Introduction to Machine Learning & Deep Learning

Nikesh Bajaj, PhD
Research Associate
Imperial College London

http://nikeshbajaj.in

https://spkit.github.io/tutorials

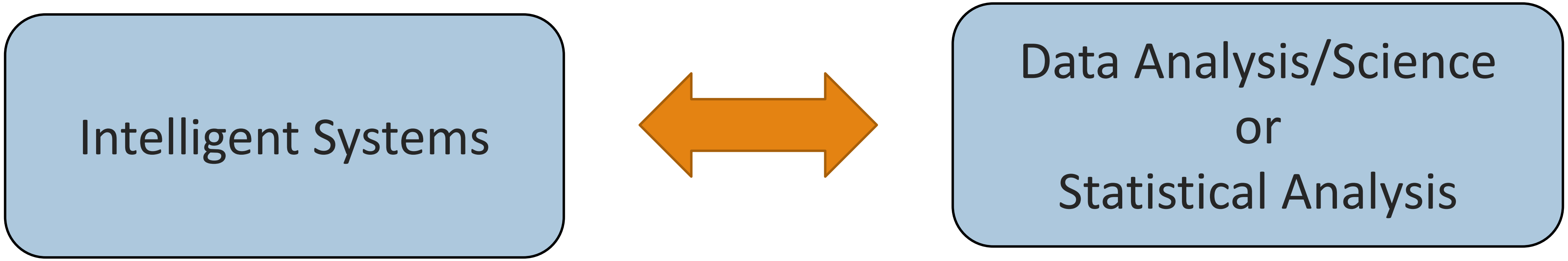
Agenda

- Introduction
- What is (Machine) Learning?
- Types of Machine Learning Problems
  - Supervised Learning
    - Regression
    - Classification
    - Excercises and examples
    - Performance metrics
- Deeplearning
  - Neural Networks
  - Examples
  - Conclusion + Resouces

About me …

- Current work
- PhD Work
- What I like about Machine Learning and AI
- Activities
  - Consultant with deeplearning.ai
  - Mentor at Coursera
  - Competitions, Study groups, Certifications and Courses

What is (machine) learning?



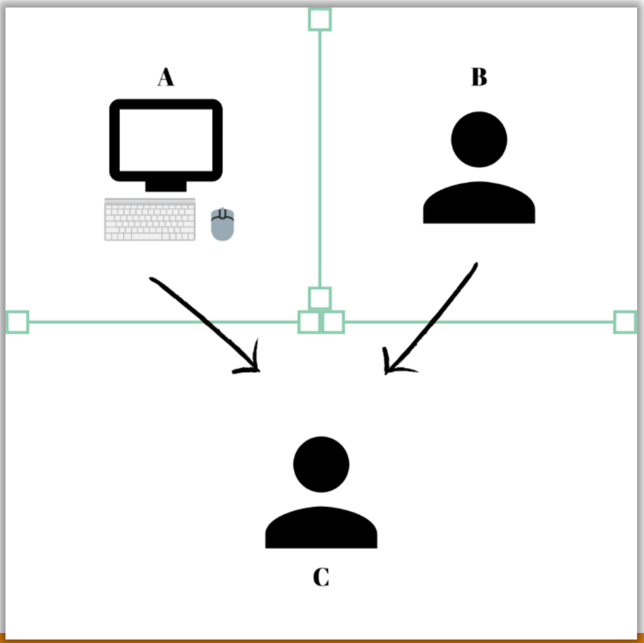
Intelligent Systems

Data Analysis/Science
or
Statistical Analysis

- A way to design intelligent systems using data
- Early intelligent systems and algorithms were build by hand crafted rules.
- Designing a machine that learn from given data. Without hardcoding the rules.

Artificial Intelligence

Turing Test



Fields in Artificial Intelligence

- Computer Vision
- Robotics
- Natural Language Processing
- Speech/Audio Processing
- Knowledge/Data Representation
- Machine Learning
- .
- .

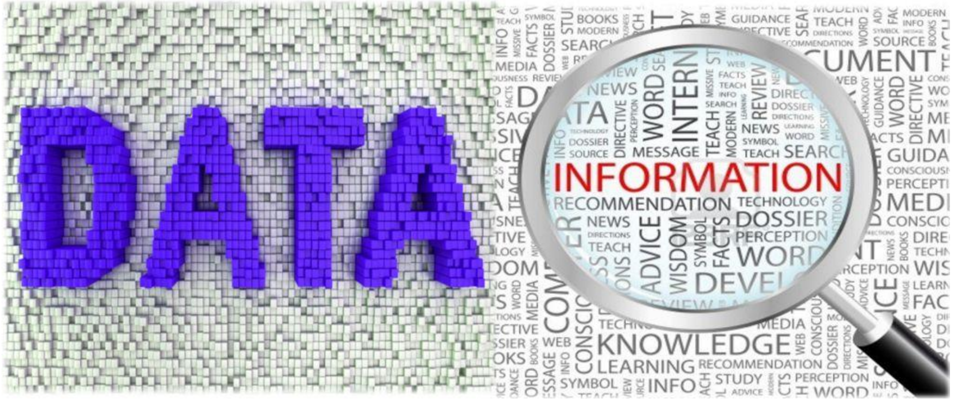
Examples (classical)

- Automatic face tagging
- Speech to text, Translation
- Face/emotion recognition
- Sentiment analysis
- Music classification
- Movie/product recommendation
- Search engine
- IoT
- … many more

How do we do that?

Computer(s)

Data (signal)



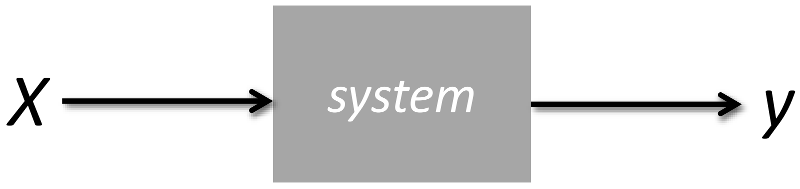
How do you know, if it is working?

- A machine is said to ***learn*** from experience **E**, with respect to task **T** and some performance measure **P**, if machine improves on task **T**, as measured by **P**, with respect to experience **E**. (Tom Mitchell (1998))*.
- **Properly formulating** the a problem and choosing a performance measure
- One of the **common mistakes** that I encounter, that people overlook above definition or work with poorly formulated problem
- **Good News:** A lot of people, in this field have already done this part for you. You can start from textbook examples to well defined problems and performance measures.

** from Coursera*

Machine Learning Problems

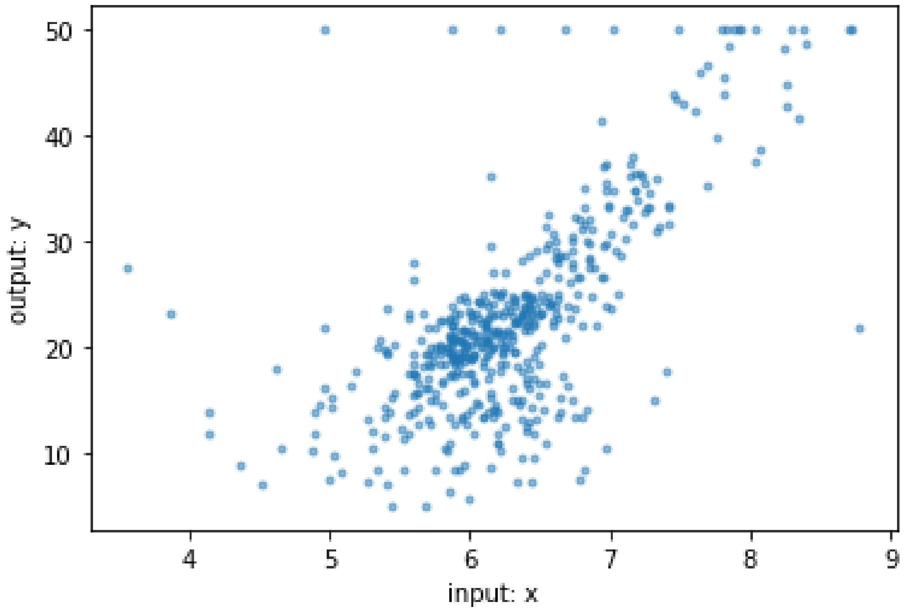
- Supervised Learning
  - *We know what we are looking for*
- Unsupervised Learning
  - *We want to find something meaningful*
- Semi-supervised learning
  - Weakly supervised learning
- Reinforcement Learning & Recommender systems

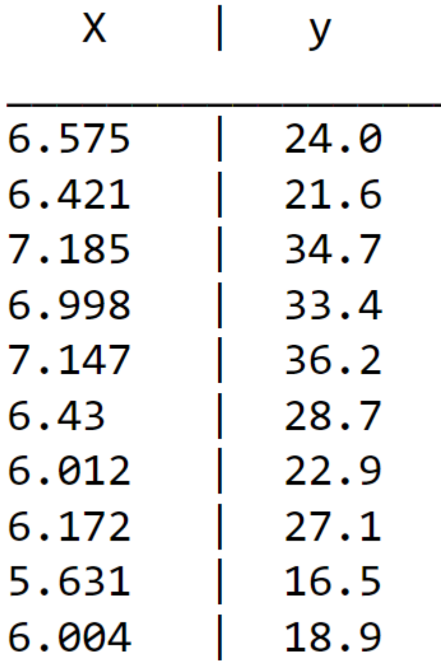


Supervised Learning

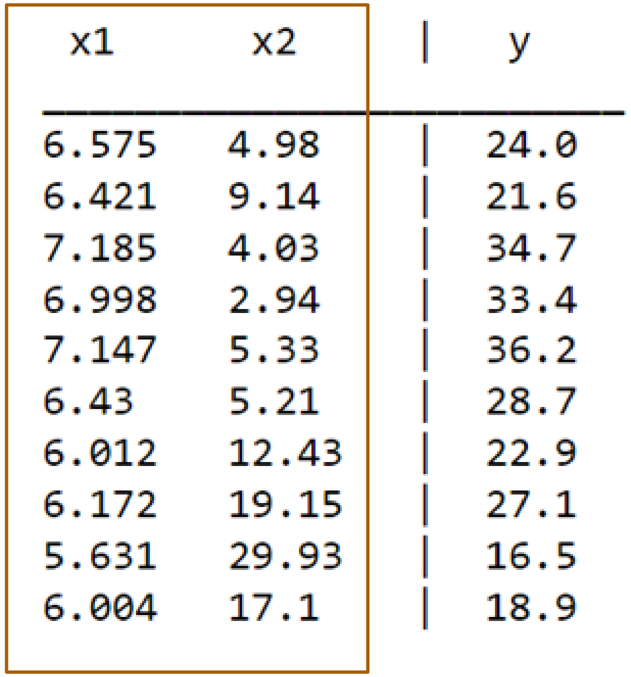
Given data – X, and corresponding target value – y, **think it as X,y problem.** *"X maps to y"*

Examples: Boston House price*





```
   X      |   y
_______________________
6.575     |  24.0
6.421     |  21.6
7.185     |  34.7
6.998     |  33.4
7.147     |  36.2
6.43      |  28.7
6.012     |  22.9
6.172     |  27.1
5.631     |  16.5
6.004     |  18.9
```



```
  x1      x2      |   y
__________________________
6.575   4.98      |  24.0
6.421   9.14      |  21.6
7.185   4.03      |  34.7
6.998   2.94      |  33.4
7.147   5.33      |  36.2
6.43    5.21      |  28.7
6.012   12.43     |  22.9
6.172   19.15     |  27.1
5.631   29.93     |  16.5
6.004   17.1      |  18.9
```

$\boldsymbol{X} \in \boldsymbol{R}^n$

**from scikit-learn*

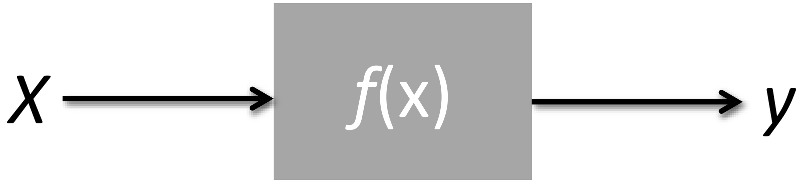
- Using LMS
- Using Gradient Decent

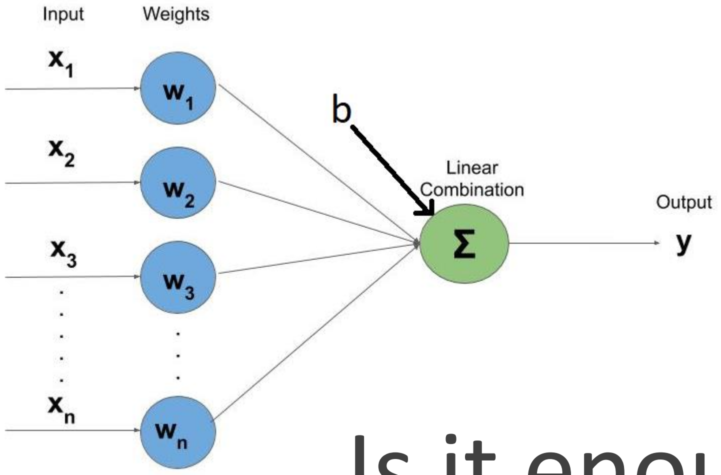
Let's try LMS

Supervised Learning

Problem (simplified way):

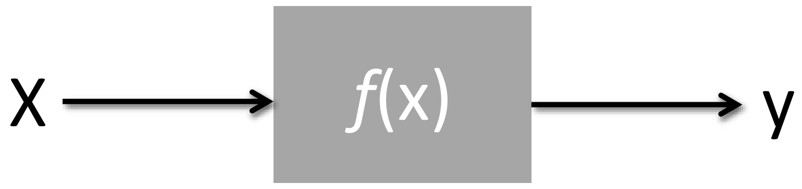
- Given *X, y*
- Find a function *f*, such that *y = f(X)*
- *f:* can be any function, linear, non-linear, etc
- Example: a linear function
  - $y' = f(X)$
  - $y' = b + w_1x_1 + w_2x_2 \, . \, . \, . \, w_nx_n$
  - *Optimise parameters **w***
  - Such that $y' \approx y$, *close enough**
  - *This is simply an optimisation problem*





Is it enough??

Supervised Learning



Difference between Learning and memorising

- A simple choice: *f* can be a look-up table, a perfect mapping of *X* to *y*

  Issue: wouldn’t know what to do with new values of *X* (unseen data - X_u)

- Not so simple choice: *f* can be a very large and complex function (e.g. deeeeep neural network). Is it good?

  Issue: lack of generality, wouldn’t do good on unseen data X_u *: Overfitting*

Supervised Learning

<u>Solution: </u>Problem (properly)

For given *X, y,* find function *f as to*

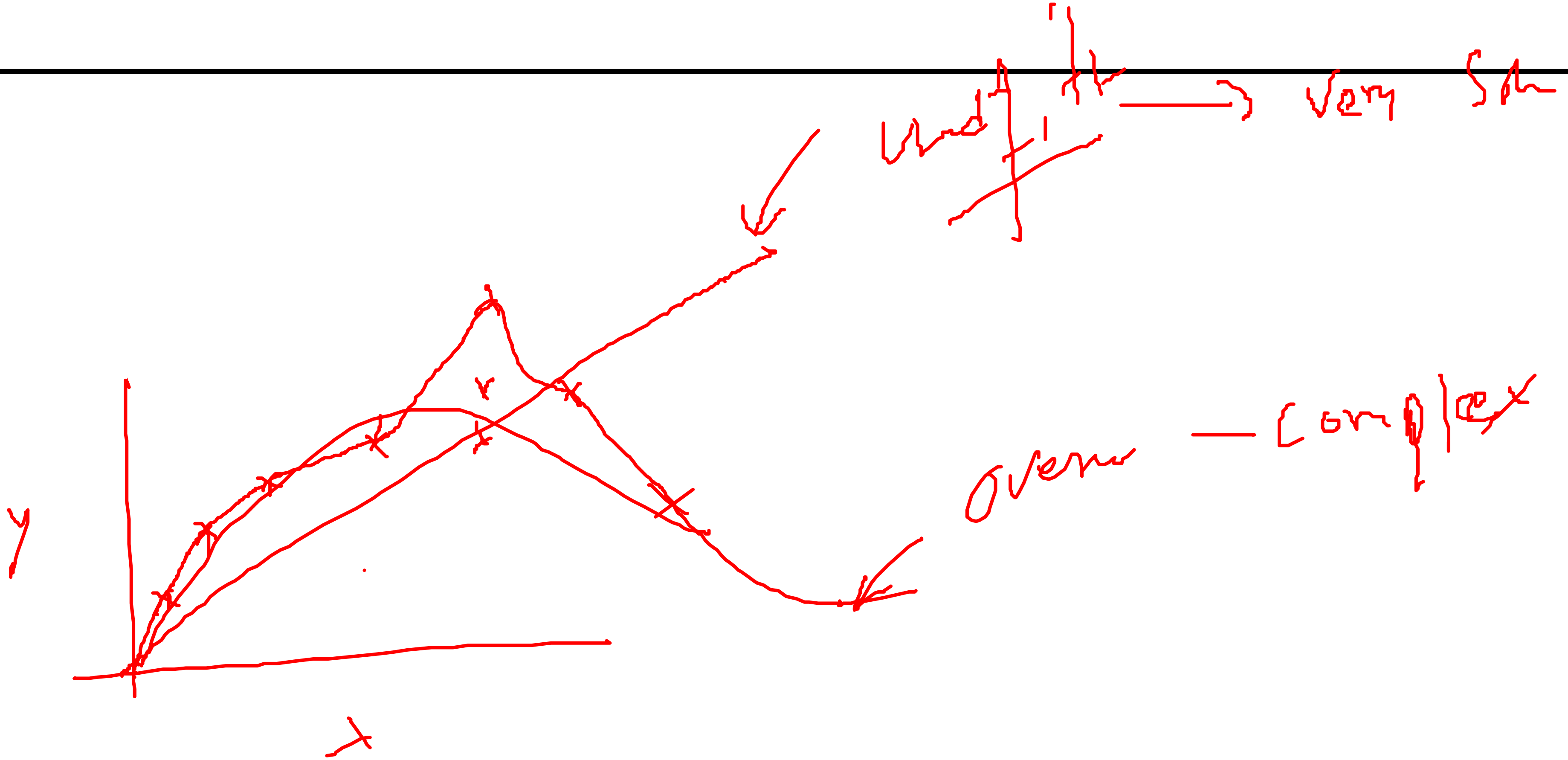
- Estimate/predict

  $y' = f(X)$

- Such that:
  - *$\min E[L(y'_u, y_u)]$*
- for unseen data X_u
- *$L(y'_u, y_u)$:* Estimated risk / Loss

Training data : Data used to find optimise parameters

Testing data: Unseen data, used to evaluate the model (function) performance


y
x
Underfit → very Sh
Overfit — complex

How does that work?

We split data *(X, y)* randomly to two sets

- Training set *(Xt, yt)*
- Testing set *(Xs, ys)*

- *How much to split?*
- *Does this work?*
- *What can go wrong?*

- *Other strategies:*
- *.. K-Fold cross validation, LOOV,*
- *.. (train, validation/dev, test)*

Regression & Classification

Question: Can we treat classification problem as regression??

Supervised Learning: Regression & Classification

Regression:

- - target (y) is continues variable y ∈ ***R***
- - Example
  - Target: Housing price, blood sugar, temperature,

Classification:

- - target (y) is categorical, or limited set of integers y ∈ [0, 1]
- - Examples:
  - Email-spam or not, cat & dog image, handwriting digits classification, Cancer or not etc

Performance Measure & Loss function

Regression:

Why so many measures?

- Mean Square Error, $L(y', y) = E[\,|y'-y|^2\,]$
- Mean Absolute Error $L(y', y) = E[\,|y'-y|\,]$
- R^2, Pearson correlation

Classification:

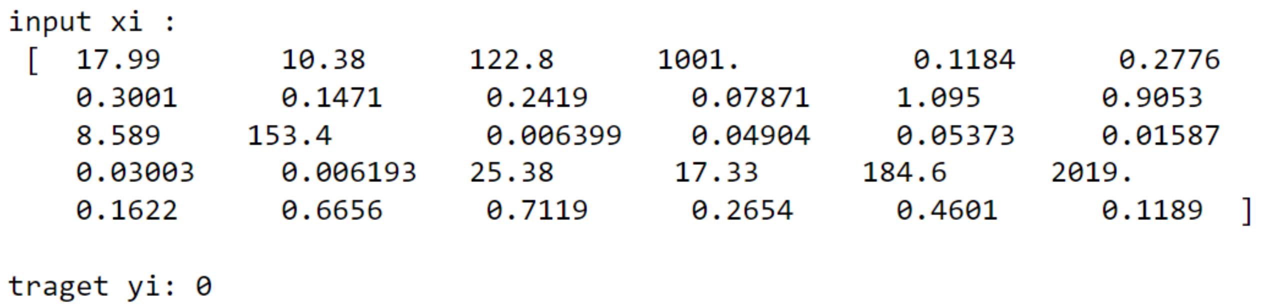
- Accuracy, $E[(y'==y)]$
- F1-score, precision, recall, AUC
- Diagnosis: Confusion Matrix, ROC, misclassification, learning curve (NN)
- Loss function: Cross-entropy, Hinge loss, logistic loss etc

Classification Problem

Example: Binary Classification

Breast cancer Classification:

- target y ∈ [0, 1]
- 30 features: X:



```
input xi :
 [  17.99        10.38       122.8       1001.          0.1184       0.2776
    0.3001       0.1471      0.2419      0.07871      1.095        0.9053
    8.589      153.4         0.006399    0.04904      0.05373      0.01587
    0.03003      0.006193   25.38       17.33       184.6        2019.
    0.1622       0.6656      0.7119      0.2654       0.4601       0.1189   ]

traget yi: 0
```

- Can we apply linear regression model?

- Kind-of, yes* and No*, until it is binary

Classification Problem

Binary Classification:

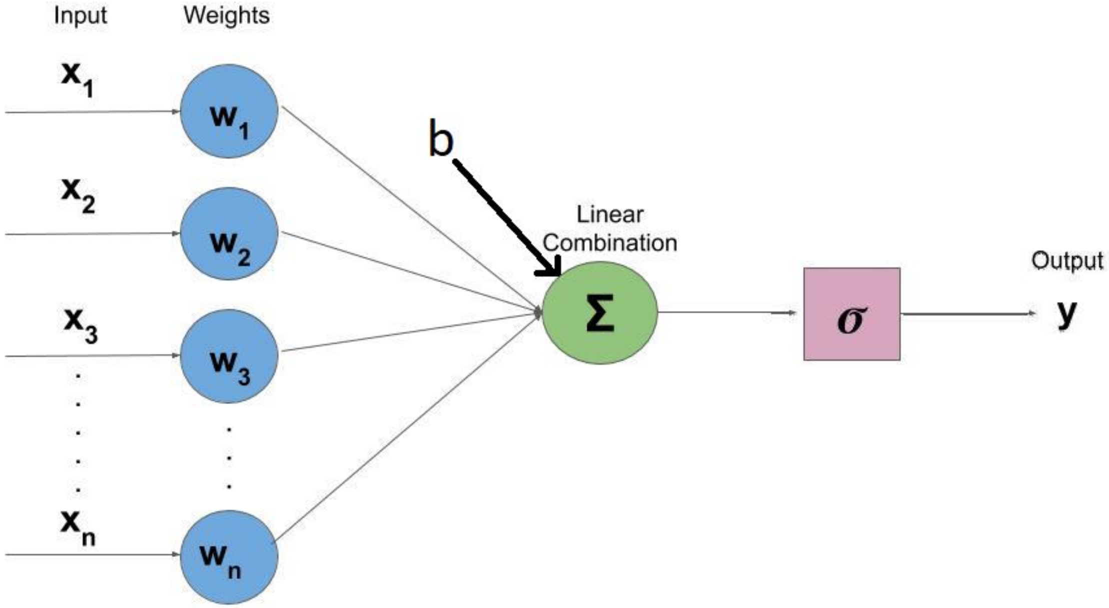
$h = b + w_1x_1 + w_2x_2 \ldots w_nx_n$

$y = 1 \quad if \; h>0 \; else \; 0$

How far h should be from 0?

As far as possible

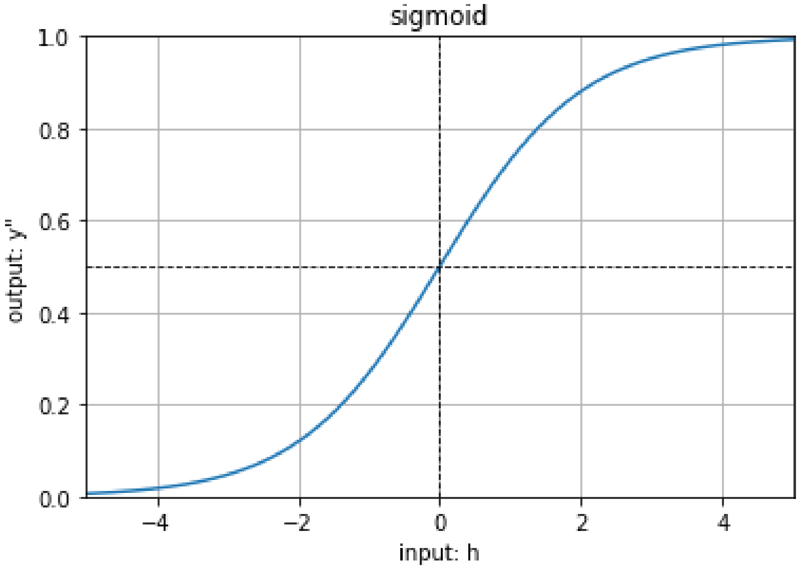
$y_p = sigmoid(h)$

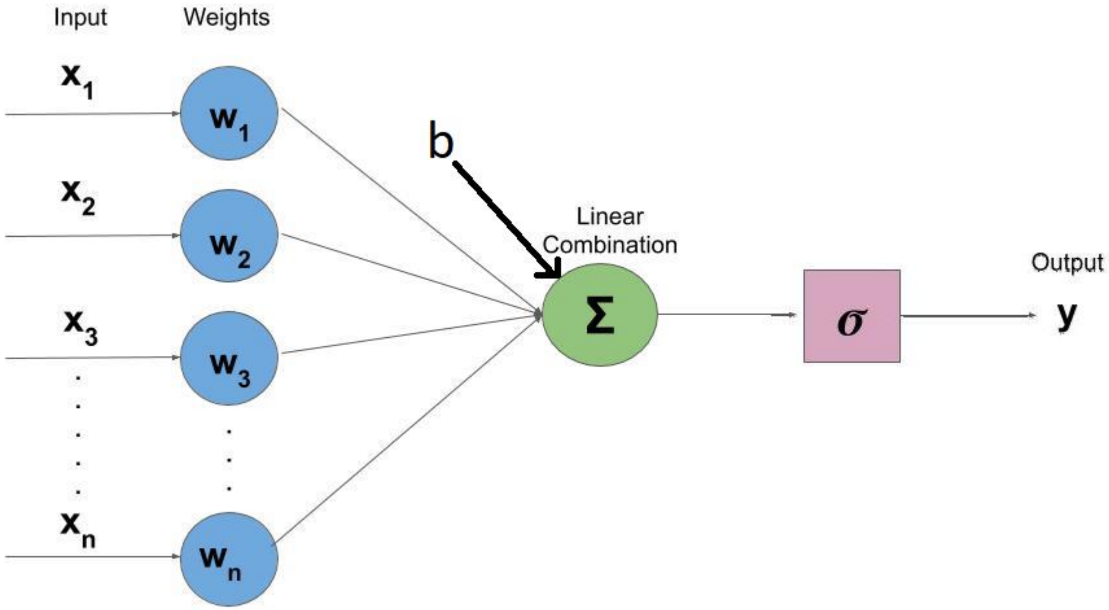


Classification Problem

Binary Classification:

yp = sigmoid(h) = 1/ (1 + exp(-h))





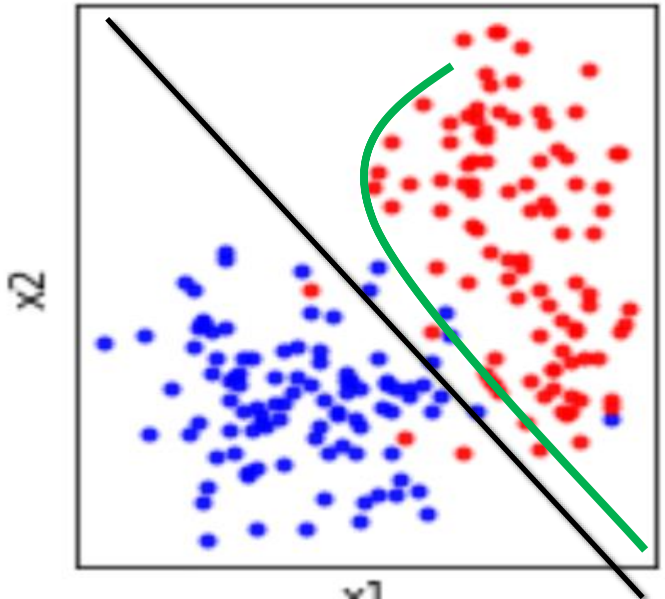
Loss $L(y,y_p) = \ ylog(y_p) + (1\text{-}y)log(1\text{-}\ y_p)$

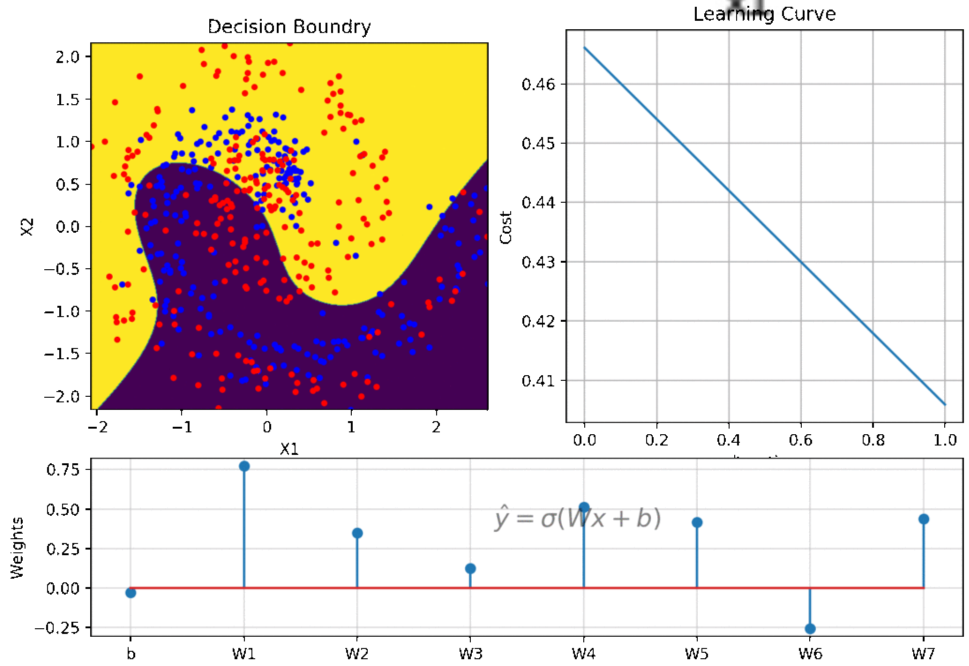
Classification Problem

Binary Classification:

$$h = b + w_1x_1 + w_2x_2$$

We need, ***h***, a plane, in hyperdimensional space that separates two classes

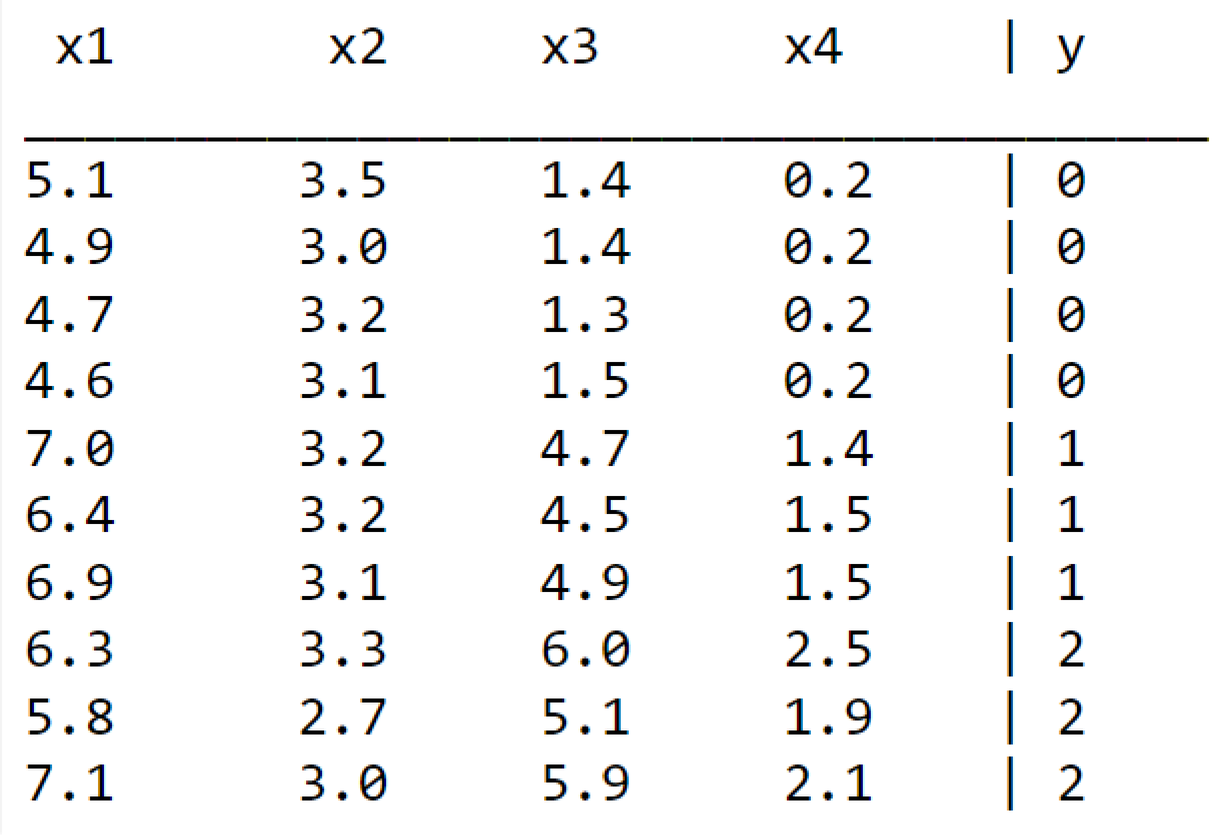




Classification Problem

Example: Iris Dataset

- Features:
  - x1: sepal length (cm)
  - x2: sepal width (cm)
  - x3: petal length (cm)
  - x4: petal width (cm)
- Three classes:
  - Setosa, Versicolor, Virginica
  - 0 1 2

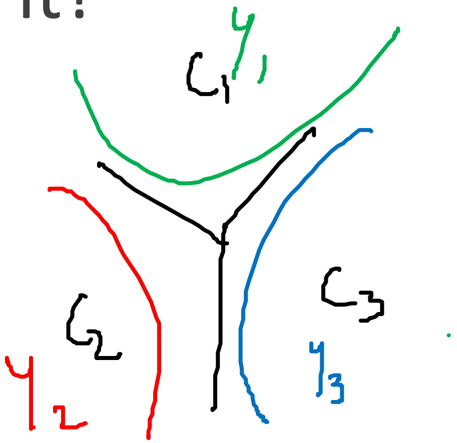


```
 x1          x2       x3        x4       | y
_________________________________________________
5.1         3.5      1.4       0.2      | 0
4.9         3.0      1.4       0.2      | 0
4.7         3.2      1.3       0.2      | 0
4.6         3.1      1.5       0.2      | 0
7.0         3.2      4.7       1.4      | 1
6.4         3.2      4.5       1.5      | 1
6.9         3.1      4.9       1.5      | 1
6.3         3.3      6.0       2.5      | 2
5.8         2.7      5.1       1.9      | 2
7.1         3.0      5.9       2.1      | 2
```

Classification Problem

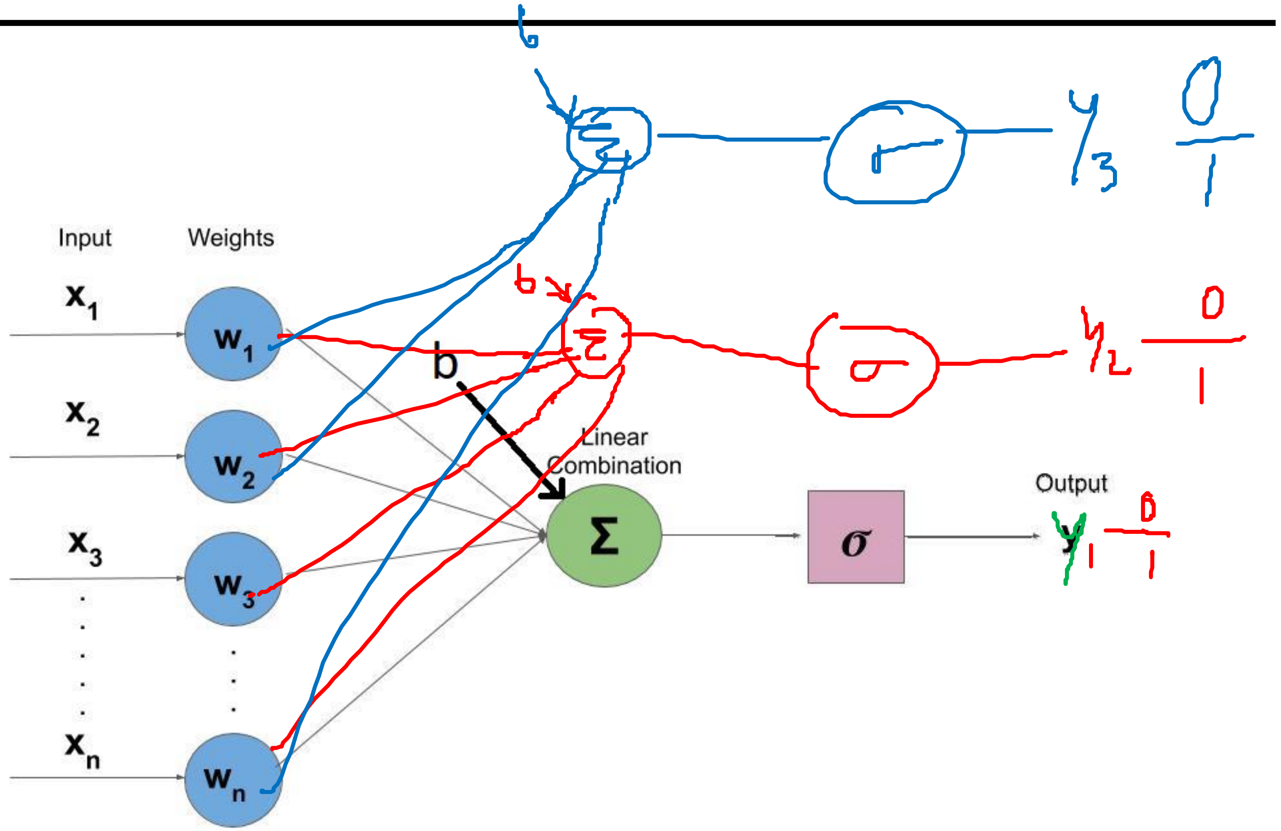
Iris dataset – Multi-class

- How do we do it?



- one vs all

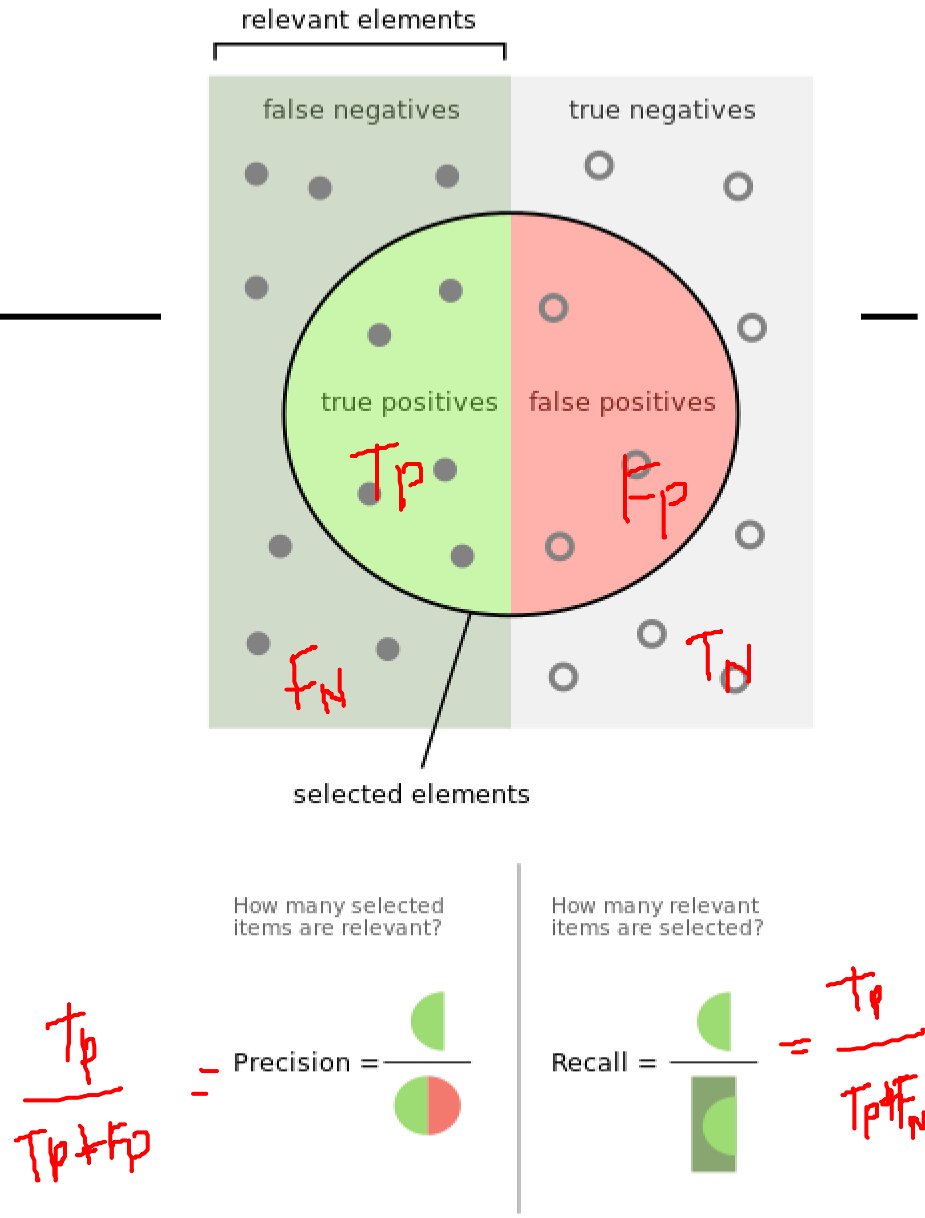
	y_1	y_2	y_3
$y=0$	1	0	0
$y=1$	0	1	0
$y=2$	0	0	1



Performance Metrics

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$



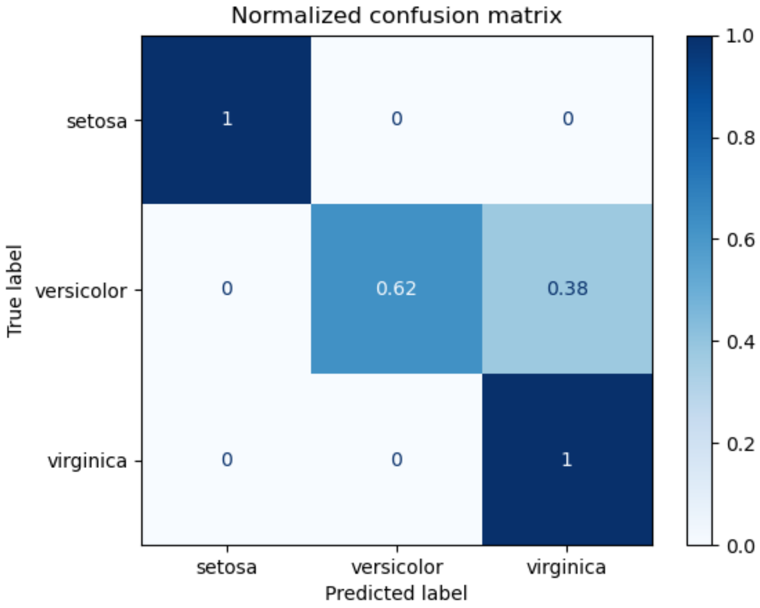
$\frac{Tp}{Tp+Fp}$ = Precision

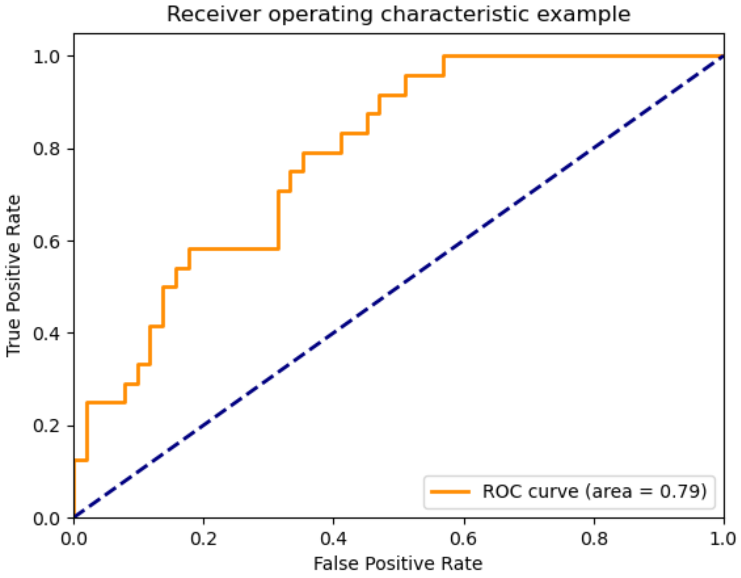
Recall = $\frac{tp}{Tp+FN}$

*Wikipedia

Performance Metrics

Confusion matric is not always same as TP/FP table





Normalising Features

Normalising :

- When?
- Why?
- How?

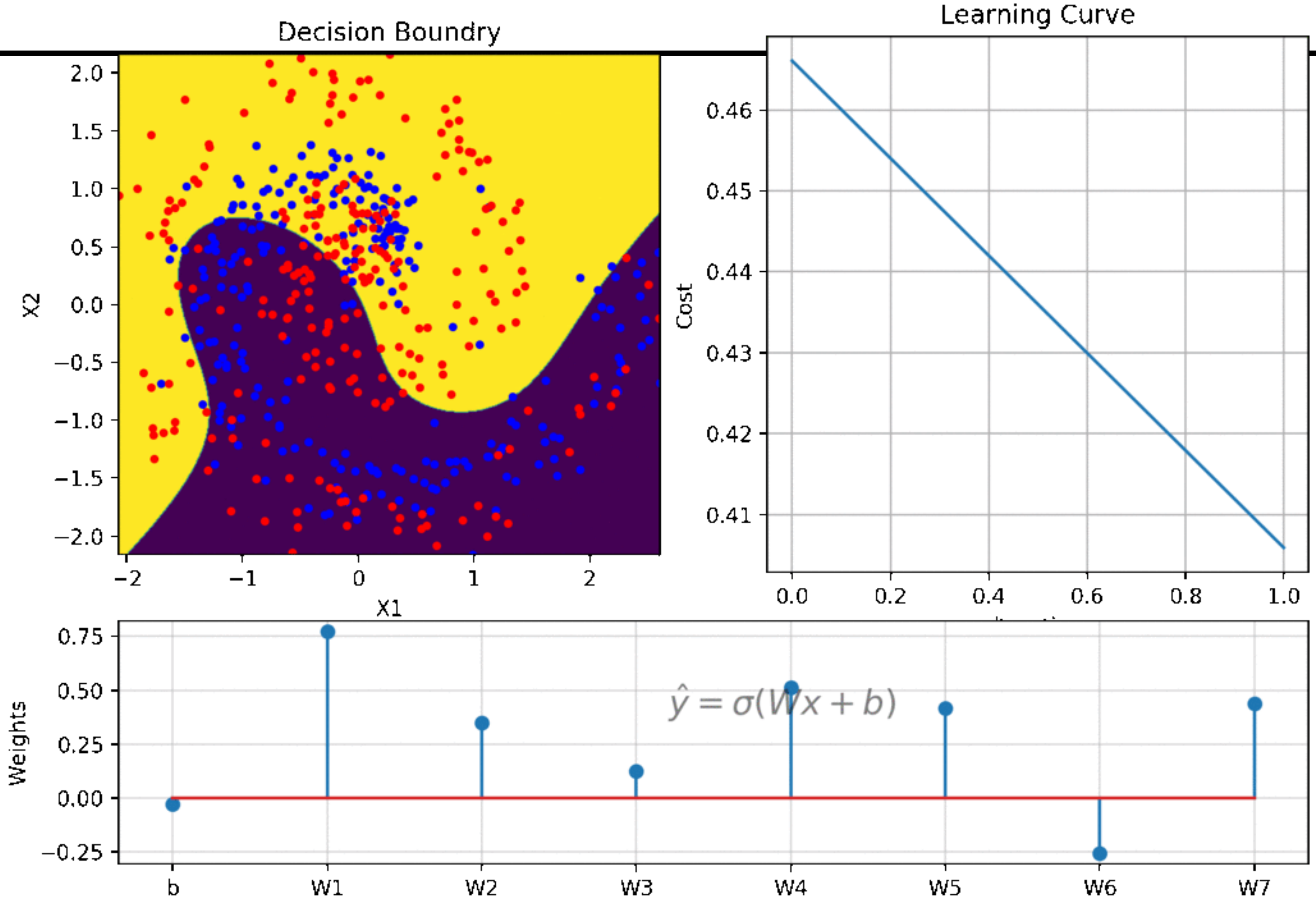
- Mean cantered: Zero-mean, remove DC
- Scale Standard Deviation
- 0-1 normlisation

Machine Learning Models

ML Models:

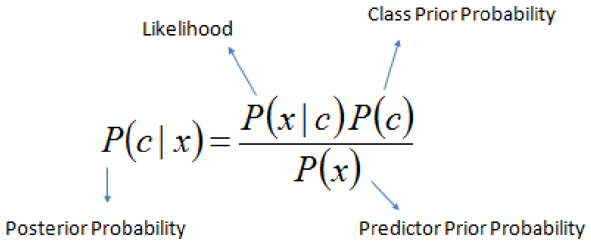
- Linear regression, Logistic Regression
- Support Vector Machine
- Decision Tree
- K-Nearest Neighbourhood (KNN)
- Naïve Bayes
- Ensemble Approach:
  - Random Forest
  - Gradient Booster
  - ExtraTree, AdaBoost

Logistic Regression



$\hat{y} = \sigma(Wx + b)$

Naïve Bayes



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

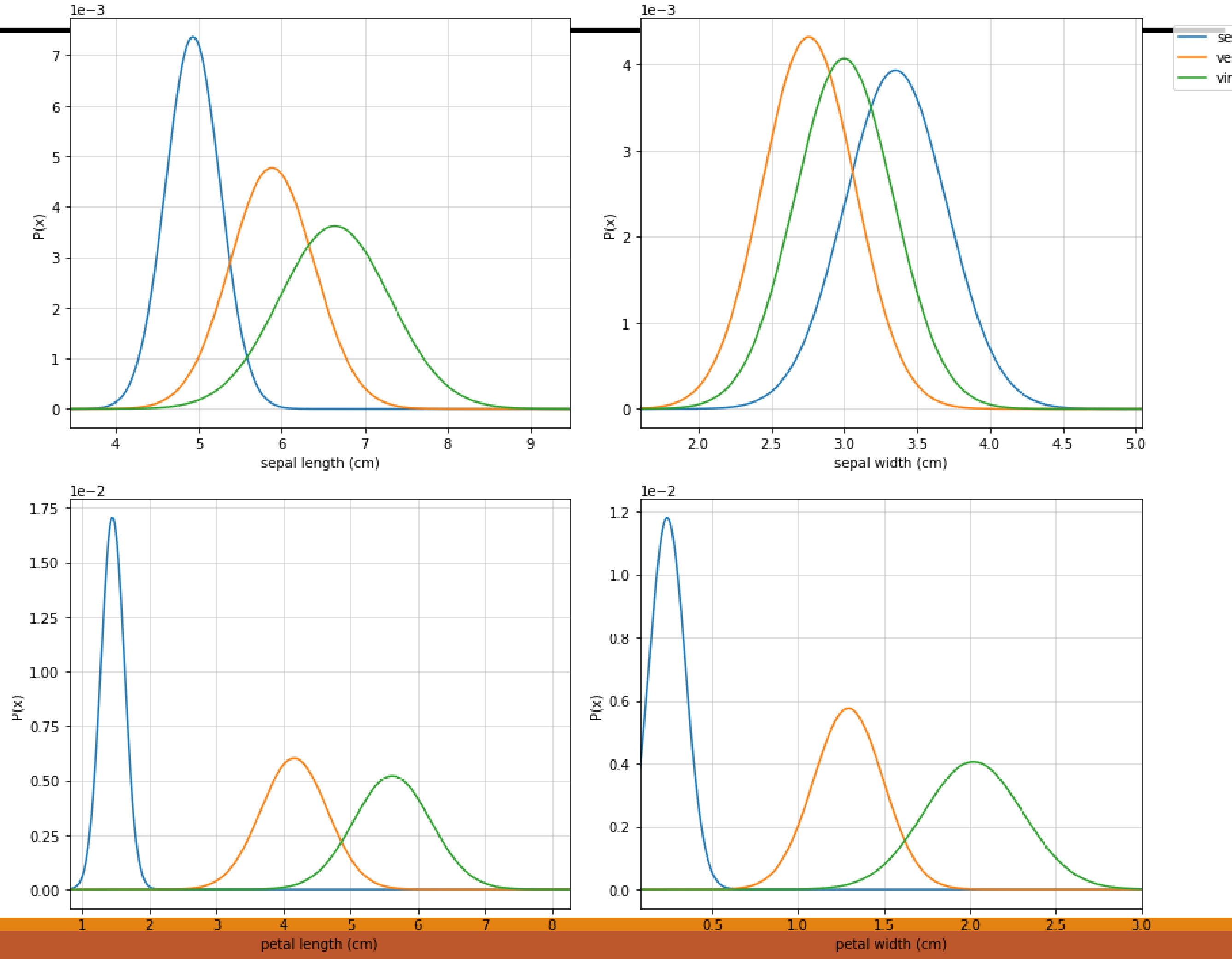
Likelihood

Class Prior Probability

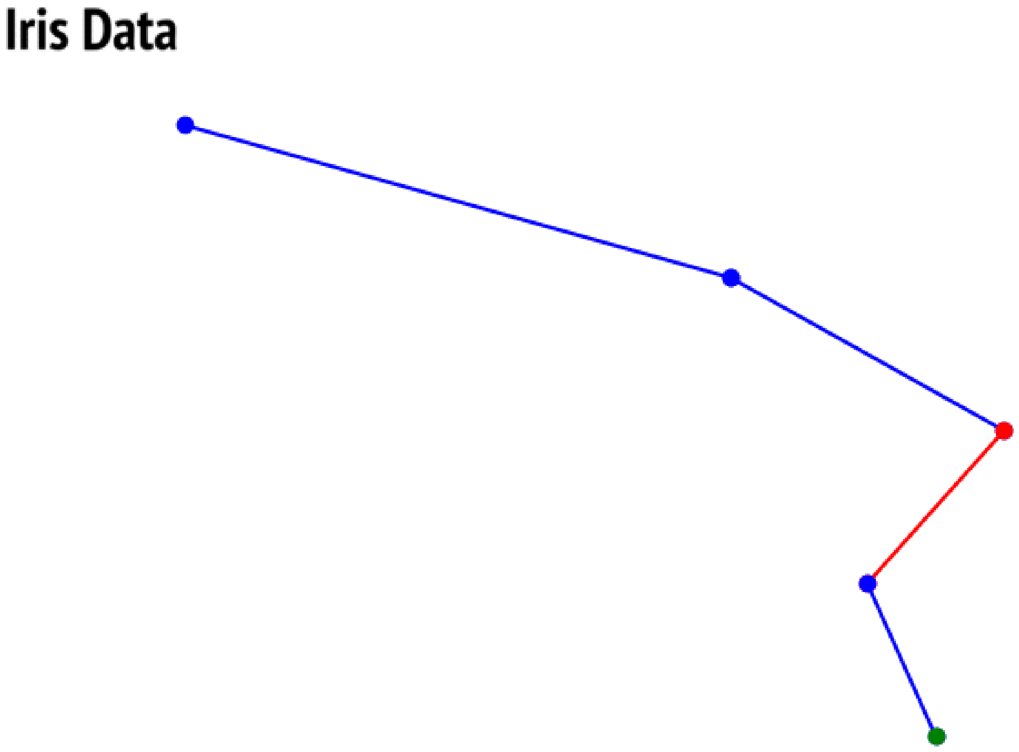
Posterior Probability

Predictor Prior Probability

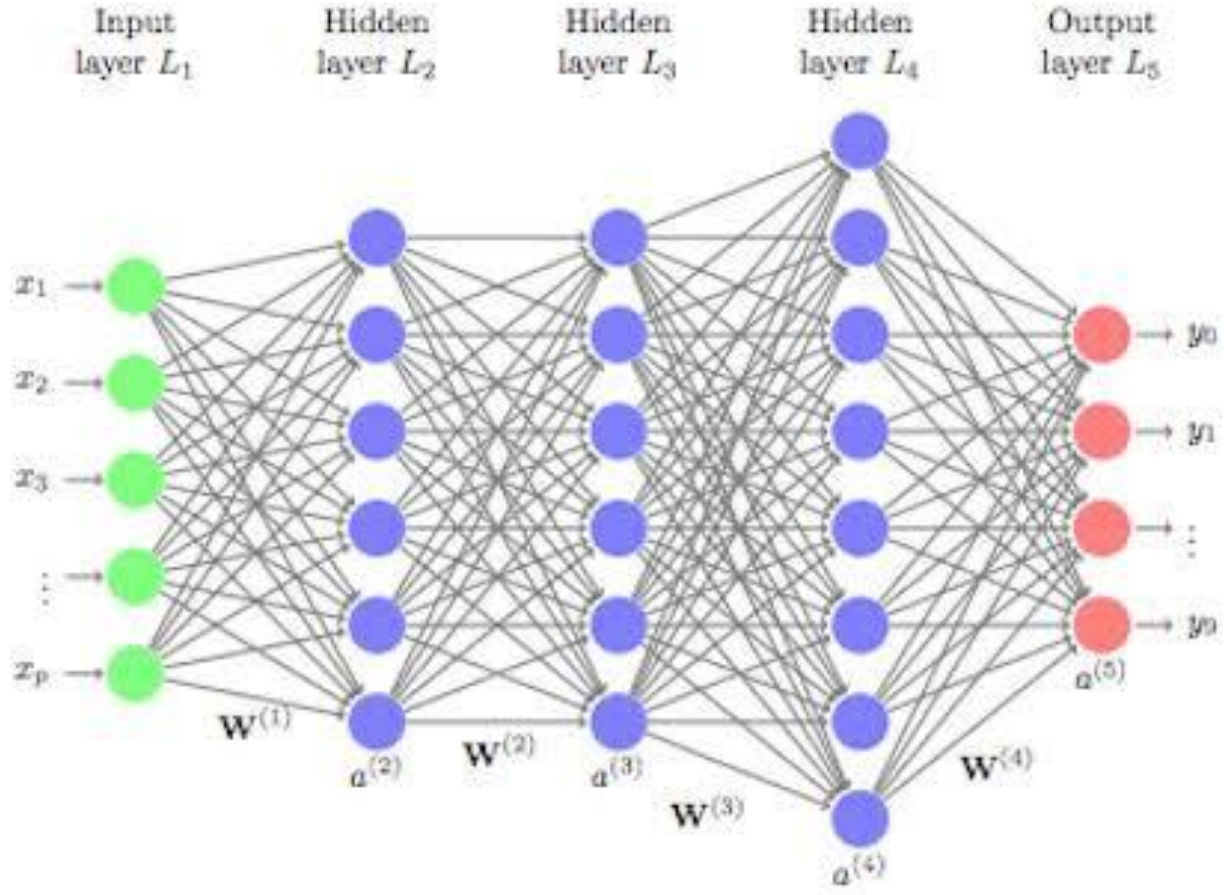
$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$



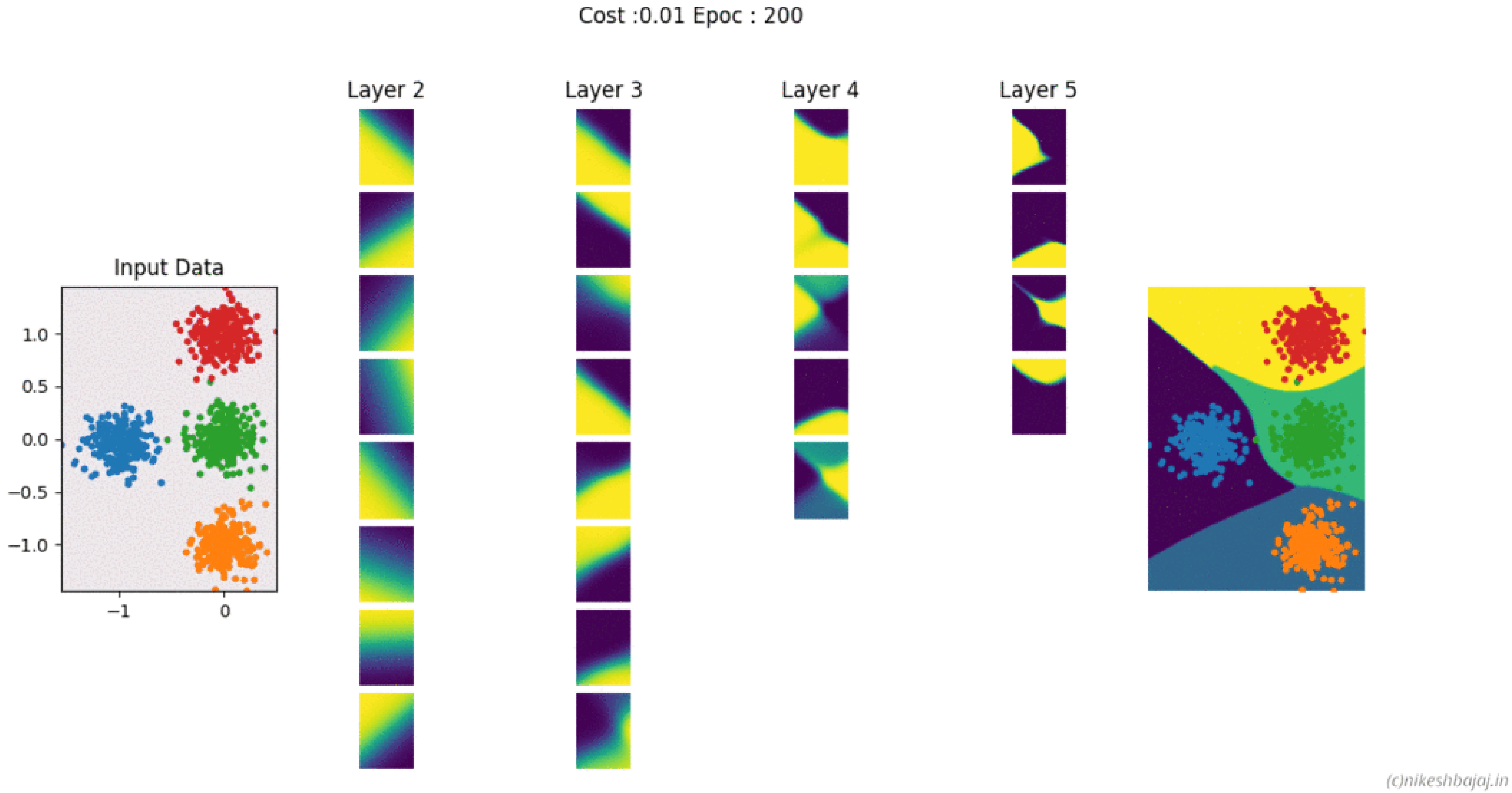
Decision Trees



Neural Network - Idea



Neural Network



Deeplearning

Deeplearning

- The deep* learning refer to a family of models based on Neural Networks. It has following aspects w.r.t conventional ML models
  - End-to-end learning
  - Complex relationship of input-target
  - Proven to solve many problems, which were not easy with conventional ML
  - Large number of parameters, heavy,
  - Not as easy to explain as conv. ML models

Deeplearning

Neural Network:

- Fully Connected Neural Network (MLP)
- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN) – LSTM, GRU
- Generative Adversarial Networks (GANs)

Example

Handwritten Digit Recognition :

Any
Questions ?