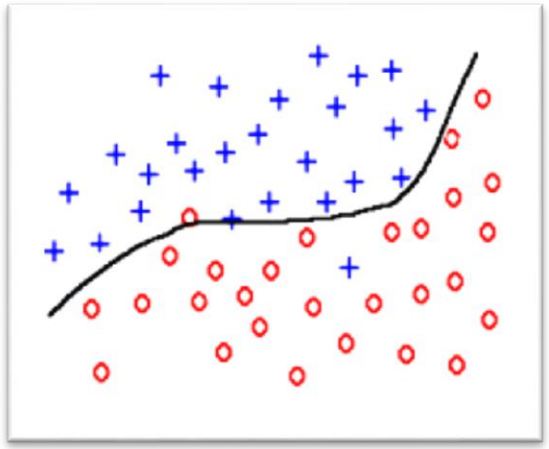
- Model is too **simple** to represent all the relevant class characteristics
- High bias and low variance
- High training error and high test error

Overfit

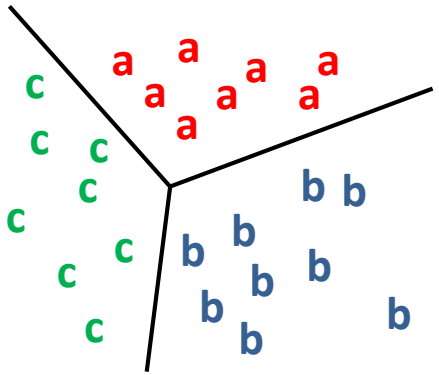


- Model is too **complex** and fits irrelevant characteristics (noise) in the data
- Low bias and high variance
- Low training error and high test error

Fit



2.9 Classification Models

Multi-class	Binary	One-class
Model classifies more than two classes  • Classes are mutually exclusive • Each new instance belongs to a single class • NB, KNN, DT, Logistic	Model classifies the objects into one of two buckets. E.g., Is an email spam or not spam?, Is the credit card transaction fraudulent or genuine? • The two classes are often referred to as the *positive class* and the *negative class*. • SVM, logistic, perception	Model is trained with **positive samples** of a single class • Not between multiple classes, or Between the positive and negative samples of the same class • Two classes are called the *target* and the *outlier* class • Support Vector, K-means, Auto-encoder-NN, Gaussian density

Evaluation Methods

Holdout Set

- The available data set *D* is divided into two disjoint subsets

Training set	D_{train}	For training the model
Test set	D_{test}	For testing the model

- Training set should not be used in testing
- Similarly, the test set should not be used for training *(Unseen test set provides a unbiased estimate of accuracy)*
- The test set is also called the holdout set.
- This method is mainly used when the data set *D* is large.

k-fold cross-validation

- The training set is randomly divided into *K* disjoint sets of equal size, where each part has roughly the same class distribution.
- The classifier is trained *K* times, each time with a different set held out as a test set.
- The final estimated accuracy of learning is the average of the *k* accuracies.
- 10-fold and 5-fold cross-validations are commonly used.
- This method is used when the available data is not large.

Leave-one-out cross-validation

- This method is used when the data set is very small.
- It is a special case of *K-fold* cross-validation with ***K = n***,
- *n* experiments are performed using *n* − 1 samples for training and the remaining sample for testing.
- The final estimated accuracy of learning is the average of the *n*-accuracies.
- It is rather computationally expensive.
- Leave-one-out cross-validation does not guarantee the same class distribution in training and test data.

Validation Set

- The available data is divided into three subsets,
  - a training set,
  - a validation set, and
  - a test set.
- A validation set is used frequently for estimating parameters in learning algorithms.
- In such cases, the values that give the best accuracy on the validation set are used as the final parameter values.
- Cross-validation can be used for parameter estimating as well.

Performance Metrics

- **True Acceptance Rate (TAR)**
  - Ratio of correctly accepted owner's attempts to all the attempts made.
  - Higher TAR indicates that the system performs better in recognizing a legitimate user.
- **False Rejection Rate (FRR)**
  - Ratio of incorrectly rejected attempts of a legitimate user to all the attempts made.
  - Calculated as FRR = 1 - TAR.
- **False Acceptance Rate (FAR)**
  - Ratio of incorrectly accepted impostor attempts to all the attempts made.
  - Lower FAR means the system is robust to impostor attempts.
- **True Rejection Rate (TRR)**
  - Ratio of correctly rejected attempts of impostors to all the attempts made.
  - Calculated as TRR = 1 - FAR.
- **Receiver- or Relative-Operating Characteristic (ROC)**
  - ROC plot is a visual characterization of trade-o between FAR and TAR.
  - In simple words, it is a plot between true alarms vs. false alarm.
  - The curve is generated by plotting the FAR versus the TAR for varying thresholds to assess classifier's performance.

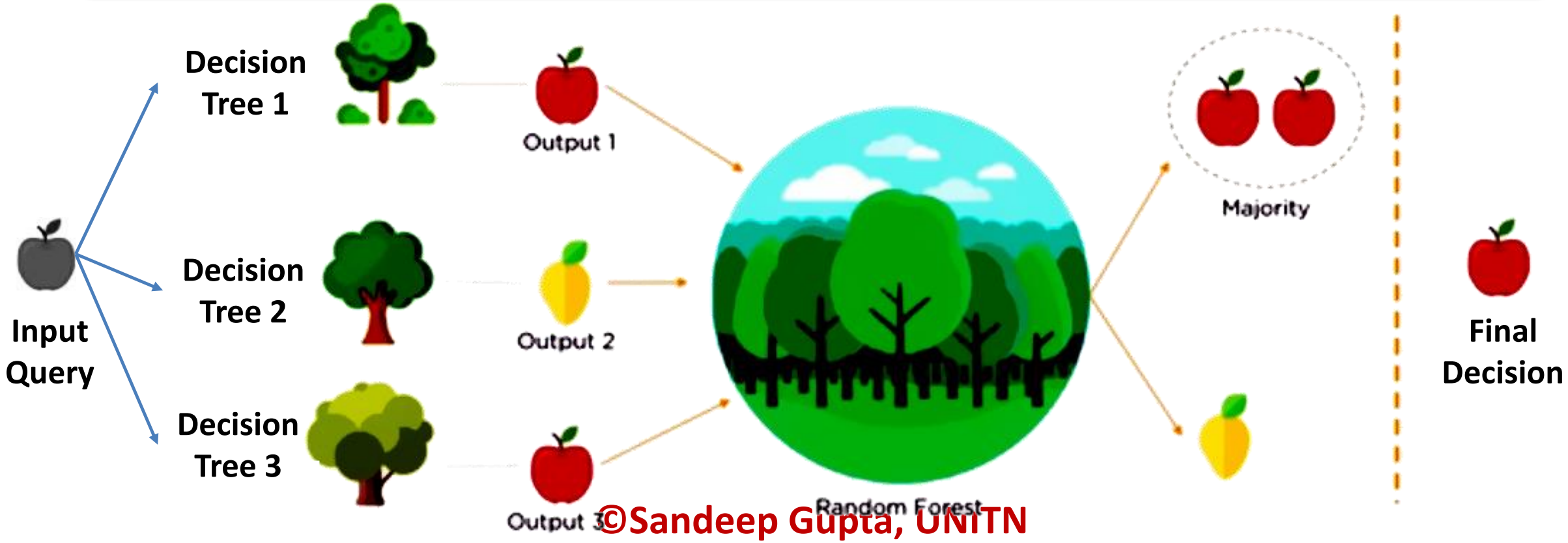
Classifiers

Random Forest Classifier

- Random forest classifier is collection of ***decision trees*** in the ensemble.
- ***Decision trees*** are generated by *randomly selecting* the ***attributes*** at each ***node*** to determine the ***split***
- During the ***classification***, each tree ***votes***
  - The class with ***maximum votes*** is accepted as the final decision
- Common methods to construct Random Forest [*Breiman 2001*] are:
  - Forest-RI (*random input selection*)
  - Forest-RC (*random linear combinations*)

What is Random Forest?

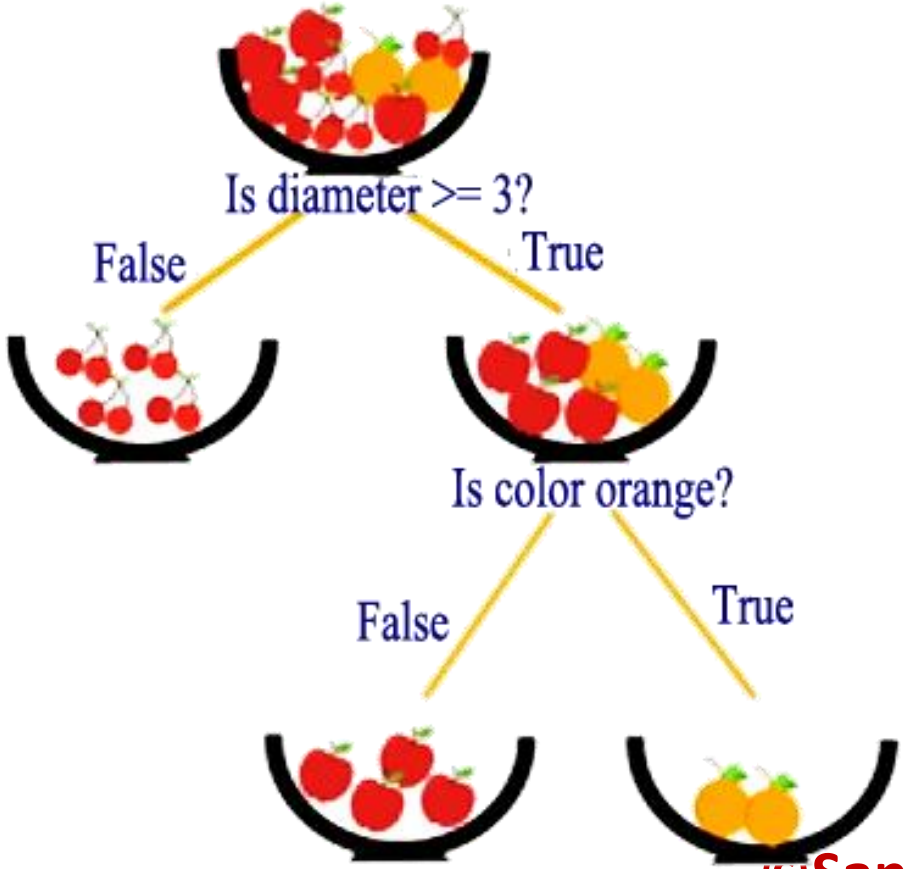
- Random forest or Random Decision Forest is a method that operates by constructing multiple "Decision Trees" during the training phase.
- The decision of majority of the trees is chosen by the random forest as the final decision.



23

Decision Tree

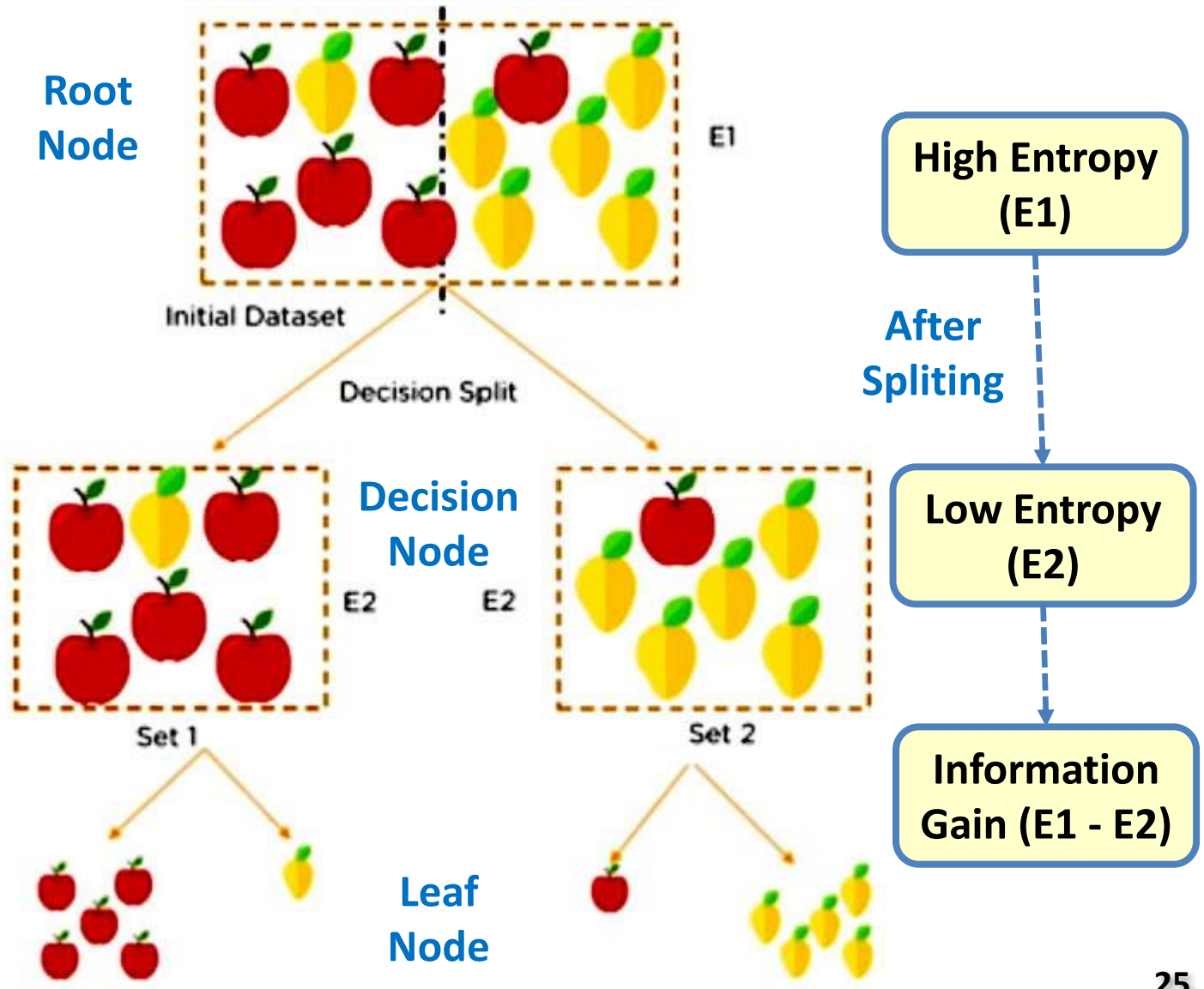
- **Decision Tree is a flowchart like tree structure that determines a course of action.**
- **Each branch (node) of the tree represents a possible decision, occurrence or reaction.**



- Here we have a bowl of fruit having cheries, apples, and oranges
- The first decision to split – 'Is diameter >= 3?'
- The second decision to split – 'Is color orange?'

Decision Tree: Important terms

- **Decision node** has 2 or more branches
- **Root node** is the top most decision node
- **Leaf node** carries the classification or the decision
- **Entropy** (Entropy is the measure of randomness or unpredicability)
- **Information gain** (It is the measure of decrease in entropy after the dataset is split)



Advantages of RF Classifier

Resistance against overfitting

- Use of multiple trees reduce the risk of overfitting.

Training time is less

- Shorter computation time.

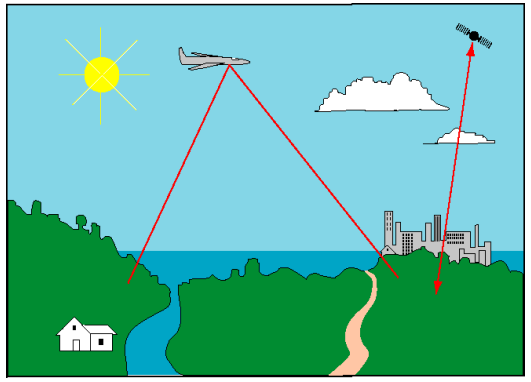
Random forrest estimates missing data

- Maintains accuracy when a large proportion of data is missing.

Runs efficiently on large datasets

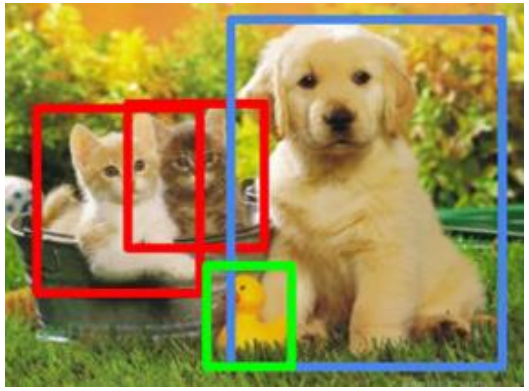
- Produces highly accurate results

Applications of RF Classifier



Remote Sensing

Used in ETM devices to acquire images of the earth's surface



Object Detection

Multiclass object detection

Kinect

Used in game console called Kinect

RF: Configuring in Matlab

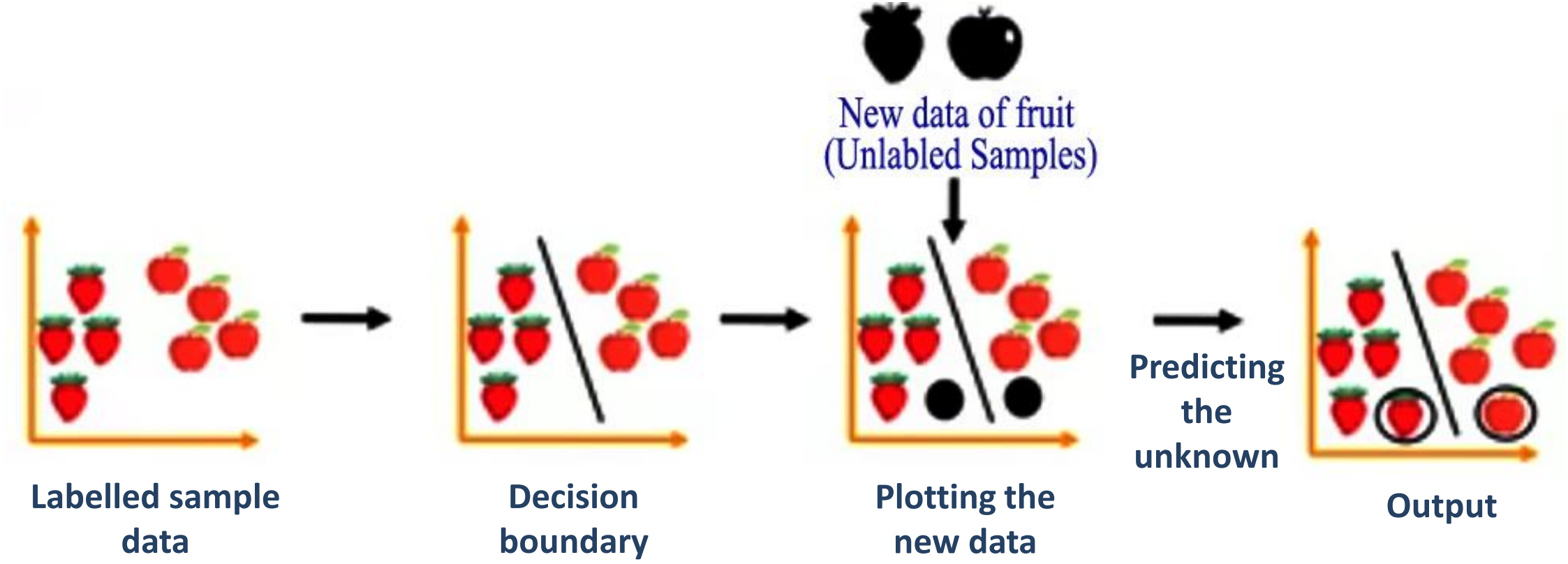
randomforestc (A, L, N)

- A is Dataset used for training
- L is the number of decision trees to be generated (default 200)
- N is the size of feature subsets to be used (default 1). Each decision tree is using random feature subsets of size N in each node.

L. Breiman, Random Forests, Machine learning, vol. 45 (1), 5-32, 2001

Support Vector Machine

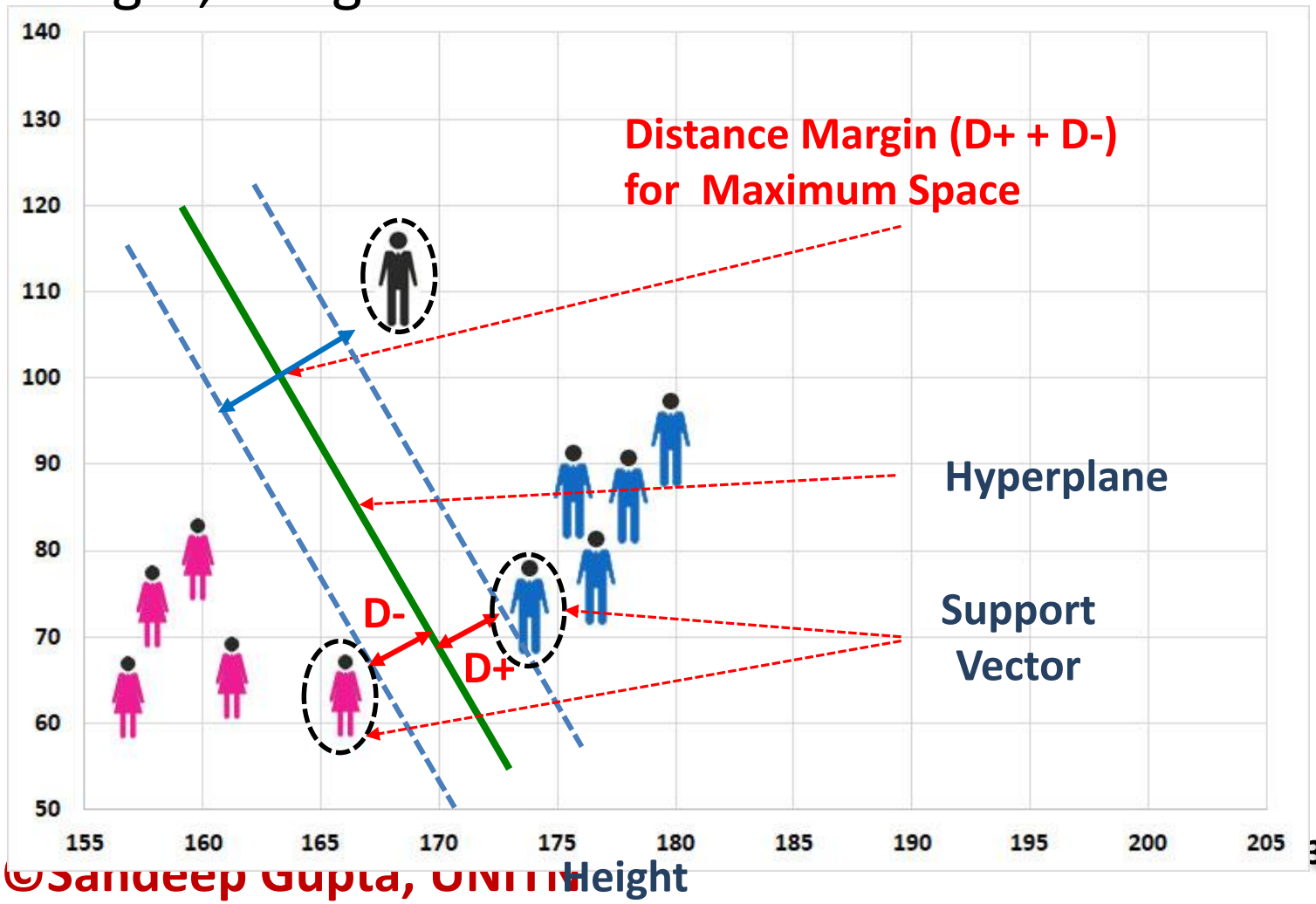
- SVM is a supervised learning method that looks at data and sort into one of the two categories.



SVM: Important terms

Height (cm)	Weight (kg)	Label
174	60	*Female*
174	88	*Female*
175	75	*Female*
180	65	*Female*
185	80	*Female*
179	90	*Male*
180	80	*Male*
183	80	*Male*
187	85	*Male*
182	72	*Male*

- We are given with a set of people with different Height, Weight.



SVM: Determining hyperplane

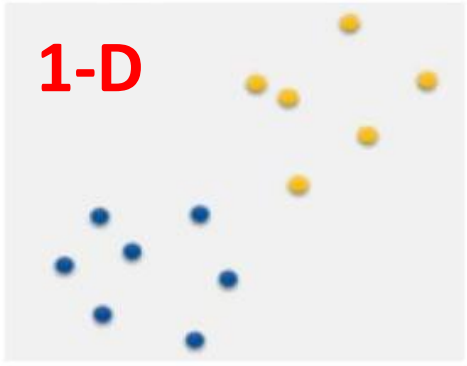


Figure A

Figure B

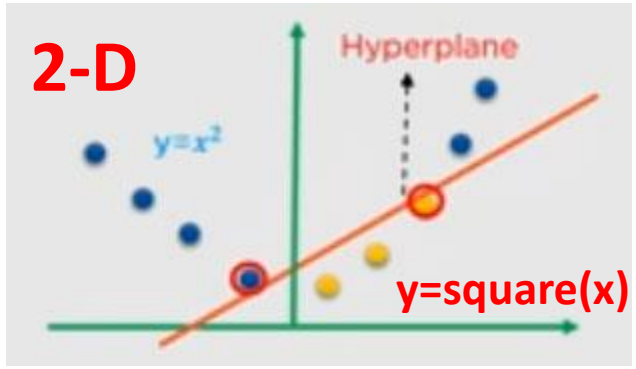


Figure C

Figure D

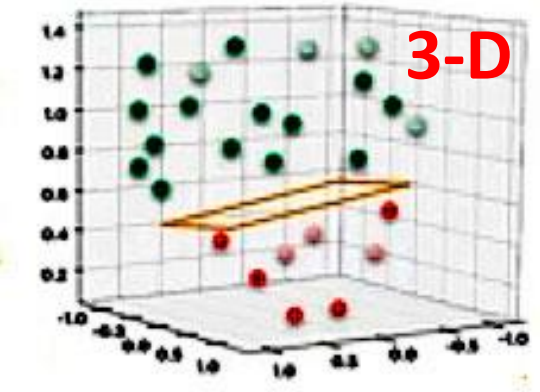


Figure E

Advantages of SVM Classifier

High dimensional input space (curse of the dimensionality)

- SVM automatically address it

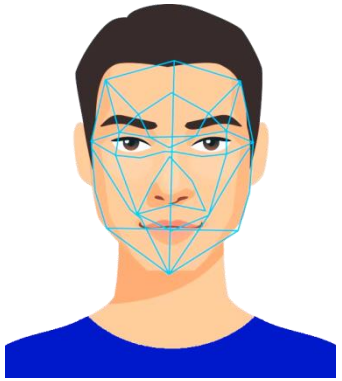
Sparse document vectors

- Can handle millions of tokens

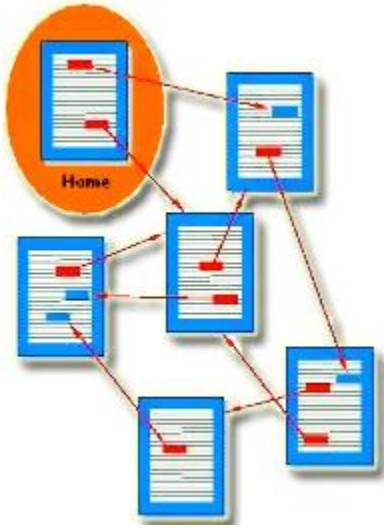
Regularization parameter (*Lambda*)

- SVM naturally avoids the bias and overfitting problems

Applications of SVM



Face Detection



Text and hypertext categorization

Classification of images

Bioinformatics

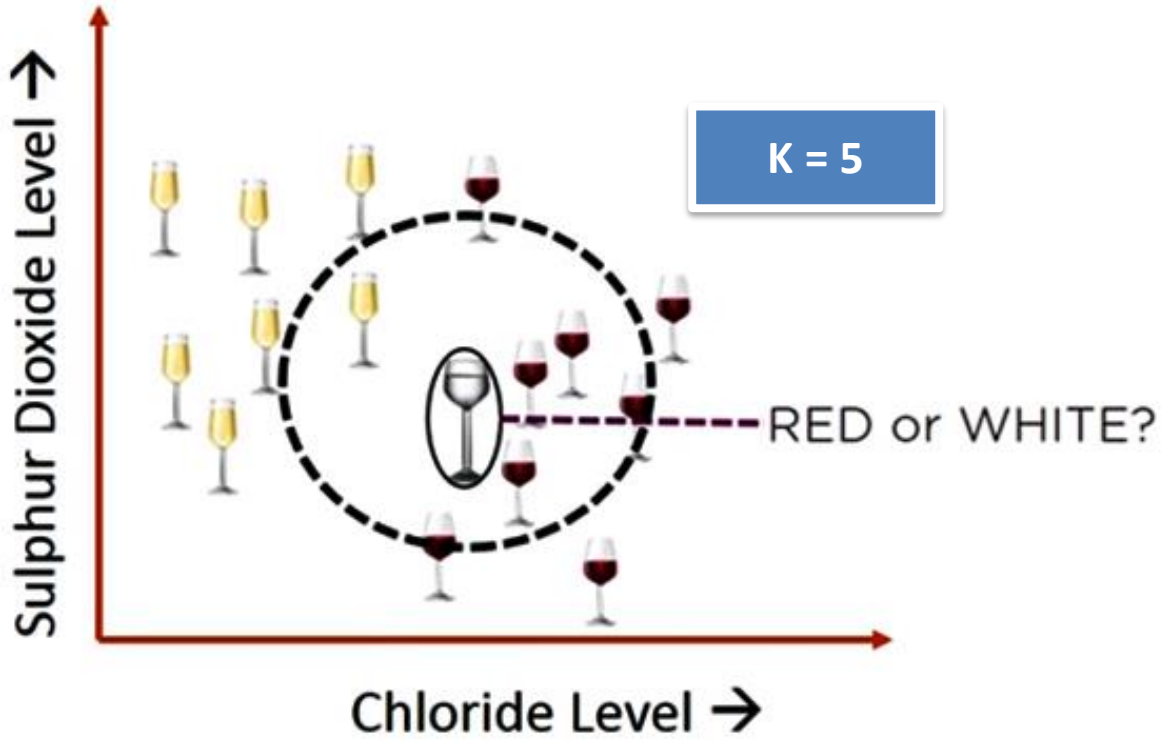
SVM: Configuring in Matlab

[W, J] = **svc**(A, KERNEL, C)

- A is the Dataset
- KERNEL Type, Default: linear kernel
- C Regularisation parameter (optional; default: 1) OUTPUT W Mapping: Support Vector Classifier J Object indices of support objects
- W Mapping: Support Vector Classifier
- J Object indices of support objects

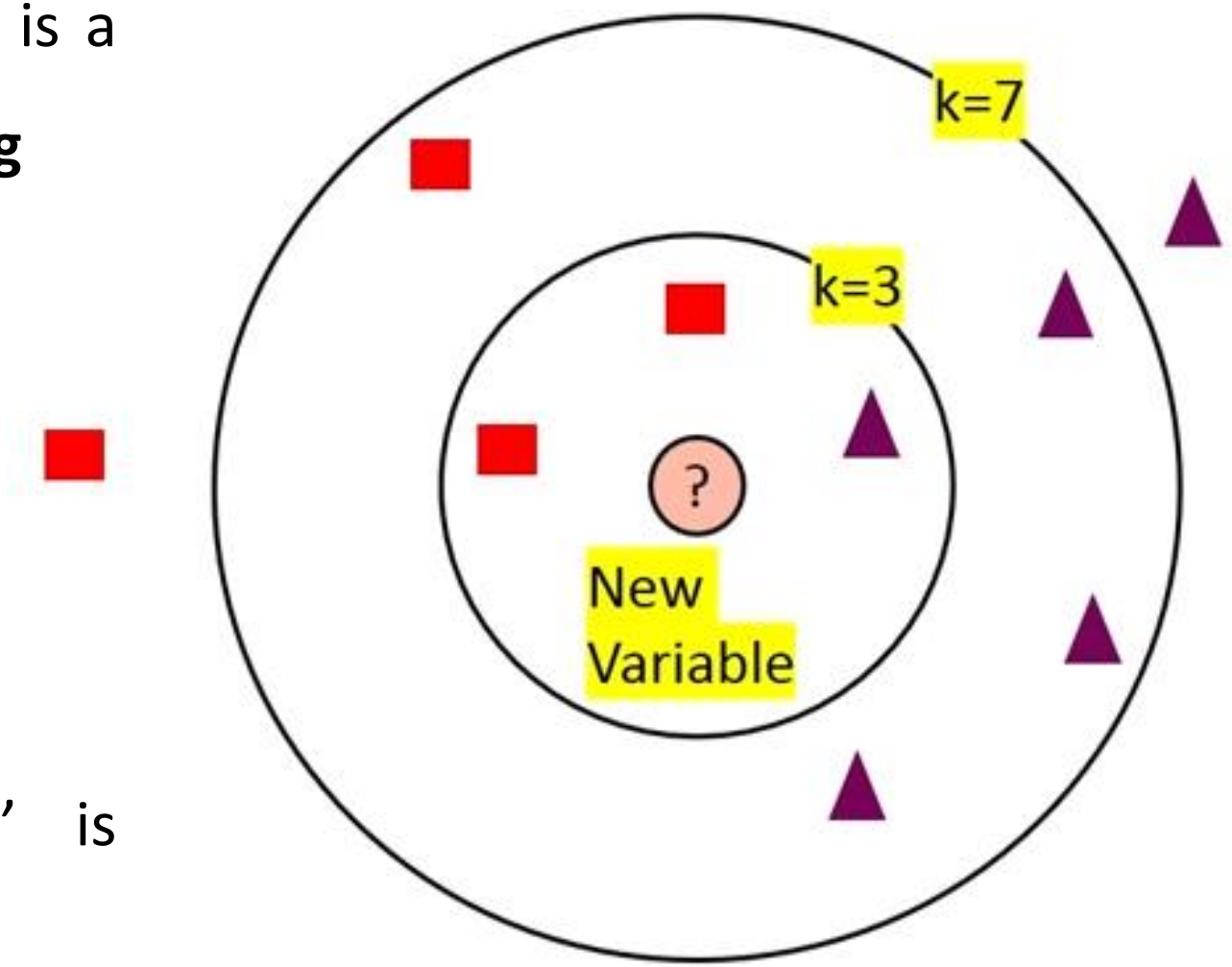
K-Nearest Neighbour Algorithm (KNN)

- **KNN is based on feature similarlity**
  - It classifies a data point based on how its neighbours are classified
  - It stores all the available cases and classifies new cases based a similarity measurement



Choose the factor ‘K’

- Choosing the right value of ‘**k**’ is a process called **Parameter Tuning**
  - *If (K = = 3)*
    - *Then **?** =* Square
  - *If (K = = 7)*
    - *Then **?** =* Triangle
- Choosing right value of ‘**K**’ is important for better accuracy.



How to choose the factor ‘K’?

To choose a value of ‘k’

Sqrt(n)

where, n is the total number of data points

Odd value of K is selected to avoid confusion between two classes of data

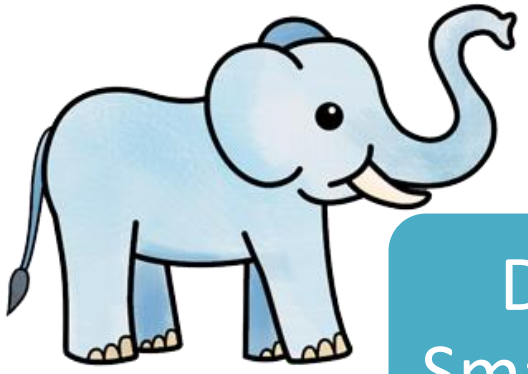
Higher value of k has lesser chance of error

When do we use KNN?

Data is labeled

Cat Dog

We can use KNN when



Dataset is Small *(few GB)*

Because KNN is a lazy learner i.e. It doesn't learn a discriminative function from the training set

Data is noise free

Weight(x2)	Height(y2)	Class
51	167	Underweight
62	182	one-fourty
69	176	23
64	173	hello kitty
65	172	Normal

Noise

Classes are ambiguish

Applications of KNN

Healthcare

Agriculture

Text Mining

Financial

KNN: Configuring in Matlab

[W,K,E] = **knnc**(A, K)

- A is the dataset
- K is the number of the nearest neighbours
  - default: K is optimised with respect to the leave-one-out error on A
- W k-NN classifier
- K is the Number of the nearest neighbours used
- E The leave-one-out error of the **knnc**

Naive Bayes Classifier

- **Naive Bayes classifier works on the principles of conditional probability as given by Bayes' Theorem.**

- **Example: Tossing two coins**
  - Sample Space: *{HH, HT, TH, TT}*
  - P(*Getting two heads*) = ¼
  - P(*Atleast one tail*) = ¾
  - P(*Second coin being head given the first coin is tail*) = ½
  - P(*Getting two heads given the first coin is a head*) = ½

Bayes Theorem

Bayes' theorem gives the conditional probability of an event A given another event B has occured.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) = Conditional Probability of A given B

P(B|A) = Conditional Probability of B given A

P(B) = Probability of event A

P(B) = Probability of event B

Tossing two coins: Simple Probability

- **Sample Space:** ***{HH, HT, TH, TT}***
  - P(Getting two heads) = ¼
  - P(Atleast one tail) = ¾

These two use simple probabilities calculated directly from the sample space

  - P(Second coin being head given first coin is tail) = ½
  - P(Getting two heads given first coin is a head) = ½

Tossing two coins: Conditional Probability

- **Sample Space:** ***{HH, HT, TH, TT}***
  - P(Getting two heads) = ¼
  - P(Atleast one tail) = ¾

This can be obtained conditional probability.

  - P(Second coin being head given first coin is tail) = ½
  - P(Getting two heads given first coin is a head) = ½

Tossing two coins: Applying Bayes Theorem

Let **A** be the event that second coin is head and **B** be the event that the first coin is tail in the sample Space: *{HH, HT, TH, TT}*

P (Second coin being head given first coin is tail)

=> P(A|B) = [P(A|B) x P(A)] / P(B)

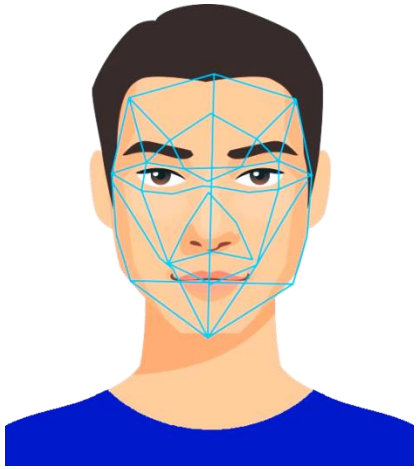
= [P(First coin being tail given second coin is head) x P(Second coin being head)] / P(First coin being tail)

= [(½) x (½)] / (½) = ½ = 0.5

Advantages of Naive Bayes Classifier

- Needs less training data
- Handles both continous and discrete data
- Highly scalable with number of predictors and data points
- Not senstive to irrelevant features
- As it is fast, it can be used in real-time predictions
- Very simple and easy to implement

Applications of NB Classifier



Face Recognition

Weather Prediction

News Classification

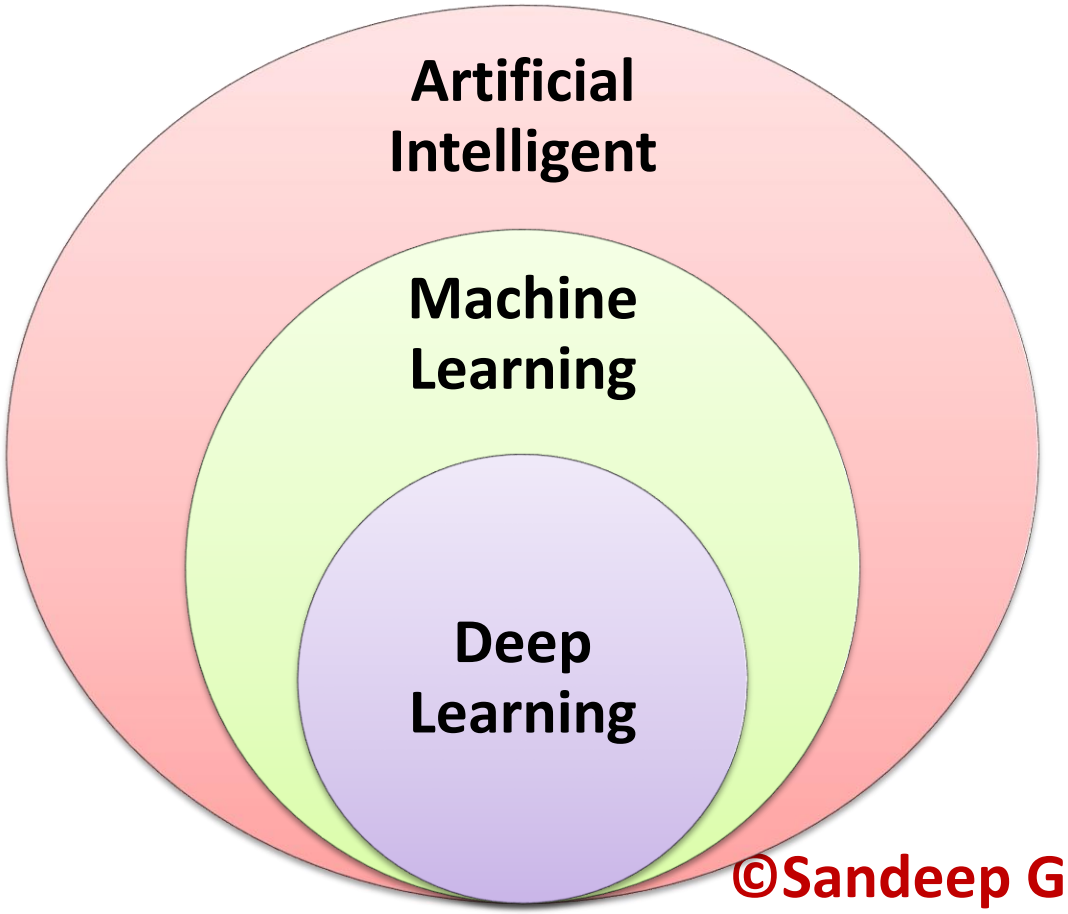
NB: Configuring in Matlab

W = **naivebc**(A,N)

- A is training dataset
- N Scalar number of bins (default: 10) e.g., DENSMAP the Untrained mapping for density estimation
- W is the Naive Bayes classifier mapping

Deep learning

Deep Learning is a subfield of Machine Learning that deals with algorithms inspired by the structure and functions of brain.



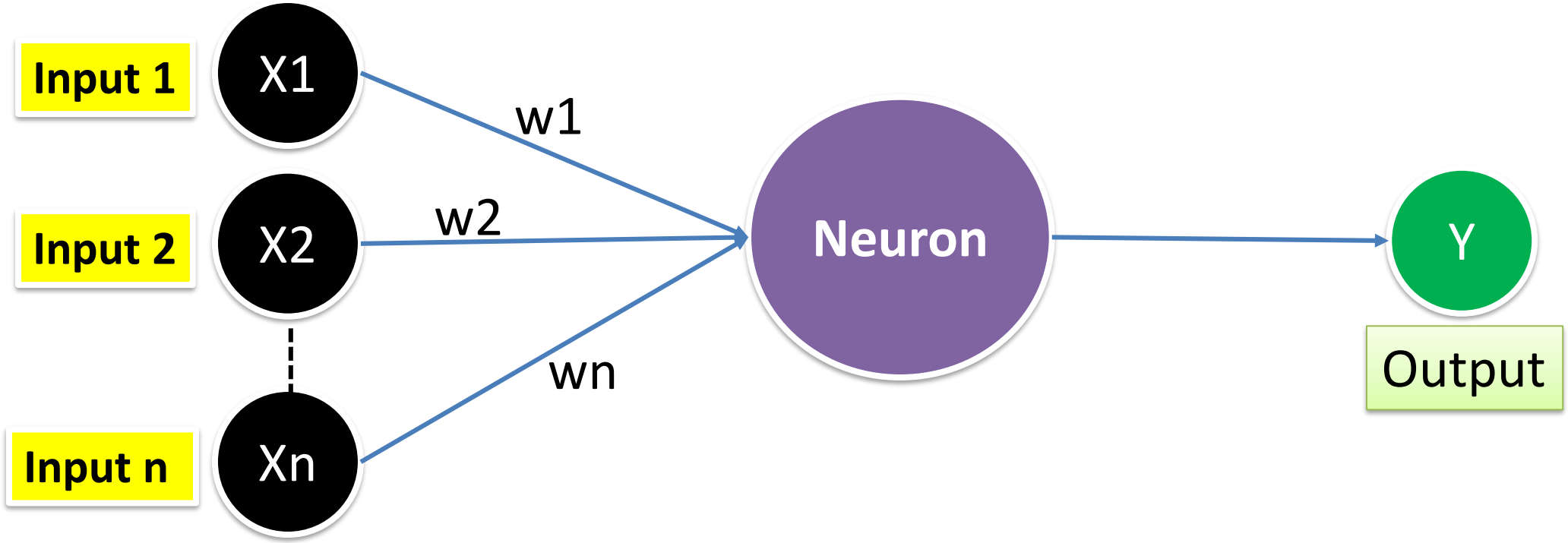
Abiity of machines to imitate intelligent human behavior

Application of AI that allows a system to automatically learn and improve from experience

Application of machine learning that uses complex algorithms and deep neural nets to train a model

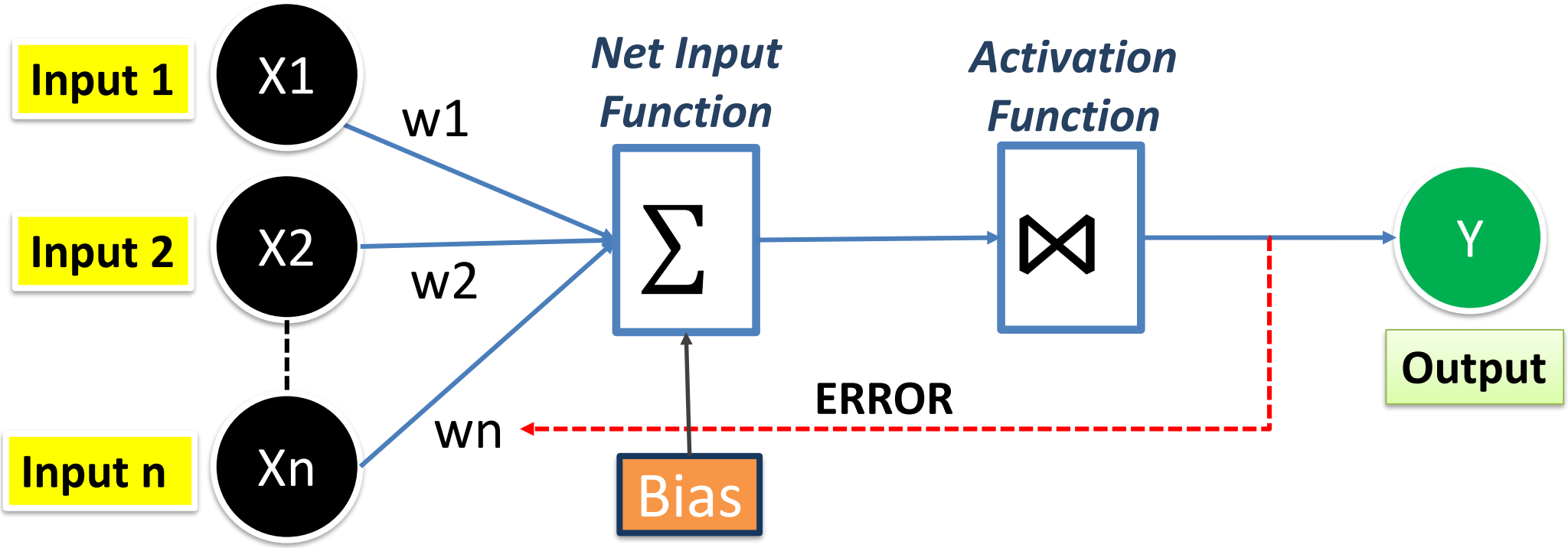
What is a Neural Networks?

- Neural networks are inspired by a human brain.
- Neural Network has interconnected neurons that receive some inputs, processes those inputs in layers to produce the output.



What is a Perceptron?

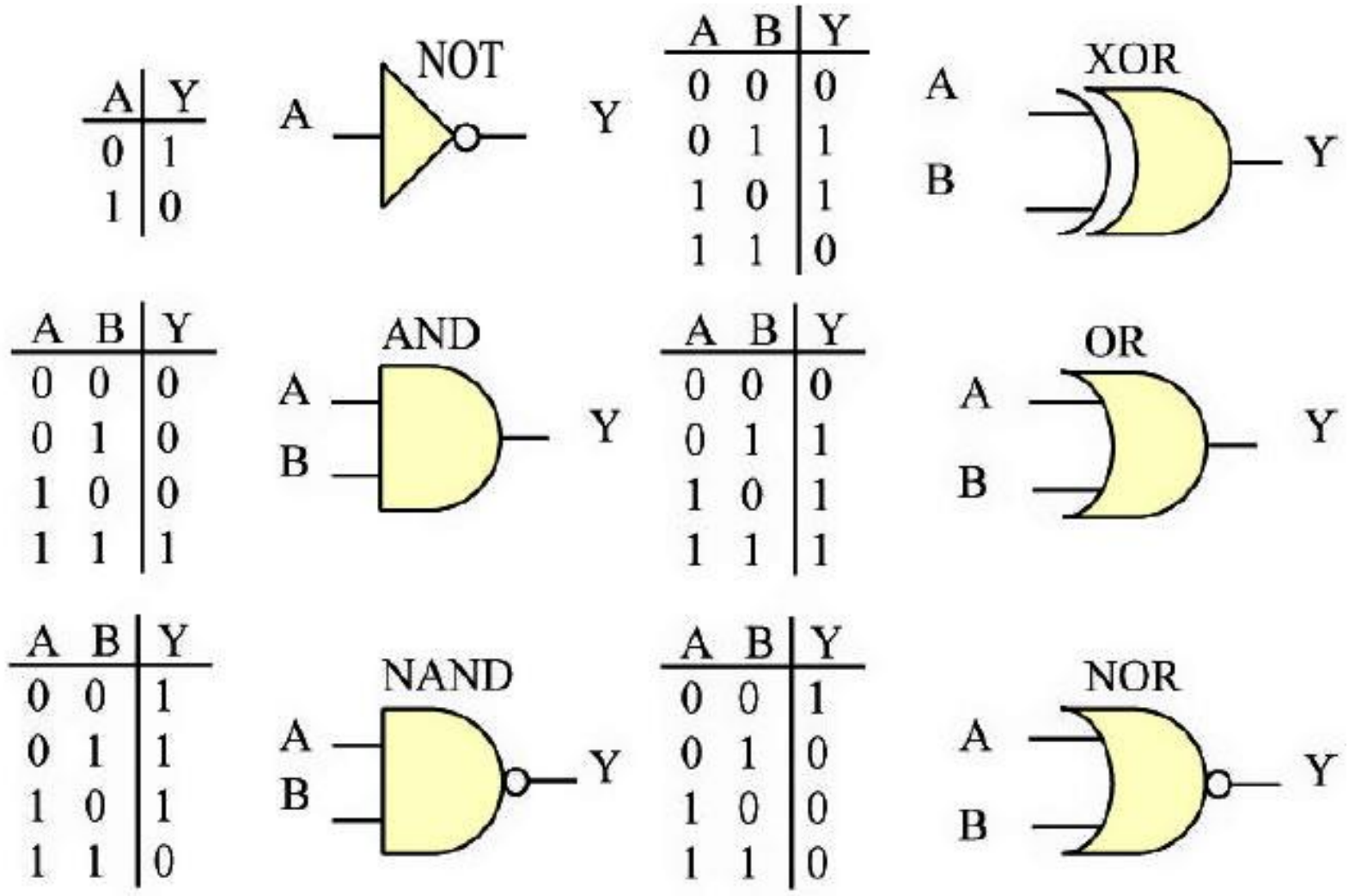
- A Perceptron is the fundamental part of a neural network.
- It represents a single artificial neuron
- It can be used for basic functions like binary classification



Weighted Sum (WS) = $\sum_{i=1}^{n} w_i * x_i + b$

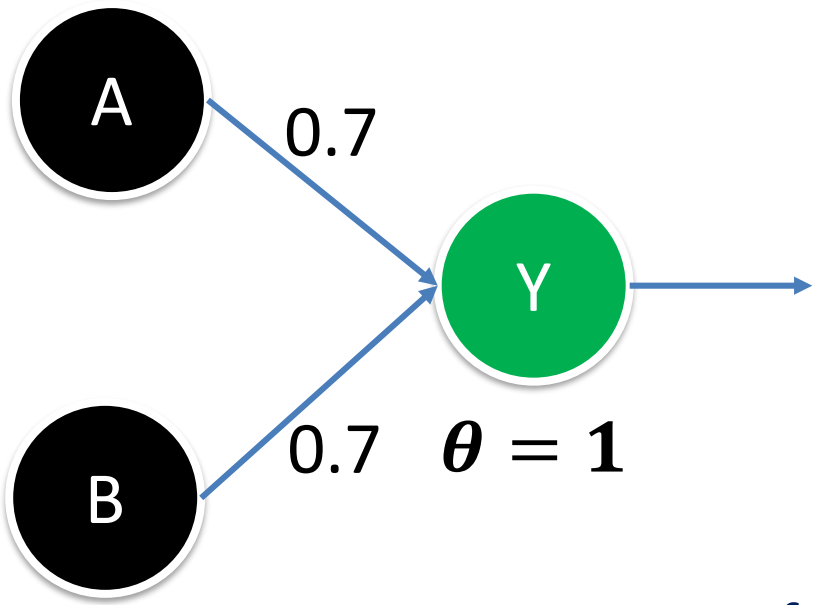
Implementing Logic Gates using Perceptron

- A logic gate is the basic building block of a **digital circuit**.
- Most logic gates have 2 inputs and 1 output.
- At any given moment, every terminal is in one of the two binary conditions low (0) or high (1), represented by different voltage levels.



Implementing AND Gate using Perceptron

- A and B are input neurons.
- The green neuron is our output neuron.
- The threshold value is 1.

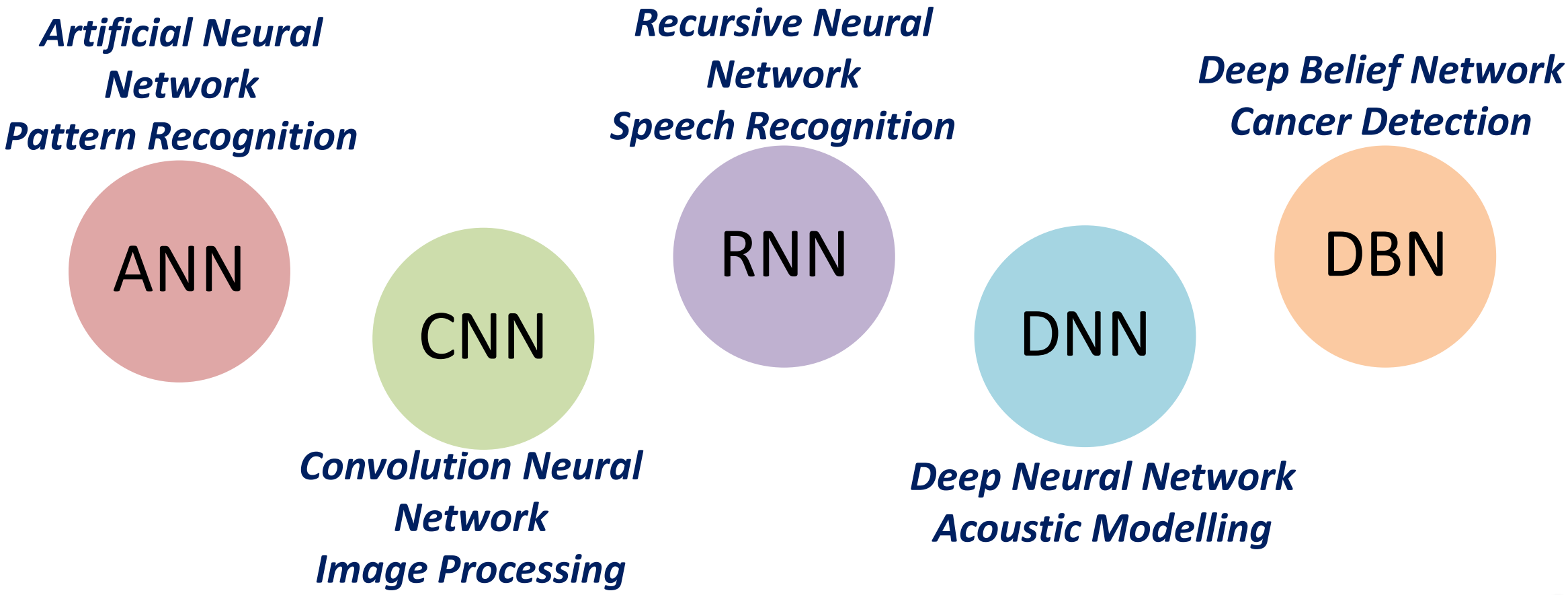


Input (A)	Input (B)	Output
$w_1 * x_1$	$w_2 * x_2$	$w_1 * x_1 + w_2 * x_2$
0.7*0	0.7*0	0
0.7*0	0.7*1	0.7
0.7*1	0.7*0	0.7
0.7*1	0.7*1	1.4

If the sum of the weighted input neurons is greater than the threshold, the output is 1 else 0

Types of Neural Network

Neural networks can be broadly classified into 5 categories.



Advantages of Deep Learning

Ability to process huge amount of data

- Deep Learning can work with both structured and unstructured data.

Ability to perform complex algorithms

- Deep Learning algorithms can perform complex operations easily.

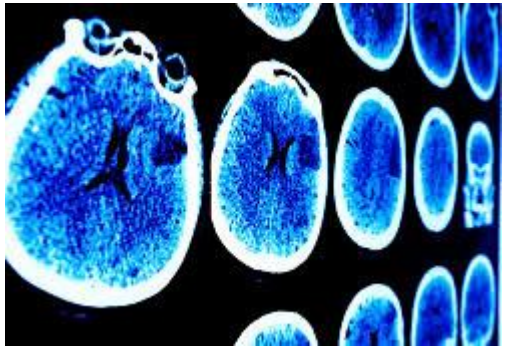
Performs better with a large amount of data

- Performance of Deep Learning algorithms increase as the amount of data increase.

Feature Extraction

- Deep Learning accepts large volumes of data as input, analyse the input to extract features out of an object and identifies similar objects.

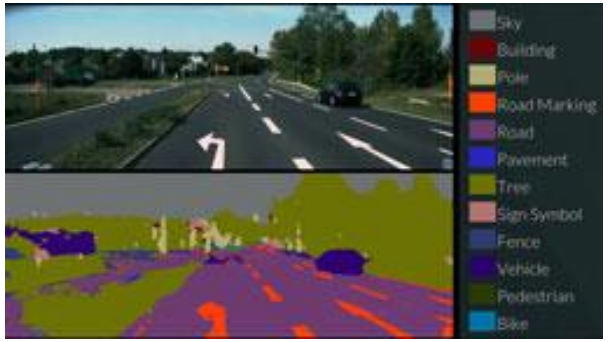
Applications of Deep Learning



Healthcare

Language Translation

Robotics



Autonomous Driving Car

MLP: Configuring in Matlab

bpxnc	Feed-forward network (multi-layer perceptron), trained by a modified back propagation algorithm with a variable learning parameter.
lmnc	Feed-forward network, trained by the Levenberg-Marquardt rule.
rbnc	Radial basis network. This network has always one hidden layer which is extended with more neurons as long as necessary.
rnnc	Feed-forward network (multi-layer perceptron) with a random input layer and a trained output layer.
perlc	Single layer perceptron with linear output and adjustable step sizes and target values.

Thank you

- *This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 675320*
- *This work reflects only the author's view and the Research Executive Agency is not responsible for any use that may be made of the information it contains*