Machine Learning

Lecture 2

Aman Agarwal
Aditya Mishra

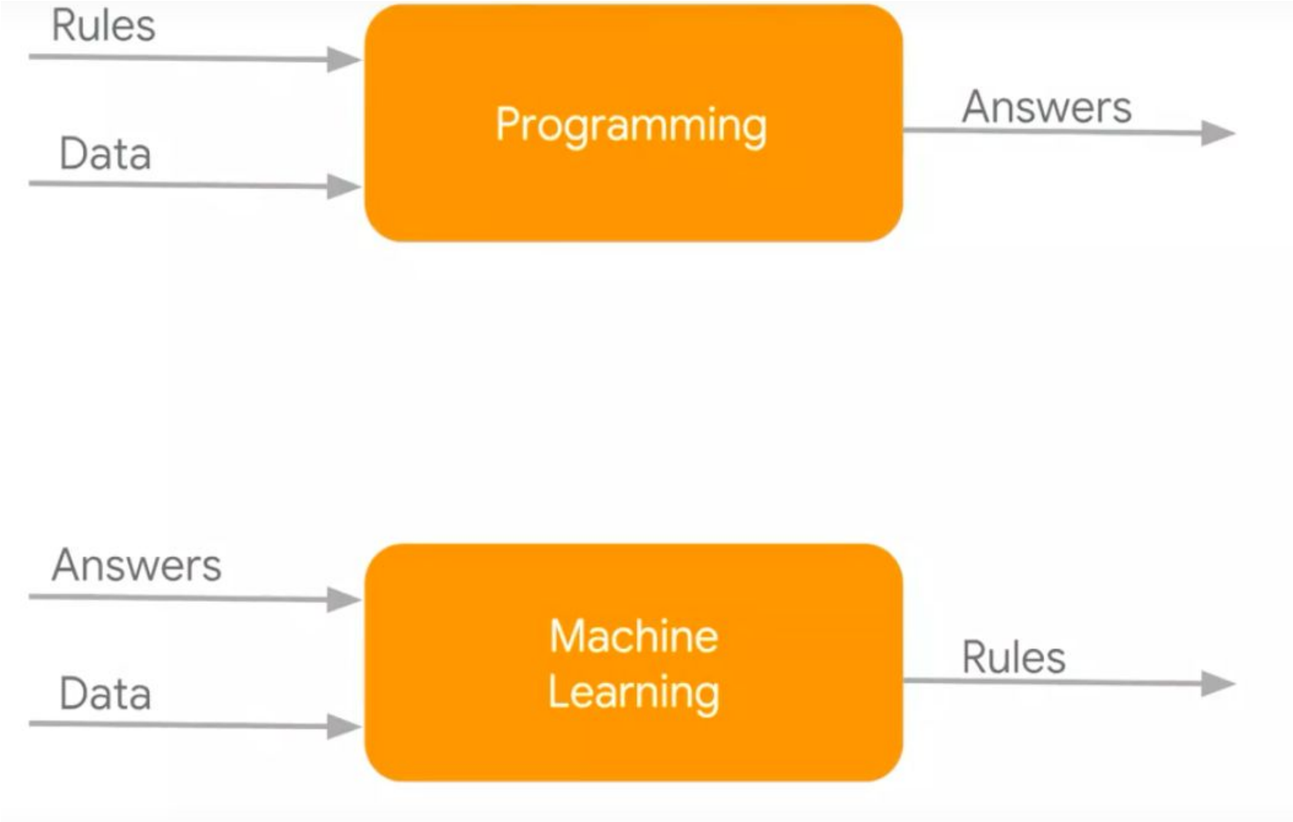
NIRMA UNIVERSITY

Introduction[1]

"It is a field of study that gives a computer the ability to learn without being explicitly programmed."

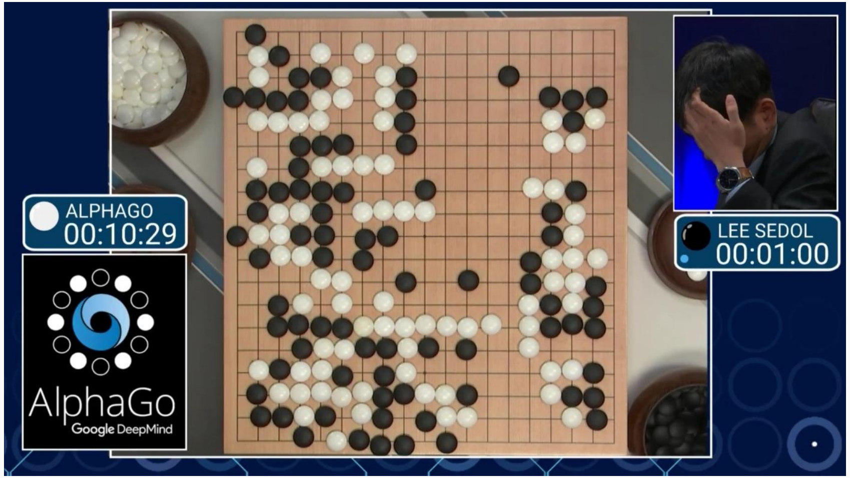
- Arthur Samuel

Machine Learning v/s Conventional Learning



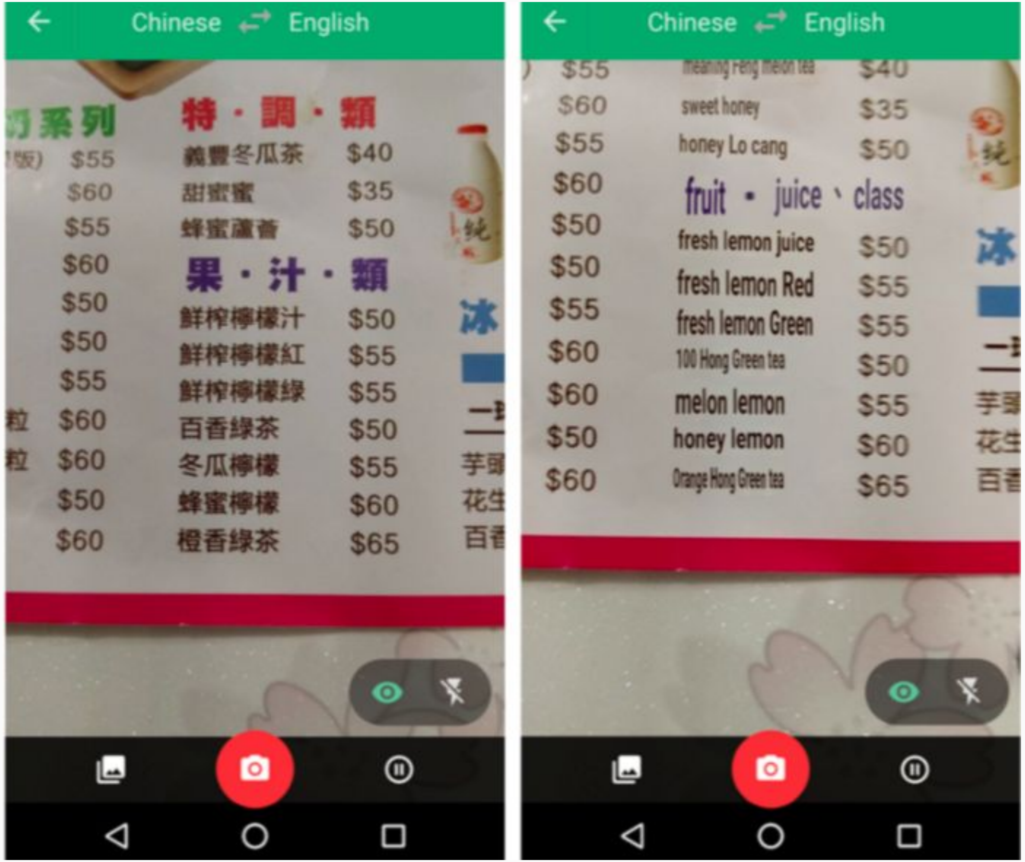
Motivation

- IBM's DeepBlue defeating world class chess champion Garry Kasparov in 1997.
- AlphaGo defeating one of the best human players at Go in 2016. The game has more than 10^{170} possible moves (there are only 10^{80} atoms in the universe).
- OpenAI defeating world's best DOTA 2 players.



- Point your camera at the menu during your next trip to Taiwan and the restaurant’s selections will magically appear in English via the Google Translate app.

- Google’s AI making appointment on your behalf.



Google Translate overlaying English translations on a drink menu in real time using convolutional neural networks.

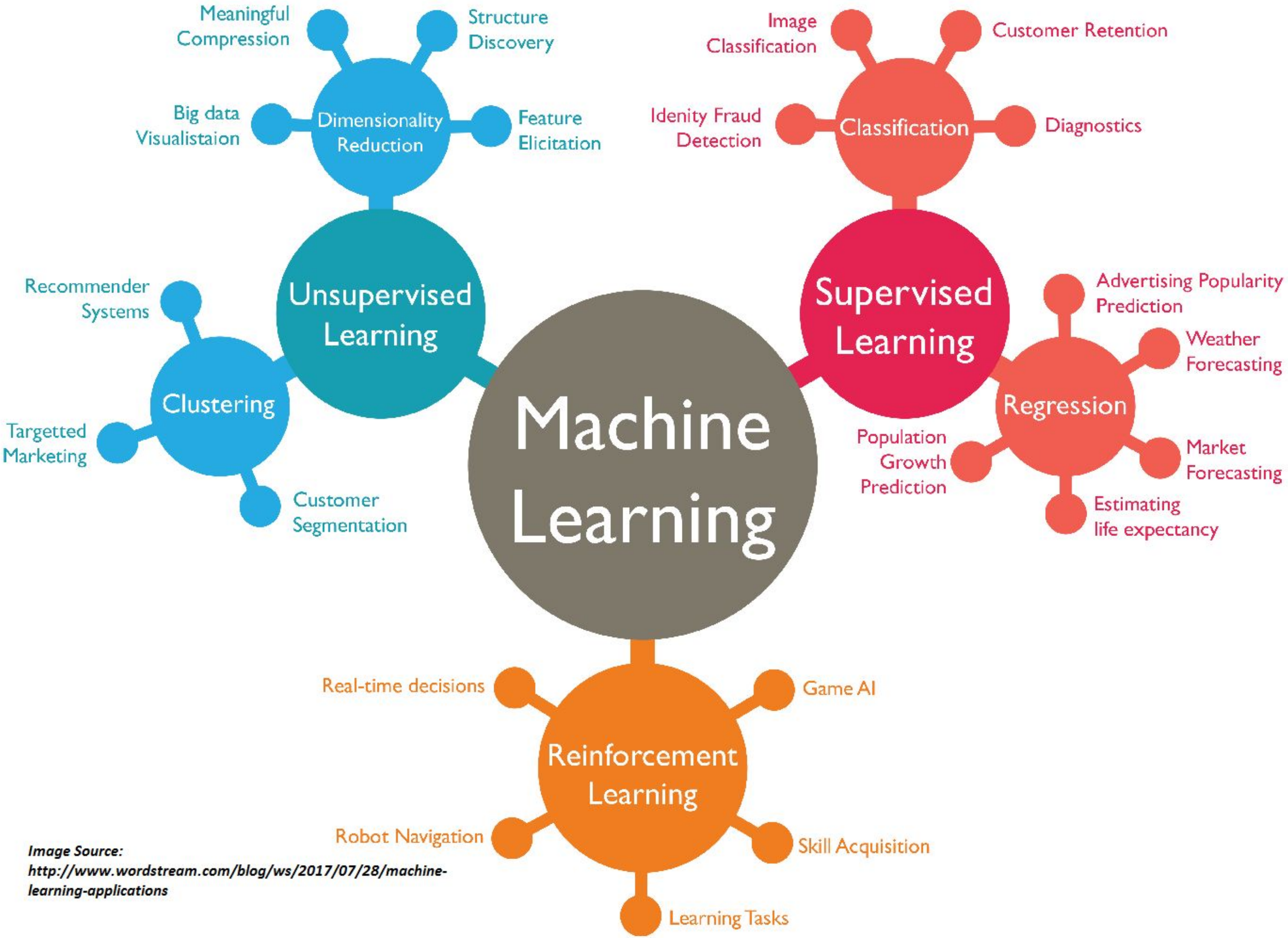
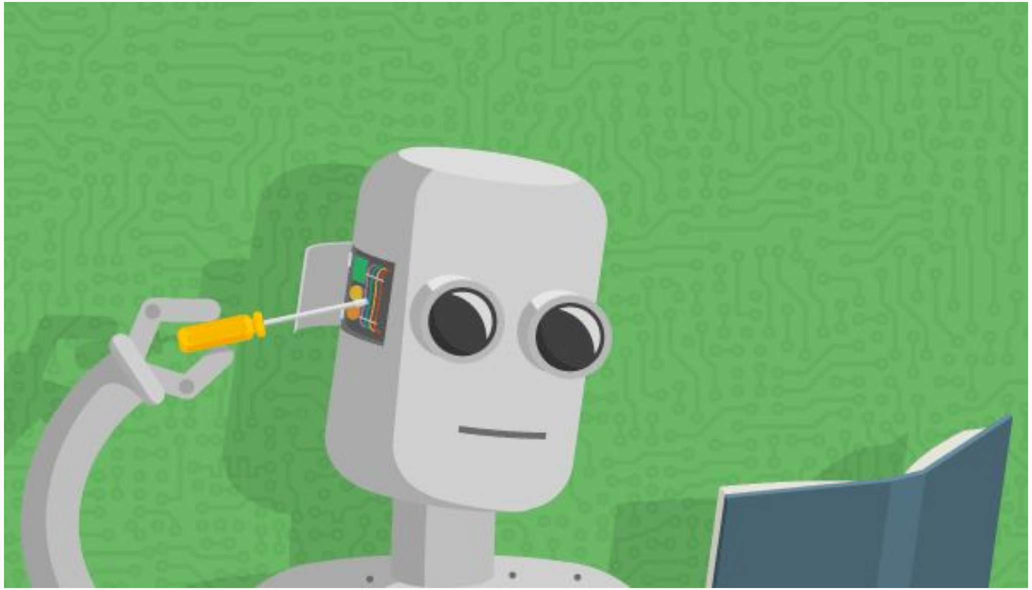


Image Source: http://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications

Lecture Overview

- Supervised Learning
  - Regression
  - Classification
  - Overfitting / Underfitting
  - Cost Function
- Unsupervised Learning
  - Clustering
  - Dimensionality Reduction
  - Feature Extraction



Supervised Learning

- Part I (Regression)
  - Linear Regression
  - Learning Rate
  - Overfitting v/s Underfitting
  - Gradient Descent
- Part II (Classification)
  - Logistic Regression
  - Cost Function
  - Support Vector Machine
  - Neural Network
- Part III (Non-parametric)
  - Decision Tree
  - Naive Bayes

Supervised Learning

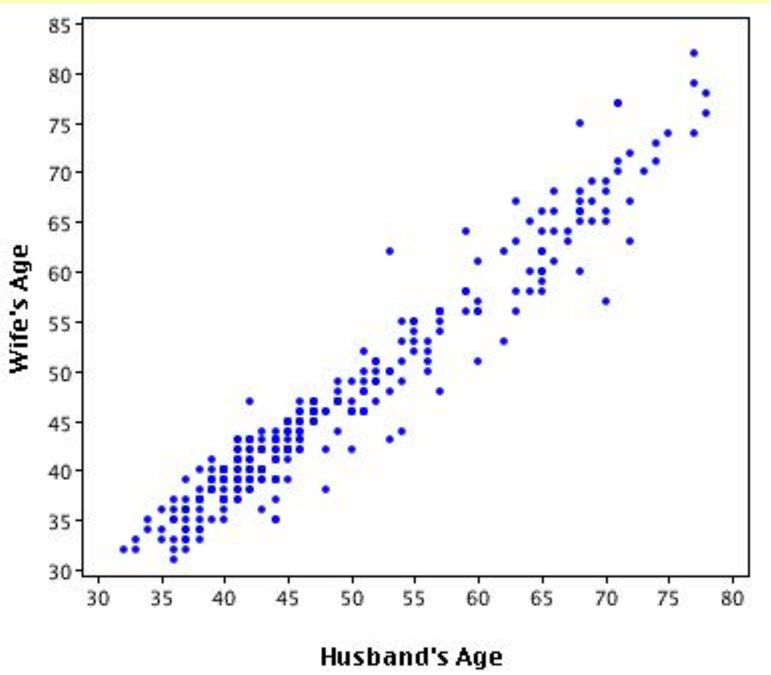
- Supervised learning is done using a ground truth i.e. we have prior knowledge of what the output values for our samples should be.
- Goal of supervised learning is to learn a function that, given a sample of data, best approximates the relationship between input and output observable in the data.
- Note that "correct" output is determined entirely from the training data, so while we do have a ground truth that our model will assume is true, it is not to say that data labels are always correct in real-world situations.
- Noisy, or incorrect, data labels will clearly reduce the effectiveness of the model.

Observation #	Years of Higher Education (X)	Income (Y)
1	4	$80,000
2	5	$91,500
3	0	$42,000
4	2	$55,000
...	...	...
N	6	$100,000

Linear Regression

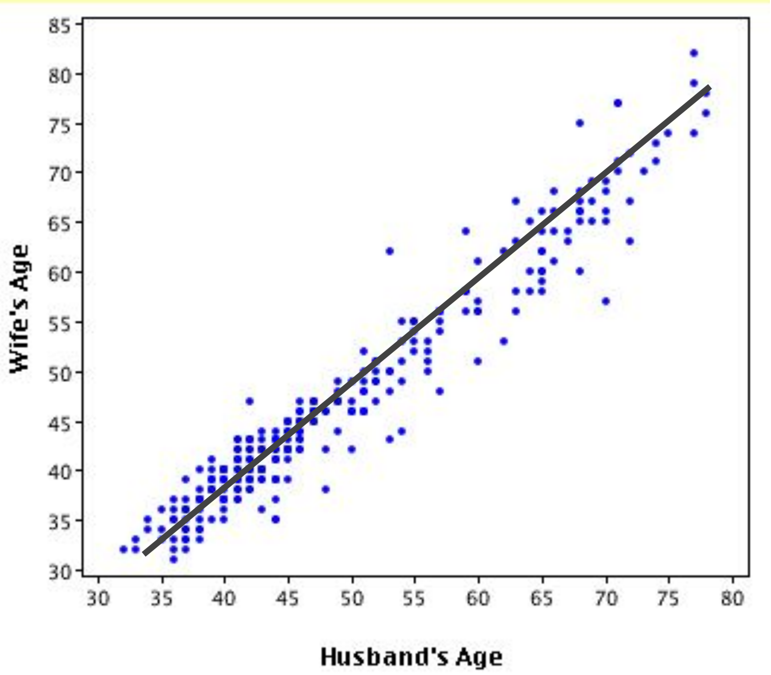
Linear Regression

- Linear regression is useful for finding relationship between two continuous variables.
- One is the predictor or independent variable and other is response or dependent variable.
- The core idea is to obtain a line that best fits the data.



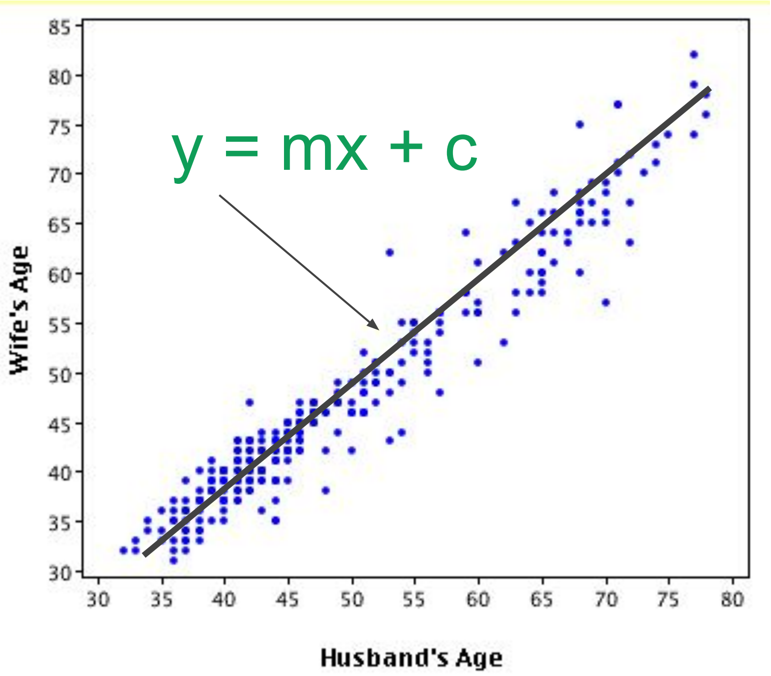
Linear Regression

- Linear regression is useful for finding relationship between two continuous variables.
- One is the predictor or independent variable and other is response or dependent variable.
- The core idea is to obtain a line that best fits the data.



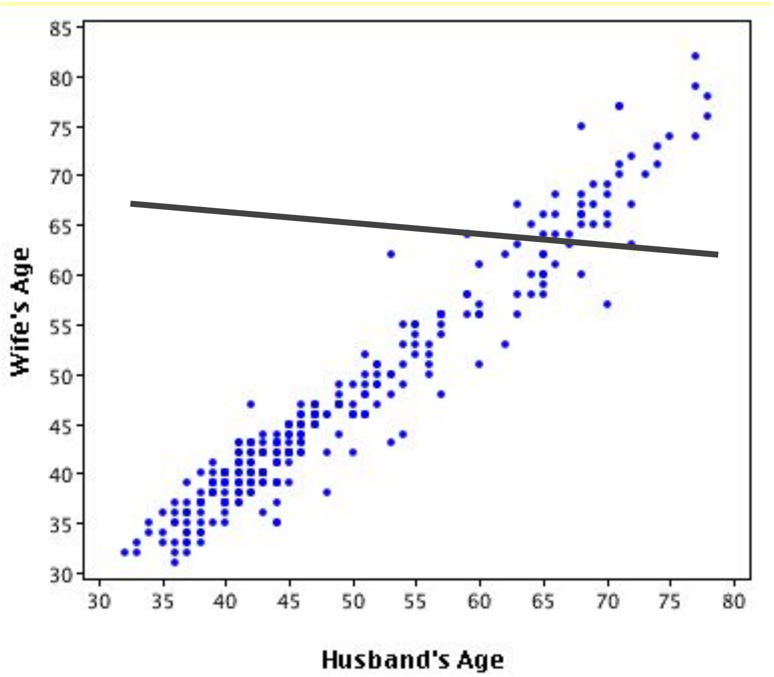
Linear Regression

- Linear regression is useful for finding relationship between two continuous variables.
- One is the predictor or independent variable and other is response or dependent variable.
- The core idea is to obtain a line that best fits the data.



Goal

- The values of `m` and `c` must be chosen so that they minimize the error.
- If sum of squared error is taken as a metric to evaluate the model, then goal to obtain a line that best reduces the error.



Steps

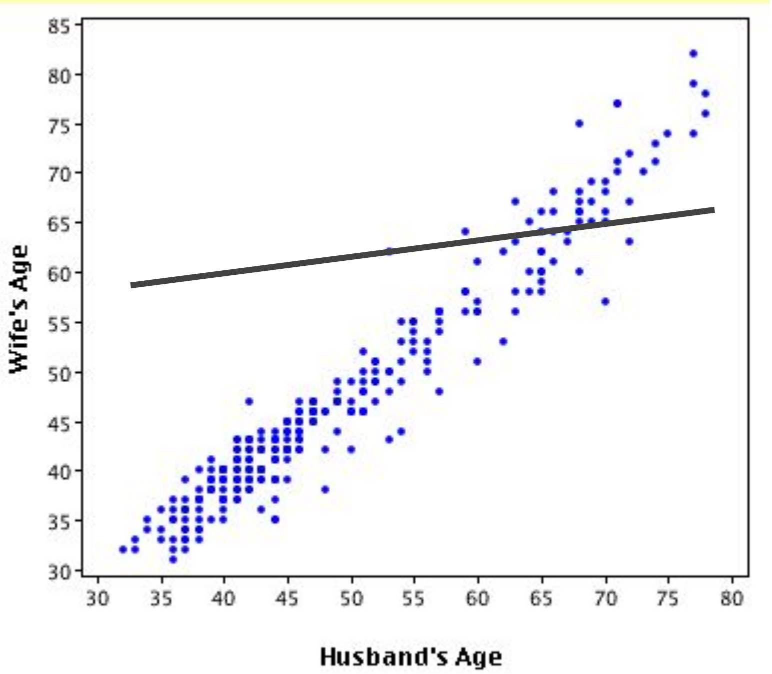
- Select a random value of `m` and `c`.
- Pick an example, and compute the value of `y'`.
- Take the difference (error) of Actual `y` and computed `y'`.
- Update the values of `m` and `c`.
- Repeat till error becomes acceptable.

Goal

- The values of `m` and `c` must be chosen so that they minimize the error.
- If sum of squared error is taken as a metric to evaluate the model, then goal to obtain a line that best reduces the error.

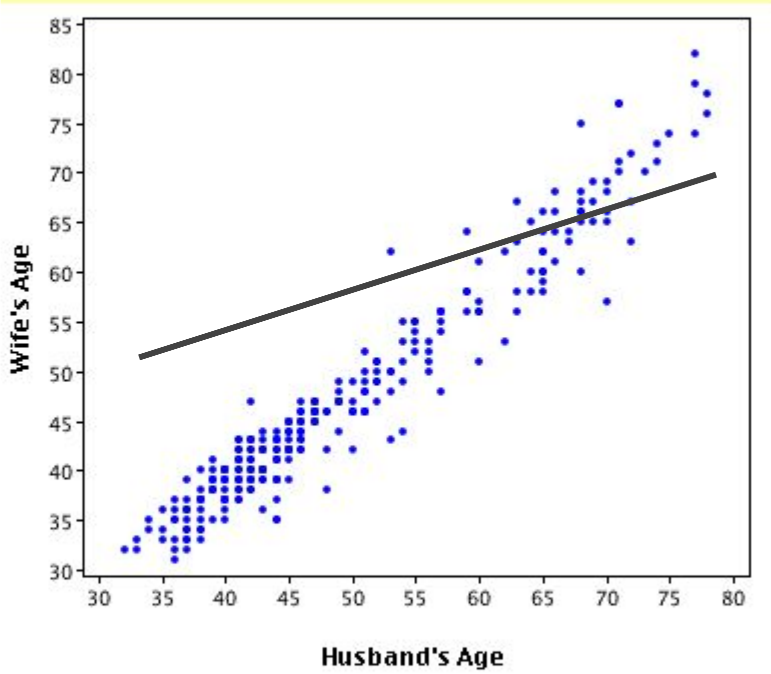
Steps

- Select a random value of `m` and `c`.
- Pick an example, and compute the value of `y'`.
- Take the difference (error) of Actual `y` and computed `y'`.
- Update the values of `m` and `c`.
- Repeat till error becomes acceptable.



Goal

- The values of `m` and `c` must be chosen so that they minimize the error.
- If sum of squared error is taken as a metric to evaluate the model, then goal to obtain a line that best reduces the error.

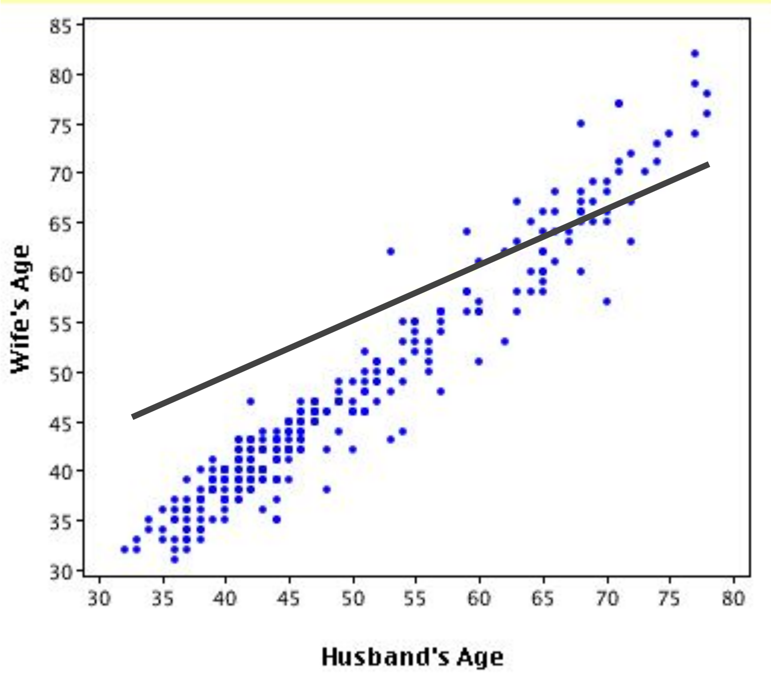


Steps

- Select a random value of `m` and `c`.
- Pick an example, and compute the value of `y'`.
- Take the difference (error) of Actual `y` and computed `y'`.
- Update the values of `m` and `c`.
- Repeat till error becomes acceptable.

Goal

- The values of `m` and `c` must be chosen so that they minimize the error.
- If sum of squared error is taken as a metric to evaluate the model, then goal to obtain a line that best reduces the error.

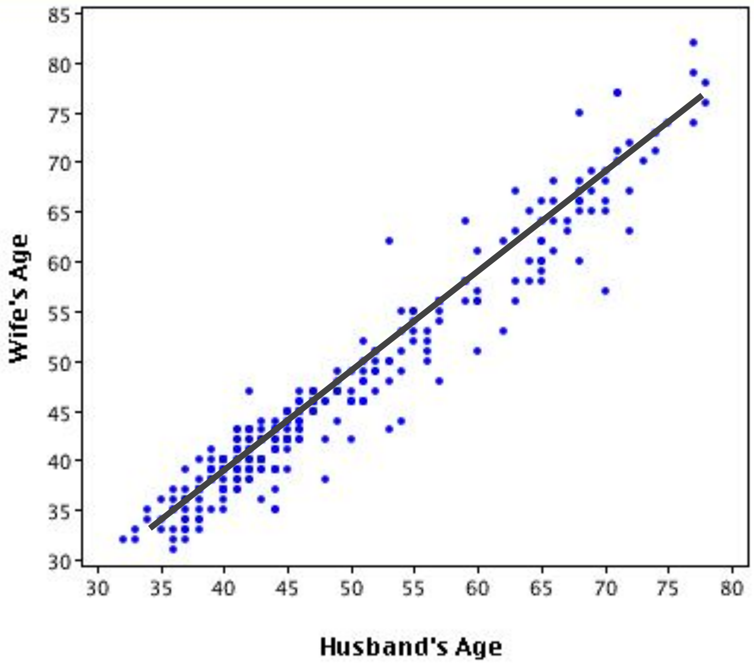


Steps

- Select a random value of `m` and `c`.
- Pick an example, and compute the value of `y'`.
- Take the difference (error) of Actual `y` and computed `y'`.
- Update the values of `m` and `c`.
- Repeat till error becomes acceptable.

Goal

- The values of m and c must be chosen so that they minimize the error.
- If sum of squared error is taken as a metric to evaluate the model, then goal to obtain a line that best reduces the error.



Steps

- Select a random value of m and c.
- Pick an example, and compute the value of y'.
- Take the difference (error) of Actual y and computed y'.
- Update the values of m and c.
- Repeat till error becomes acceptable.

Some Magic!

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

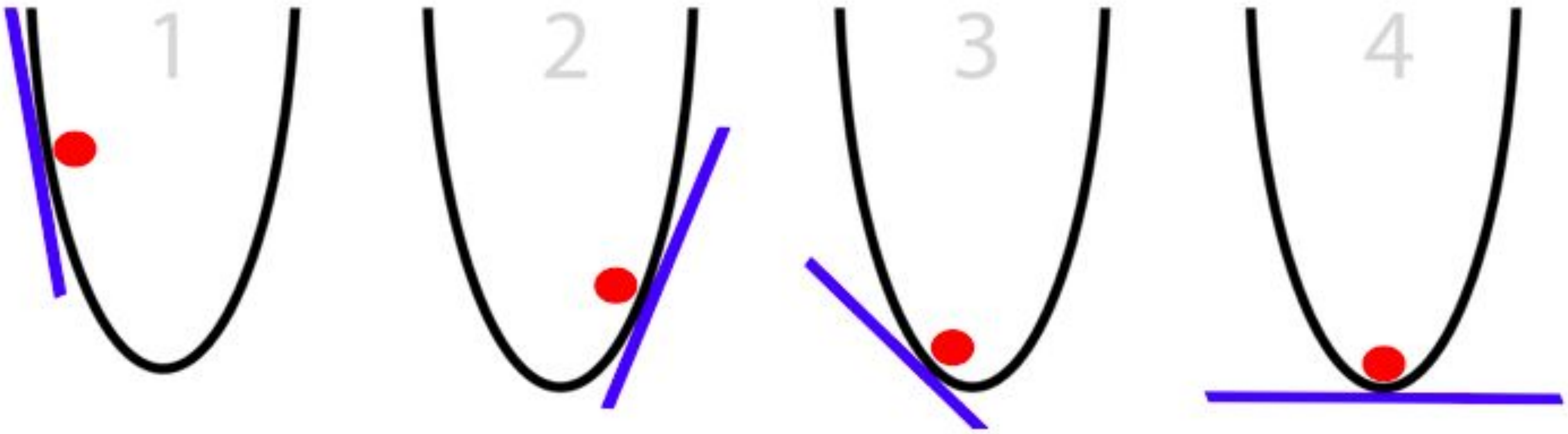
Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Gradient Descent[3]

Gradient Descent



More Magic!

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$(\text{for } j = 1 \text{ and } j = 0)$$

}

Linear Regression Model

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Summary

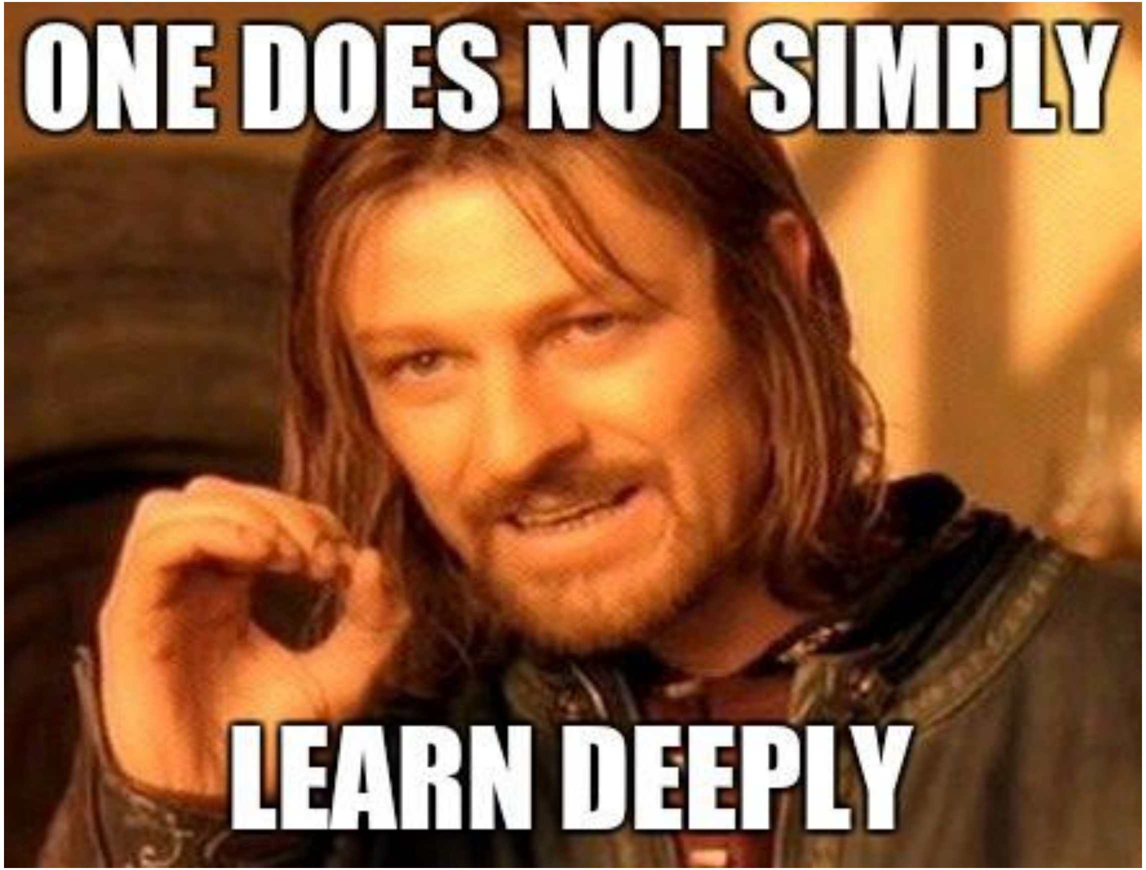
For every example x in the dataset

$$h_\theta(x) = \theta_0 + \theta_1 x$$

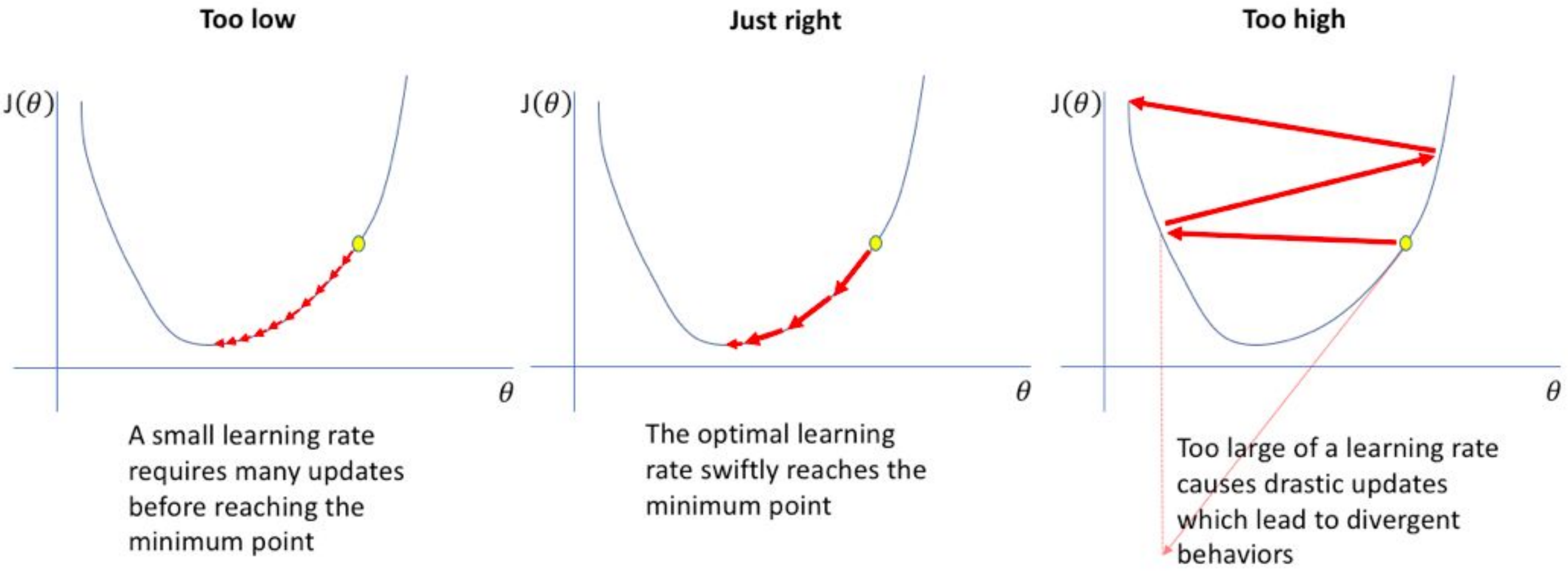
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\begin{aligned}
&\text{repeat } \{ \\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\
&\qquad (\text{for } j = 1 \text{ and } j = 0) \\
&\}
\end{aligned}$$

Learning Rate[4]



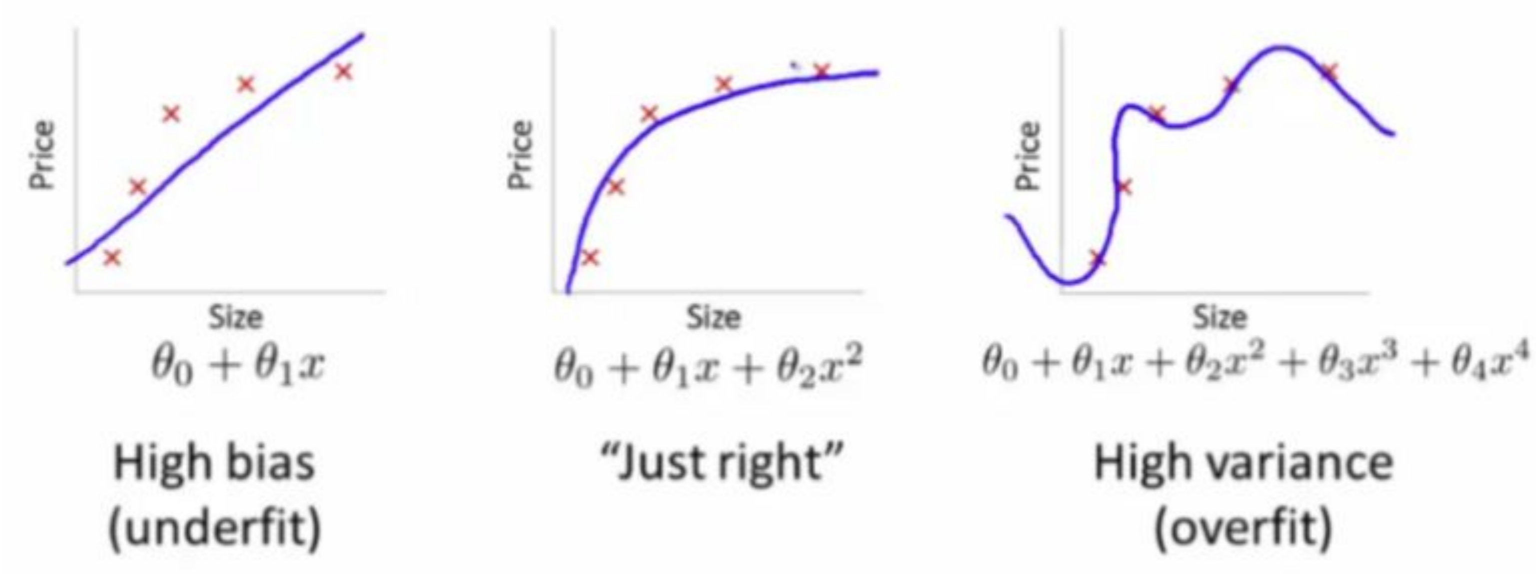
Learning Rate



Underfitting (bias) v/s Overfitting[2] (variance)

Underfitting v/s Overfitting

- **Overfitting**: Doing well on training set but bad on testing set.
- **Underfitting**: Doing bad on both training set as well as in the testing set.



Underfitting v/s Overfitting[2]

Overcoming Underfitting

- Increase number of features
- Train longer
- Change the model architecture

Overcoming Overfitting

- Reduce number of features
- Add Regularization
- Use more training data

Regularization

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

- Regularization adds a penalty for large theta coefficients that give too much explanatory power to any specific feature.
- The lambda coefficient of the regularization term in the cost function is a hyperparameter.
- Higher value of lambda will more harshly penalize the theta coefficients.

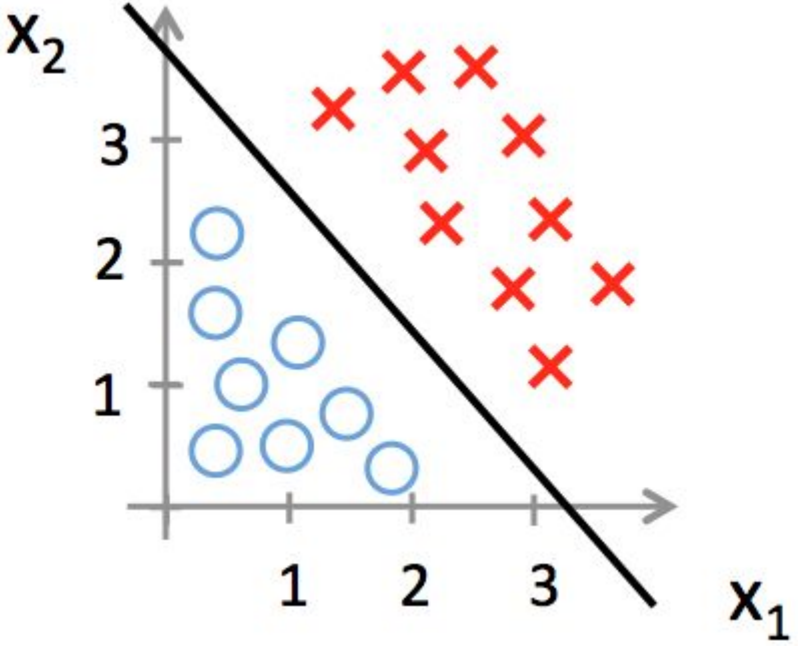
Logistic Regression (Classification)

Logistic Regression

- Logistic Regression is used for classification of data into different classes.

- Example:
  - Email - Spam / Not Spam
  - Tumor - Benign / Malignant
  - Image - Cat / Dog

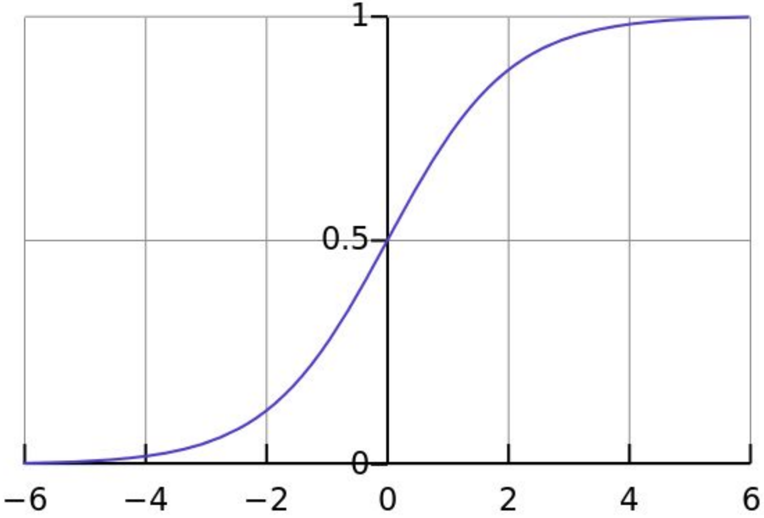
- The model outputs the probability of a categorical target variable Y belonging to a certain class.



Logistic Regression

- The logit model is a modification of linear regression that makes sure to output a probability between 0 and 1 by applying the **sigmoid function.**

$$A = \frac{1}{1+e^{-x}}$$



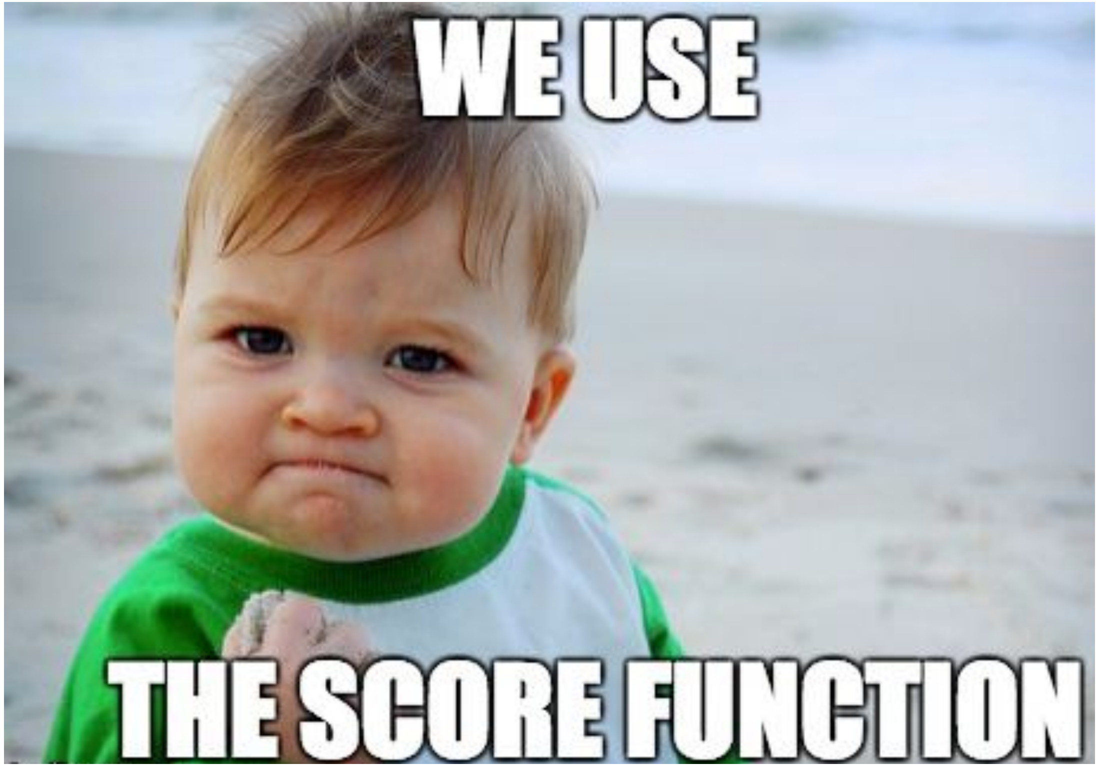
For every example x in the dataset

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))]$$

$$\begin{aligned} &\text{repeat } \{ \\ &\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\ &\qquad (\text{for } j = 1 \text{ and } j = 0) \\ &\} \end{aligned}$$

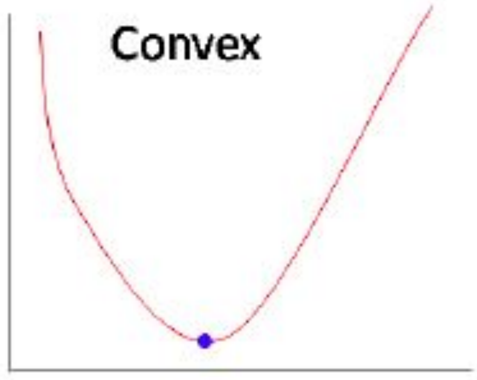
Selecting the Cost Function[5]

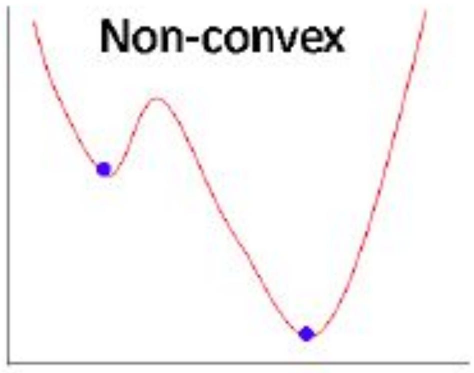


Selecting the cost function

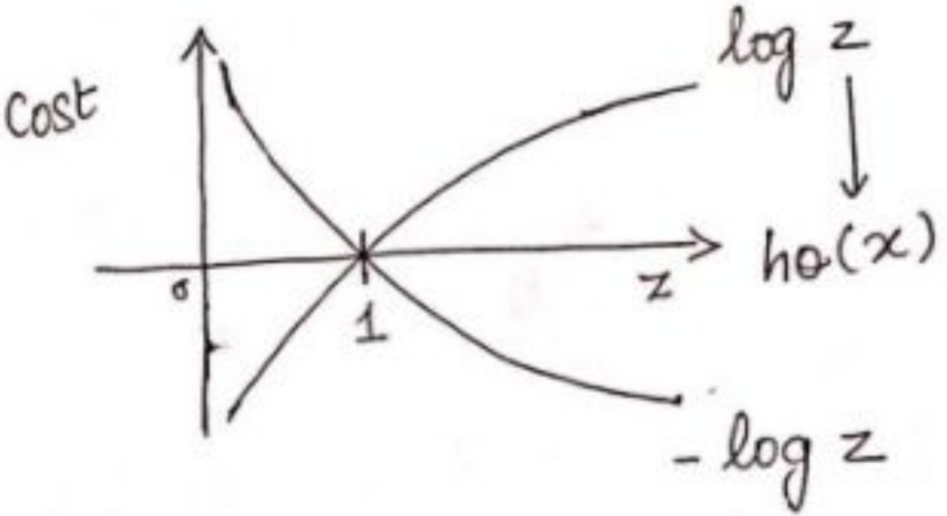
- If mean squared error function is used for logistic regression, then it will be a non-convex function of parameters (theta).

- Gradient descent will converge into global minimum only if the function is convex.





Selecting the cost function

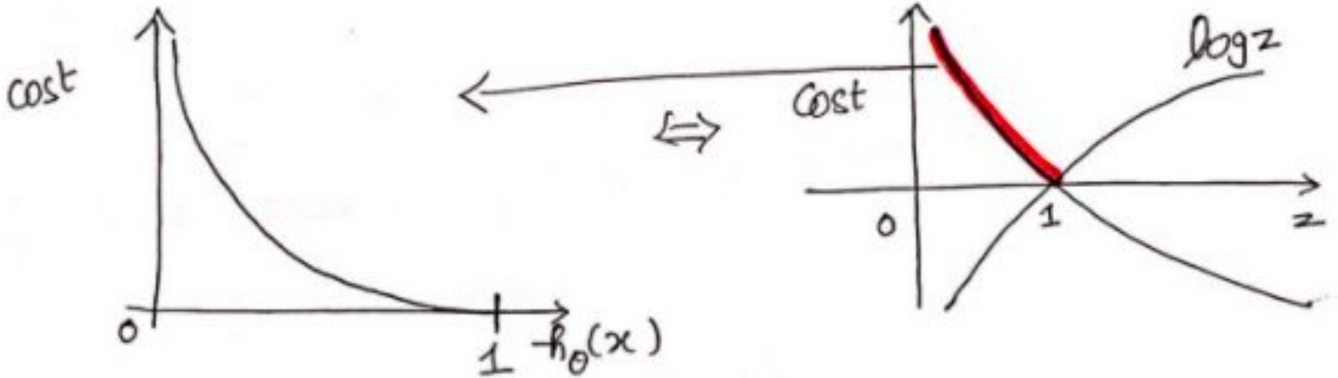


$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Selecting the cost function

If $y = 1$,

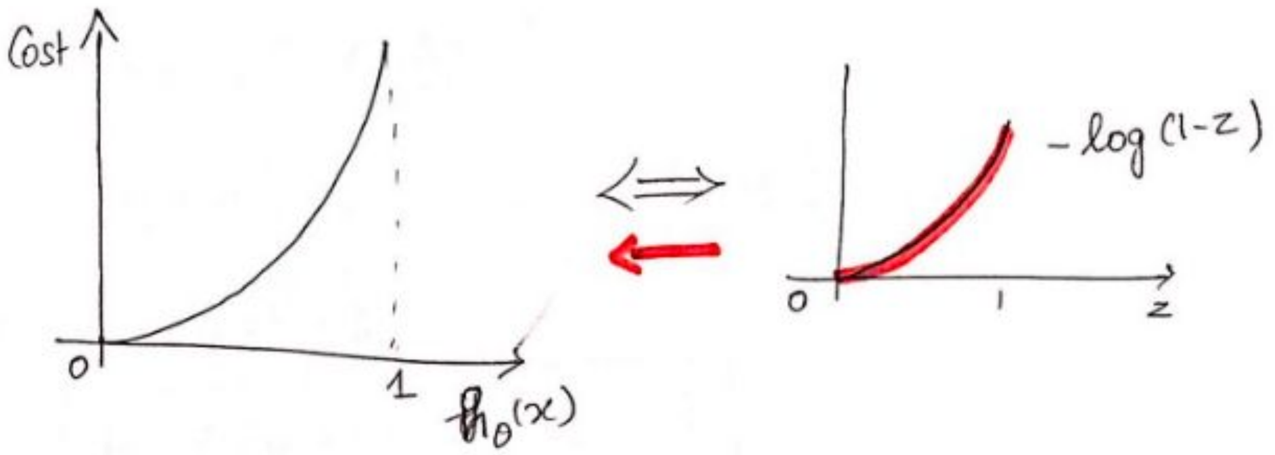
$$\text{Cost}(h_\theta(x), y) = -\log(h_\theta(x))$$



If, $\text{Cost} = 0 \Rightarrow y = 1 \Rightarrow h_\theta(x) = 1$

$\text{Cost} = \text{infinity}$ for $h_\theta(x) = 0$

If $h_\theta(x) = 0$, it is similar to predicting $P(y=1 \mid x; \theta) = 0$



If $\text{cost} = 0 \Rightarrow h_\theta(x) = 0 \Rightarrow y = 0$

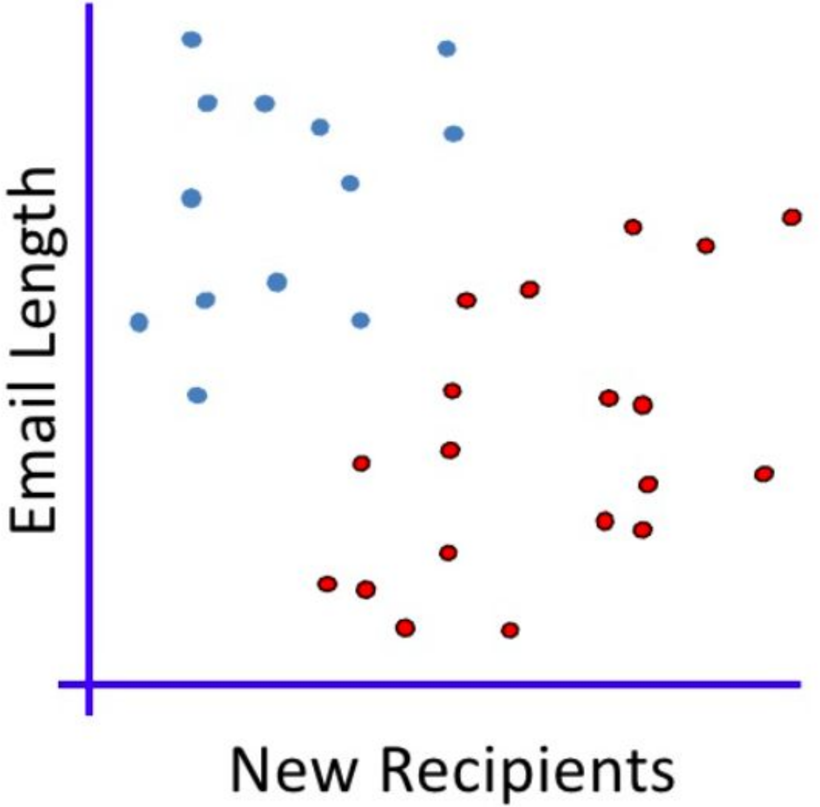
$\text{cost} = \infty \Rightarrow h_\theta(x) = 1$

If $h_\theta(x) = 1$, it is similar to predicting

$P(y=0 \mid x; \theta) = 0$

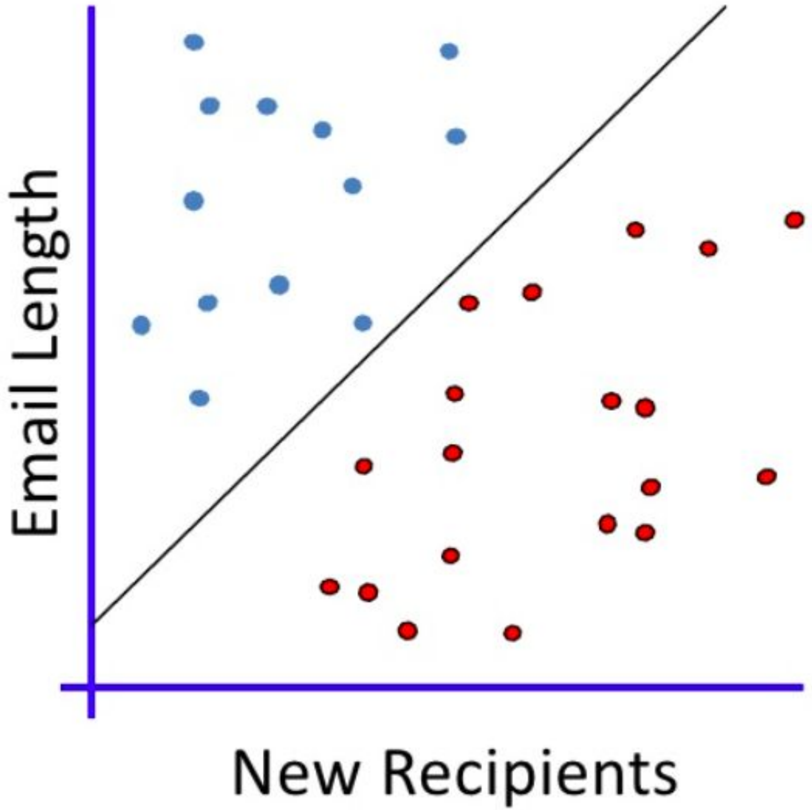
Support Vector Machine

Support Vector Machine



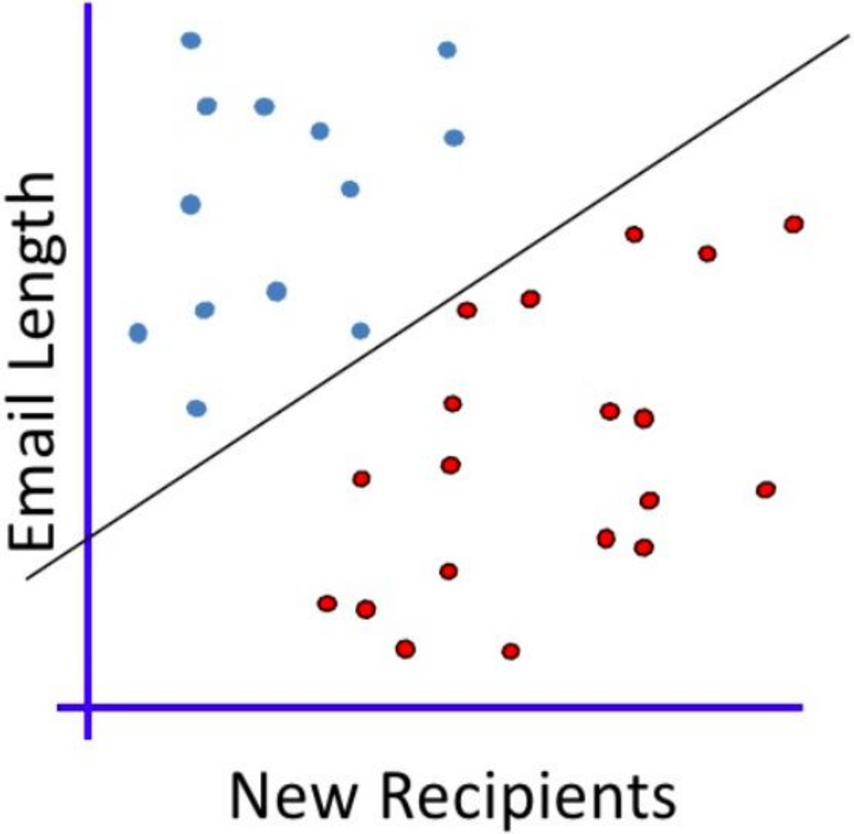
How would you classify this data?

Support Vector Machine



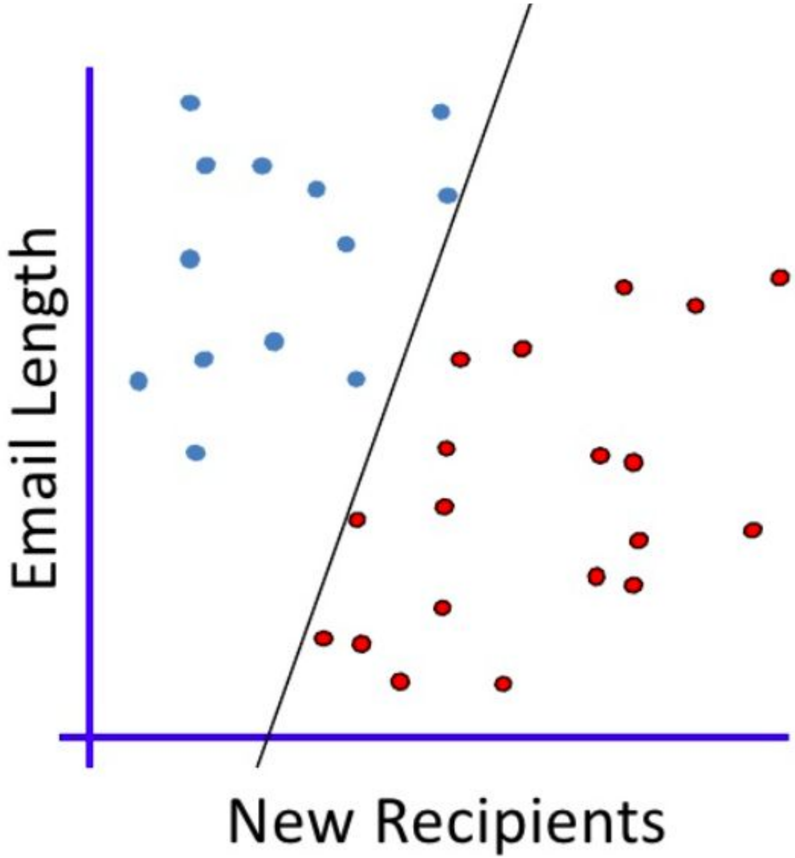
How would you classify this data?

Support Vector Machine



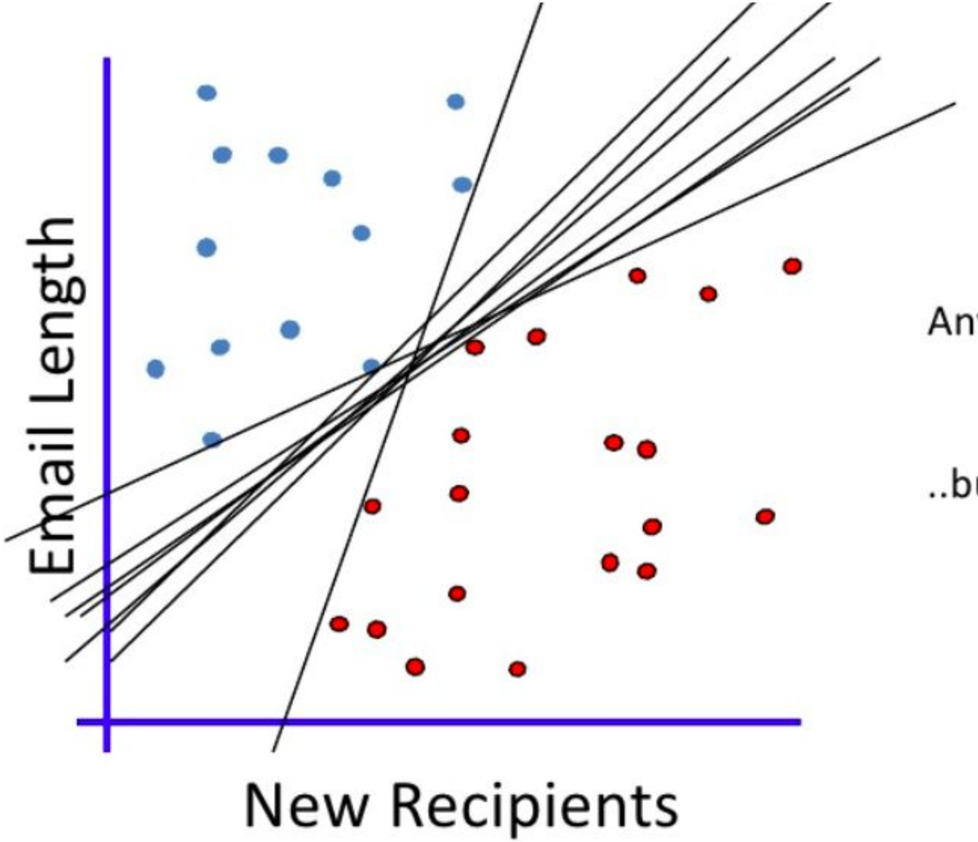
How would you classify this data?

Support Vector Machine



How would you classify this data?

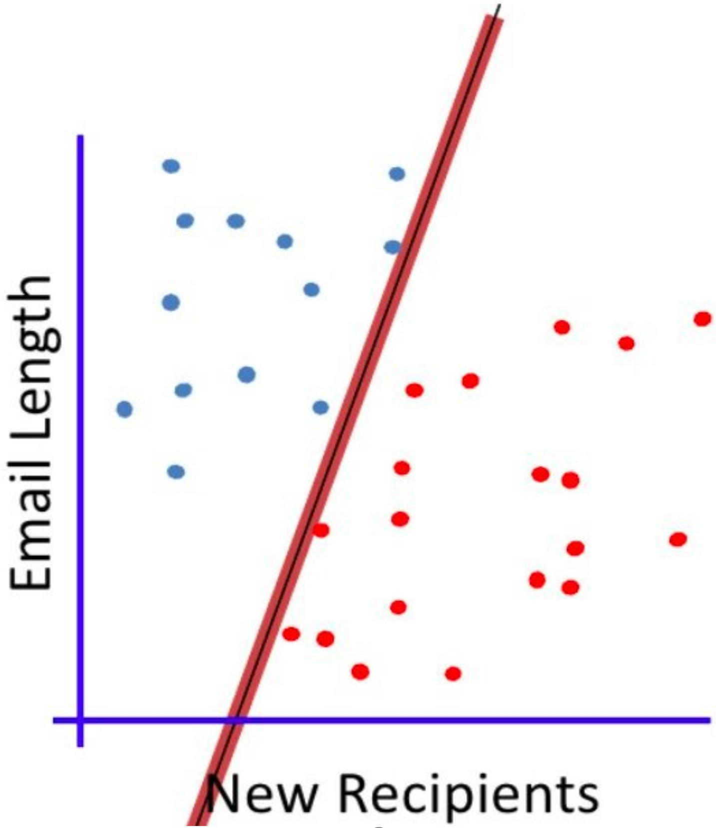
Support Vector Machine



Any of these would be fine..

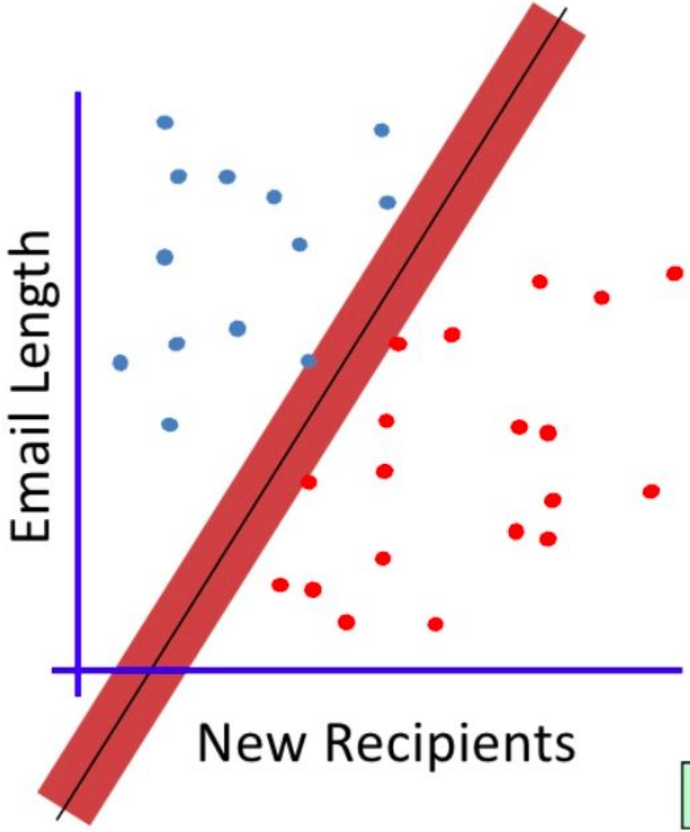
..but which is best?

Support Vector Machine



Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

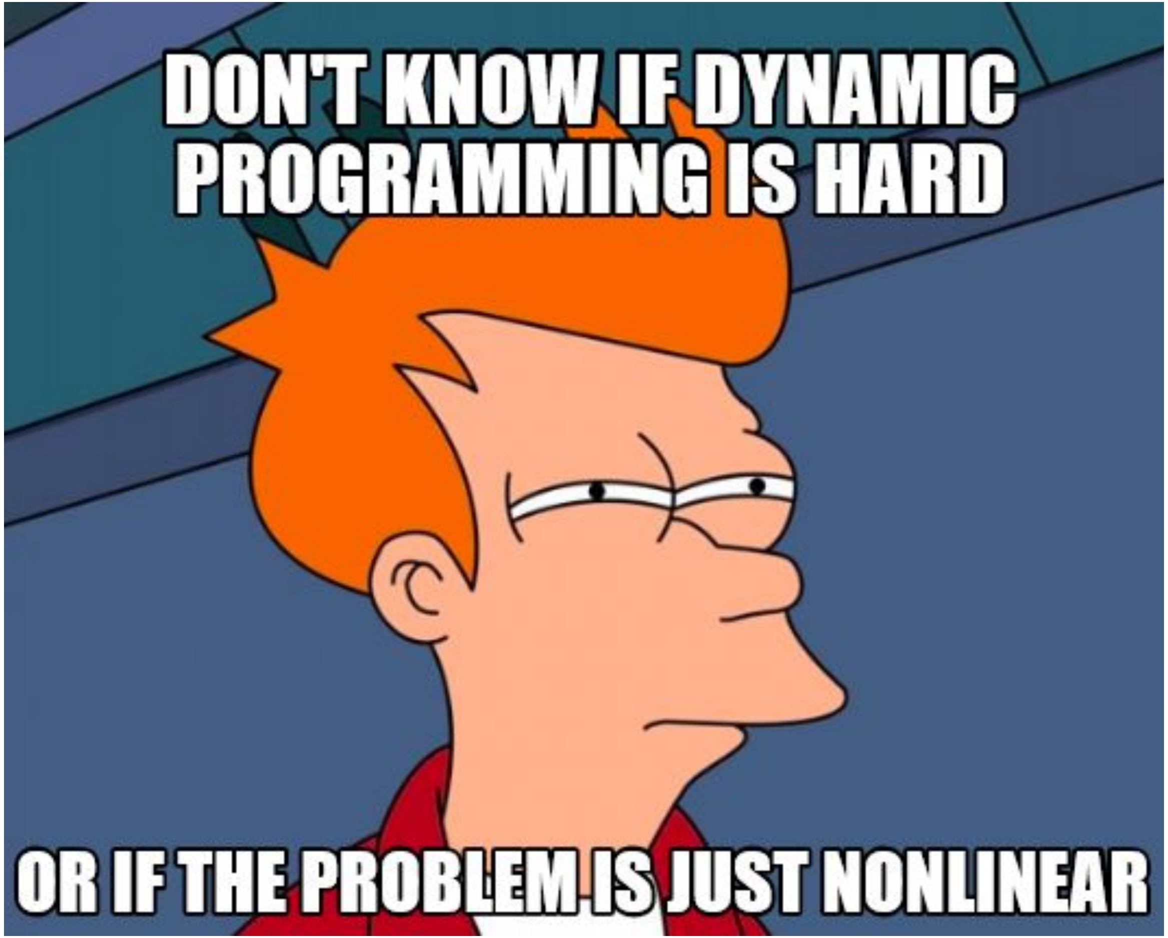
Support Vector Machine



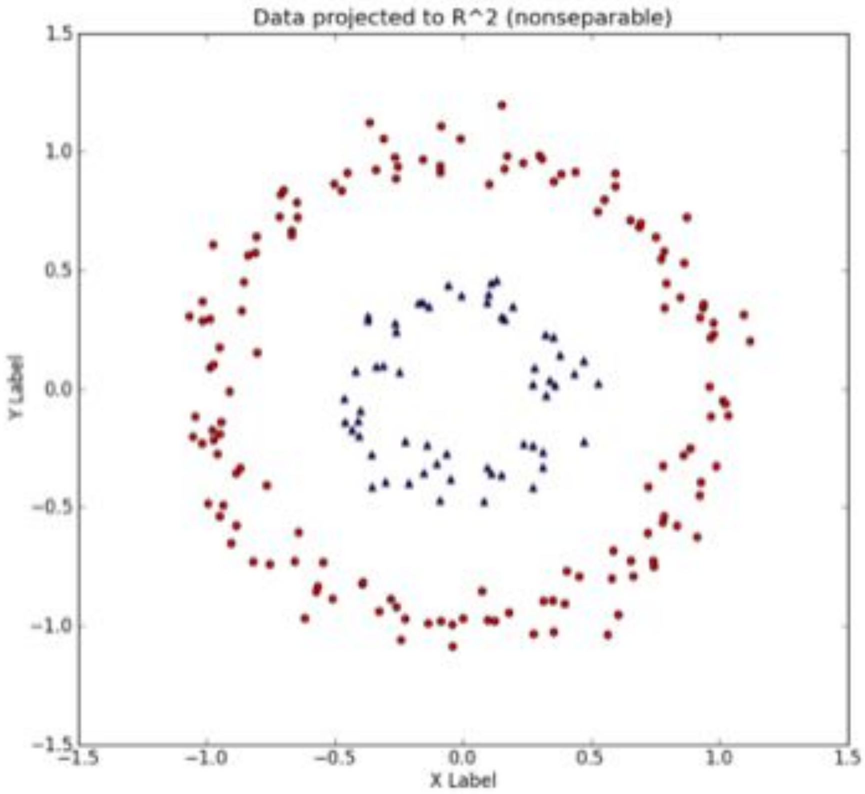
The maximum margin linear classifier is the linear classifier with the, maximum margin.
This is the simplest kind of SVM (Called an LSVM)

Linear SVM

Non Linear Classification



Support Vector Machine



Support Vector Machine

Methods to deal with Non linear data

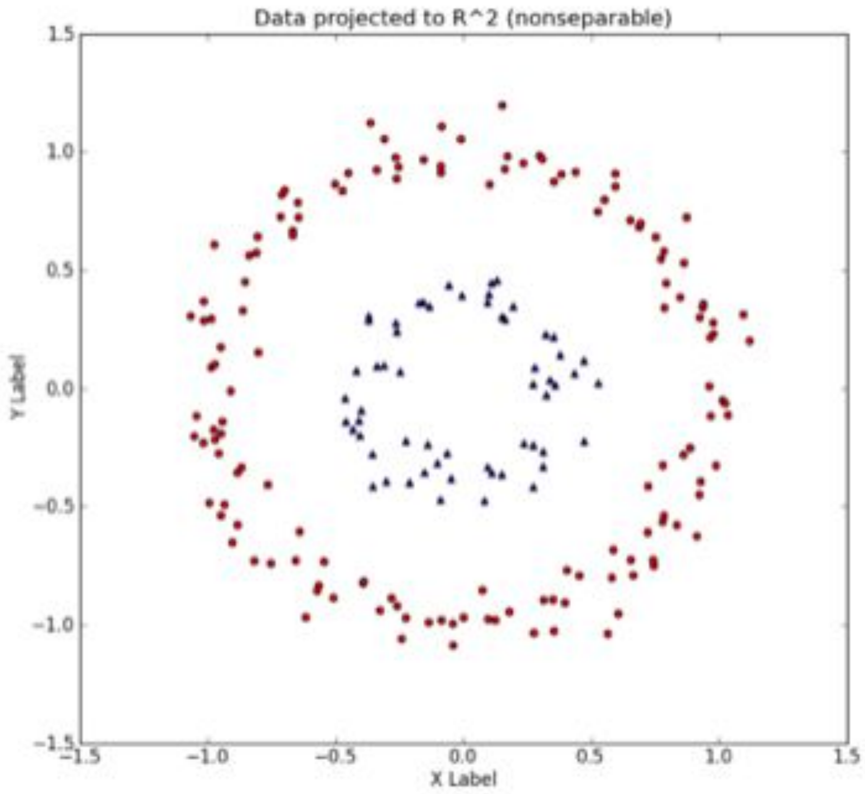
1. Soften the definition of “separate”

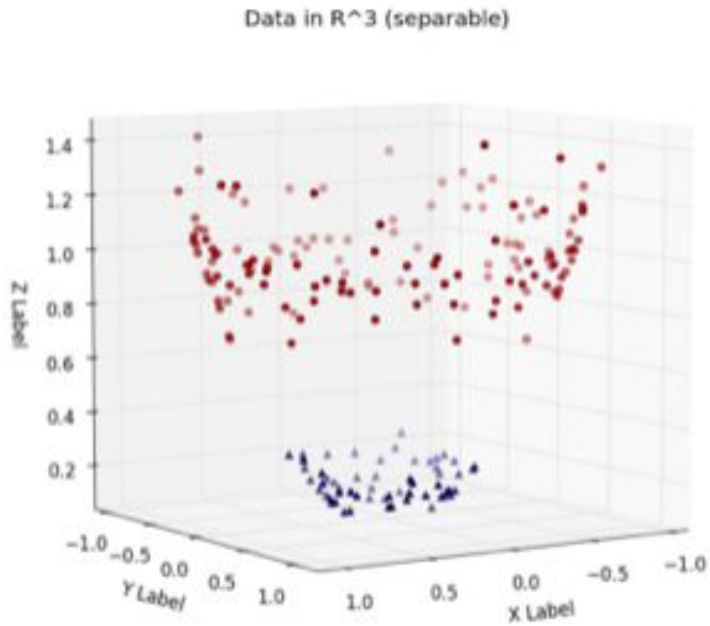
   Allow a few mistakes, meaning we allow some blue points in the red zone or some red points in the blue zone.

2. Throw the data to higher dimension

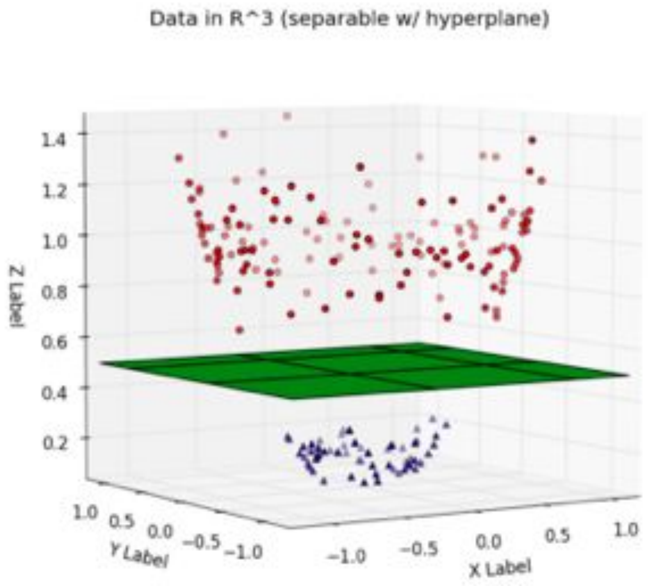
   We can create nonlinear classifiers by increasing the number of dimensions, i.e. include x^2, x^3, even $\cos(x)$, etc.

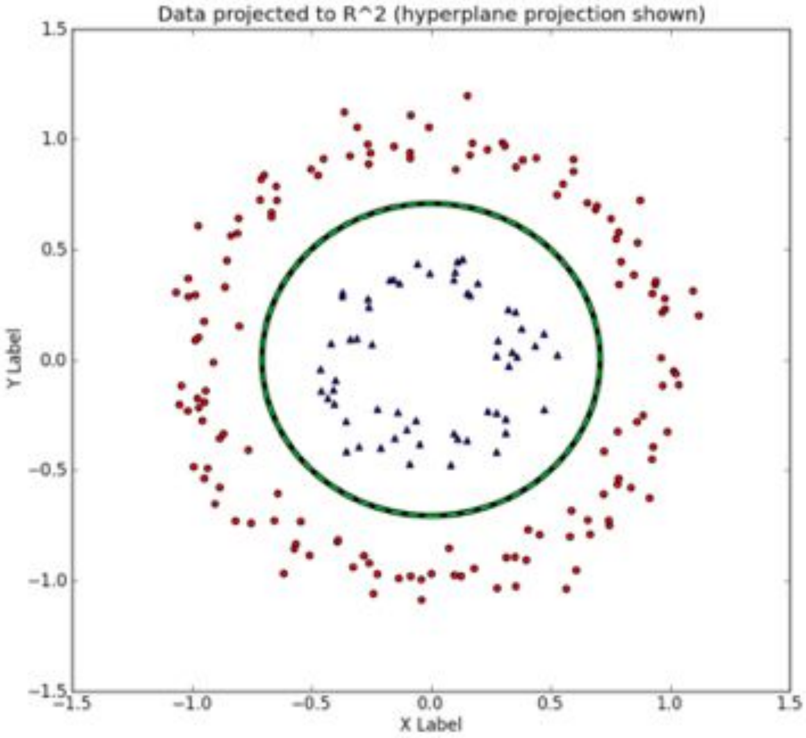
Support Vector Machine





Support Vector Machine





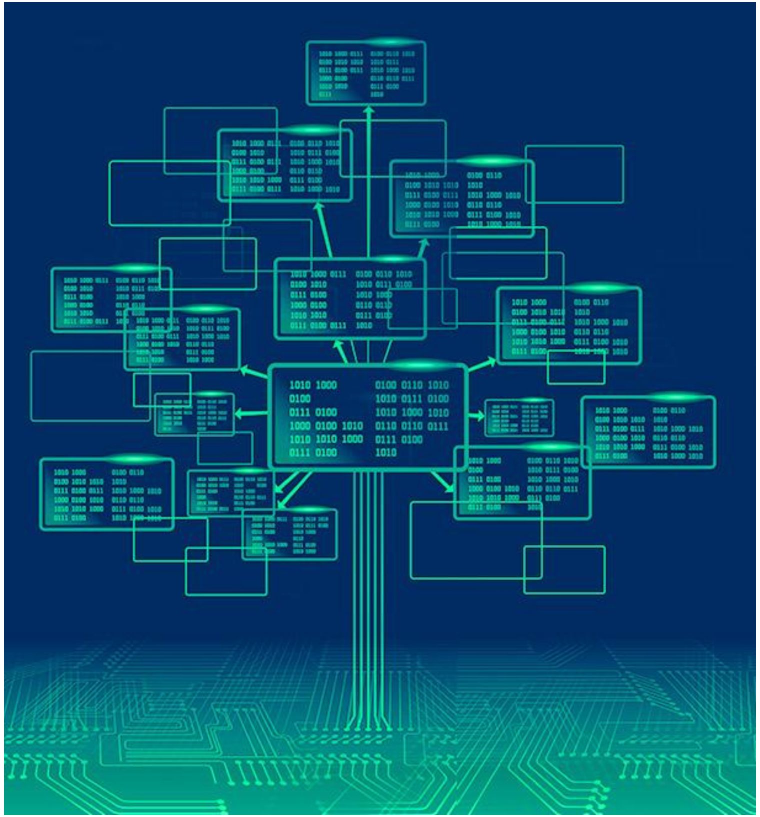
Decision Tree[6]

Decision Tree

- Decision Tree is a type of Supervised Machine Learning where the data is continuously split according to a certain parameter.

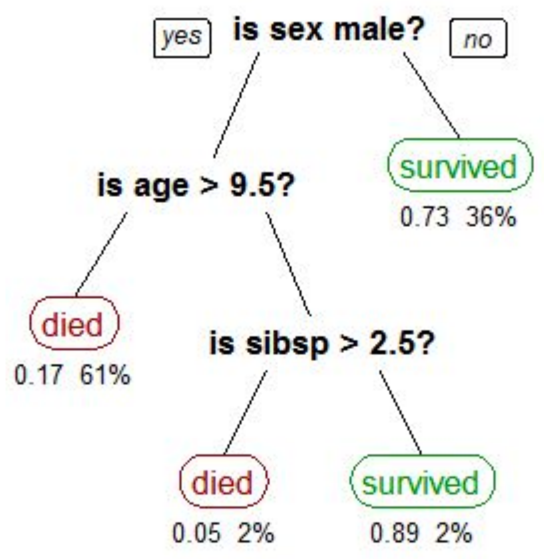
- In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making.

- As the name goes, it uses a tree-like model of decisions.



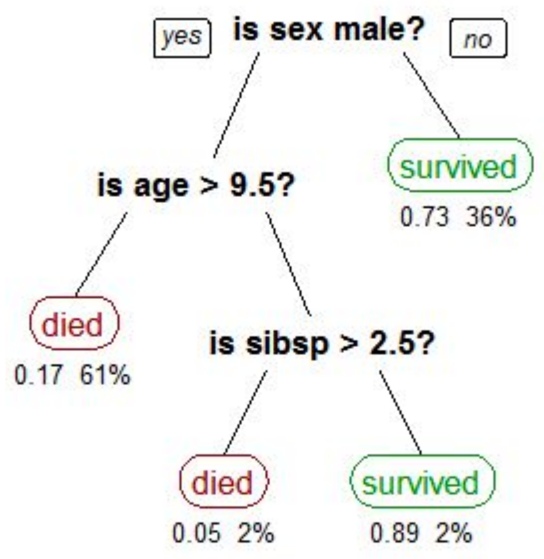
Decision Tree

- Consider an example of Titanic Dataset from Kaggle, where chances of survival of a person is the label.
- A decision tree is an upside down tree with its root at the top.
- The bold text in black represents a condition, based on which the tree splits into branches.
- The end of the branch that doesn't split anymore is the decision/leaf.
- Growing a tree involves deciding on which features to choose and what conditions to use for splitting, along with knowing when to stop.



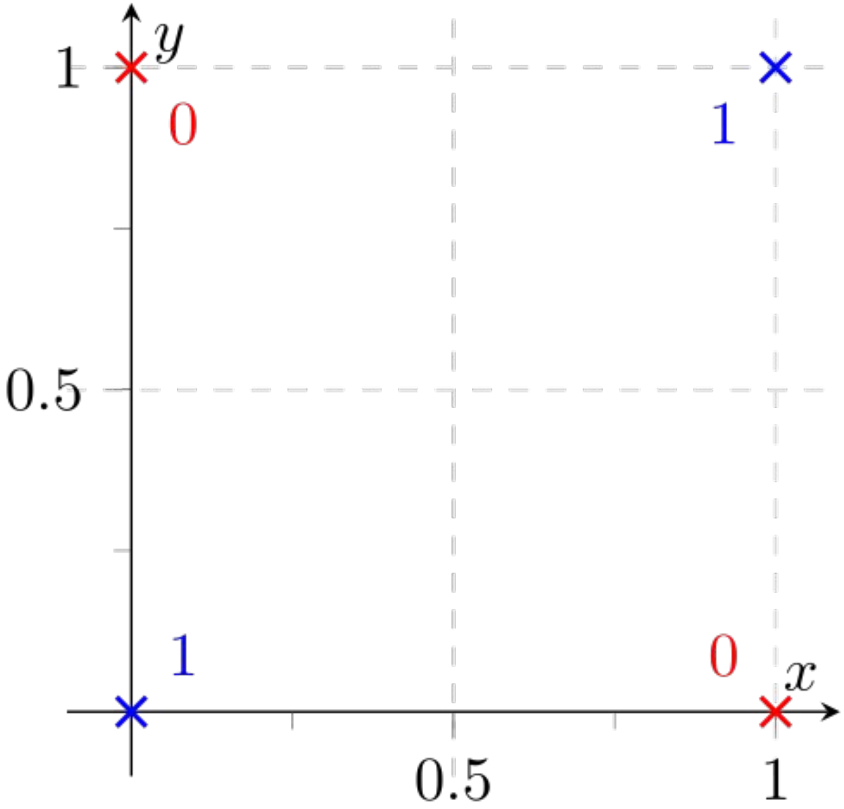
Decision Tree

- In the first split or the root, all features are considered and the training data is divided into groups based on this split.
- Calculate how much accuracy each split will cost us, using a function.
- The split that costs least is chosen, which in our example is sex of the passenger.
- This algorithm is recursive in nature as the groups formed can be subdivided using same strategy.



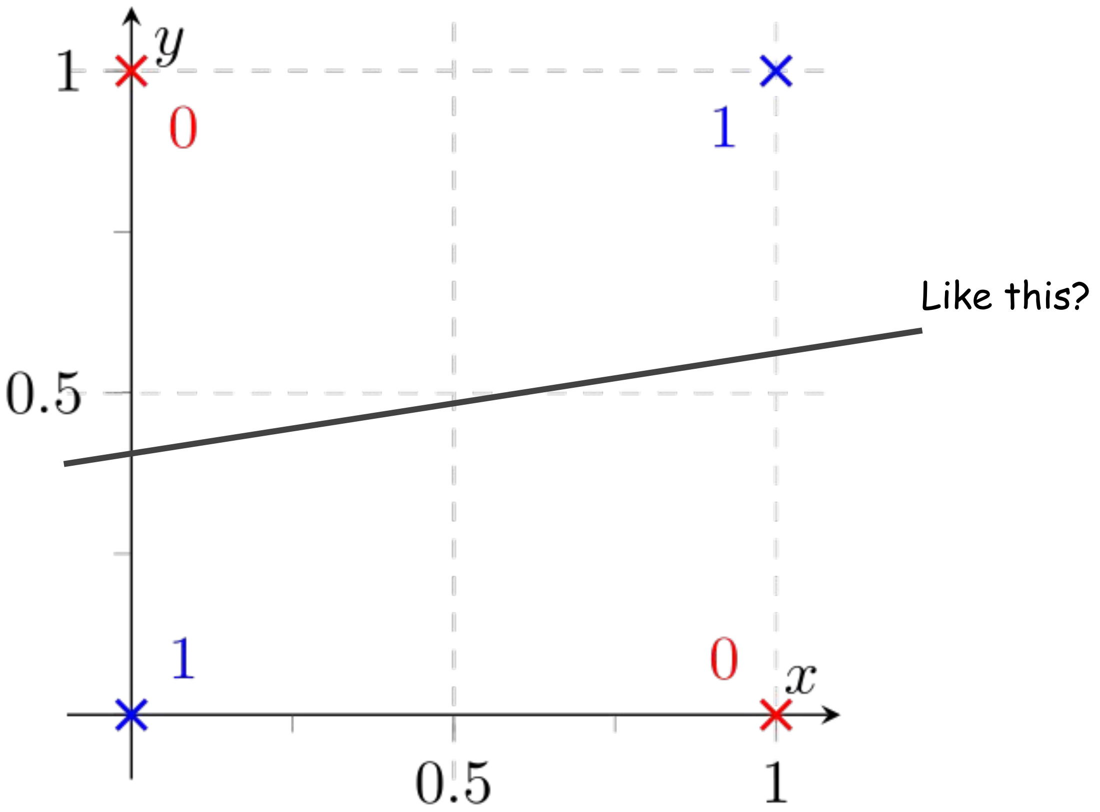
Neural Network

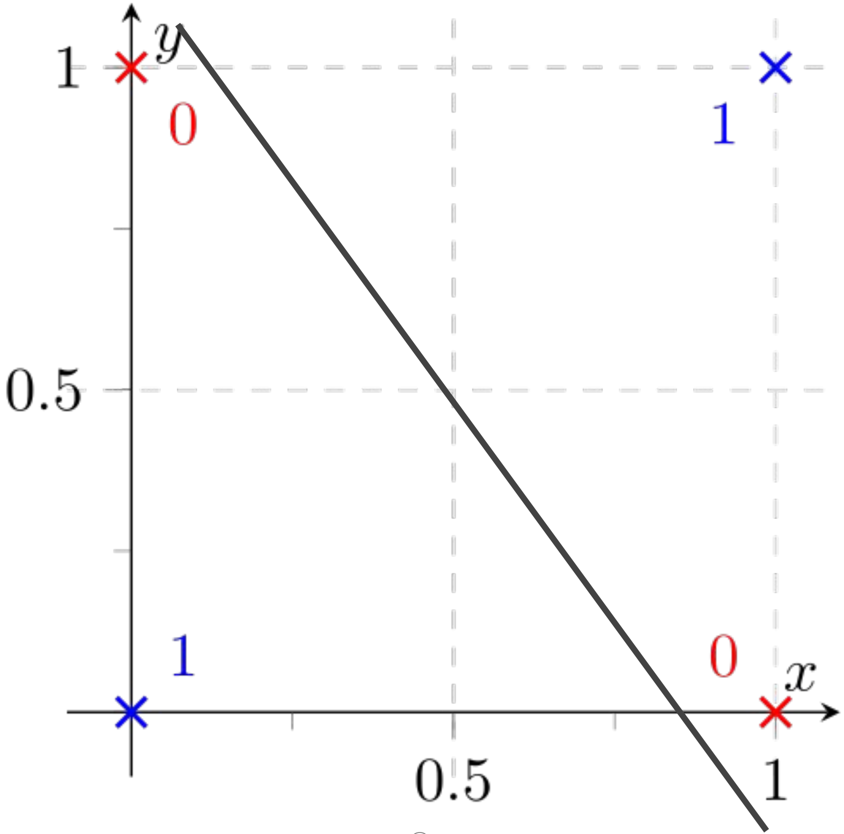
Neural Network



How would you classify this data?

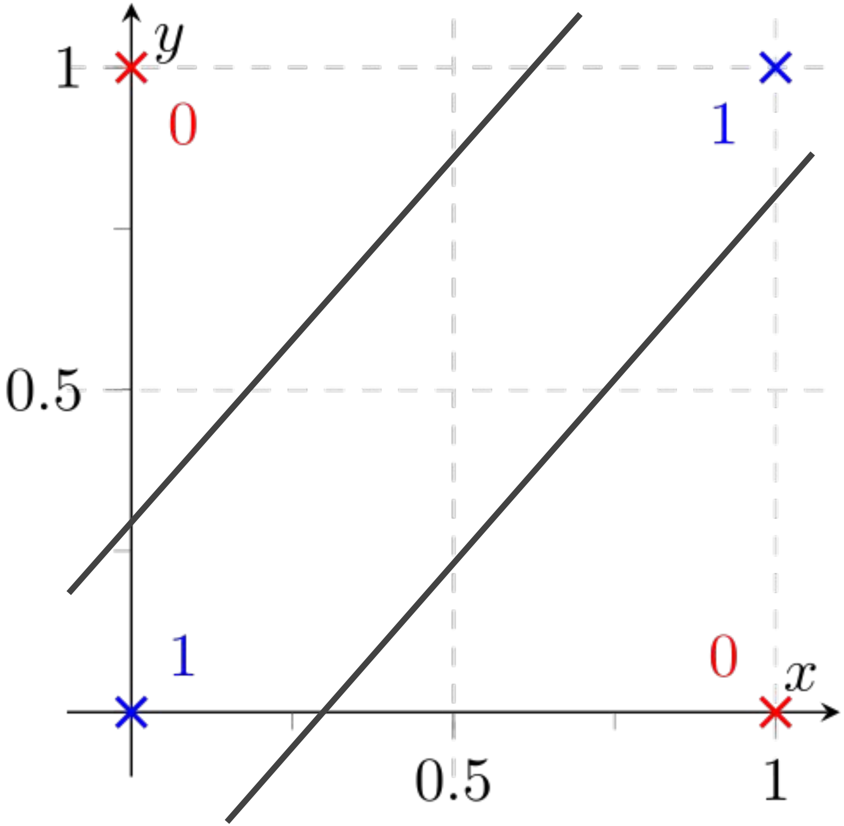
Neural Network





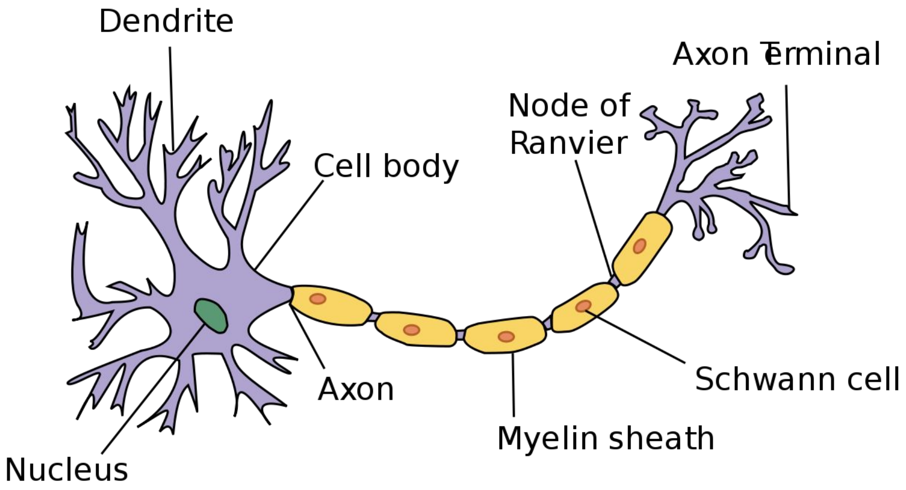
Or like this?

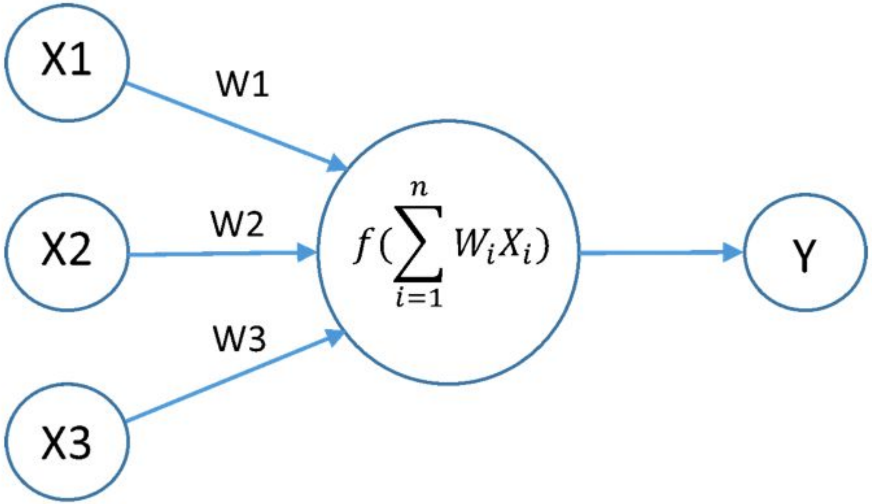
Neural Network



Better use two lines.

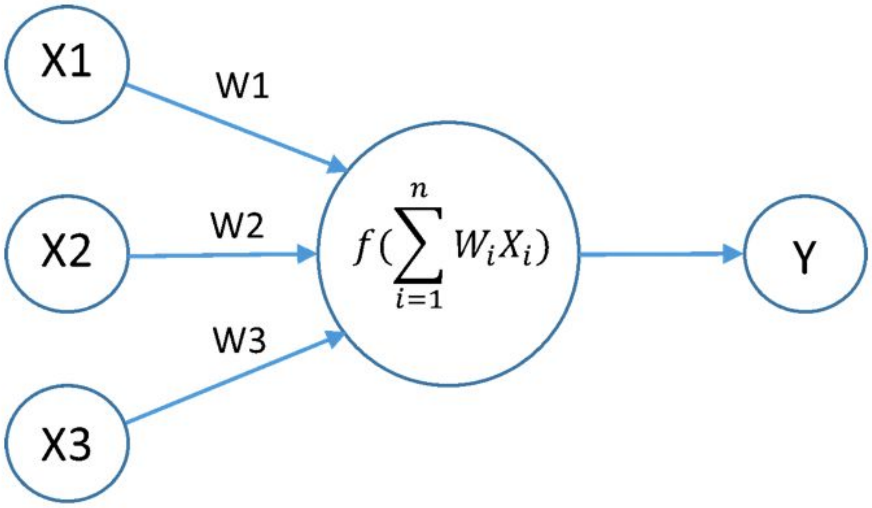
Neuron


Dendrite
Axon Terminal
Node of Ranvier
Cell body
Schwann cell
Axon
Myelin sheath
Nucleus


X1
X2
X3
W1
W2
W3
$f(\sum_{i=1}^{n} W_i X_i)$
Y

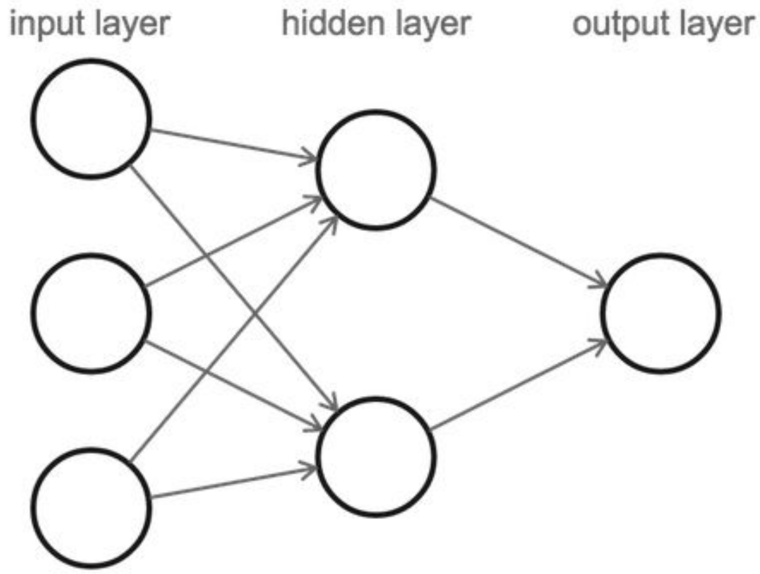
Neuron

- Neurons are processing units and can perform basic arithmetic operations.
- Each connection conveys a signal to the next one.
- A single neuron is composed of weights, biases and a transfer function (activation function).
- Multiple such neurons are used, stacked as layers, for classification.



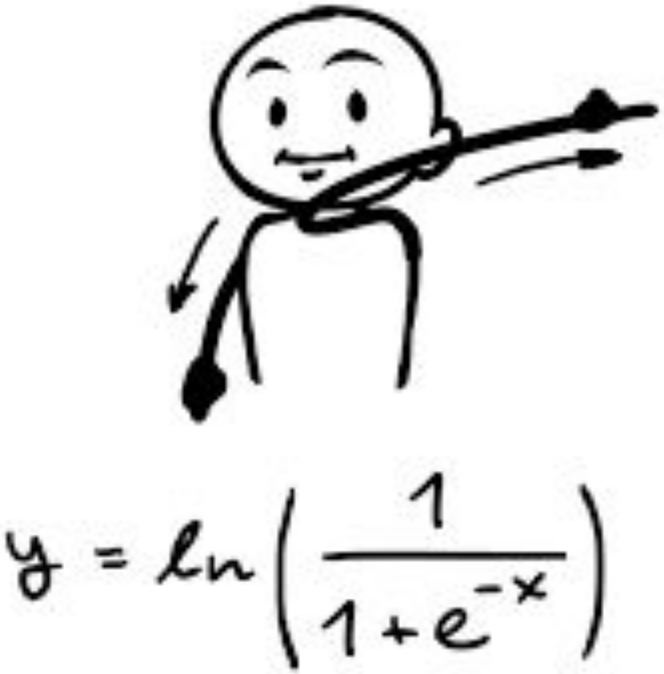
Feed forward Network

- In this network, the first layer of perceptrons/neurons is making three very simple decisions.
- Each of those perceptrons in second layer is making a decision by weighing up the results from the first layer of decision-making.
- In this way a perceptron in the second layer can make a decision at a more complex and more abstract level than perceptrons in the first layer.
- And even more complex decisions can be made by the perceptron in the third layer.
- In this way, a many-layer network of perceptrons can engage in sophisticated decision making.



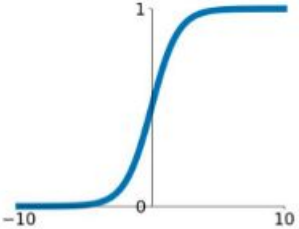
A Typical feed forward Neural Network

Activation Function



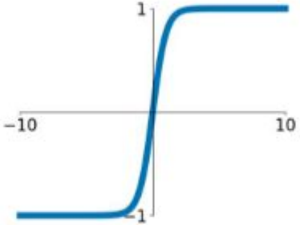
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



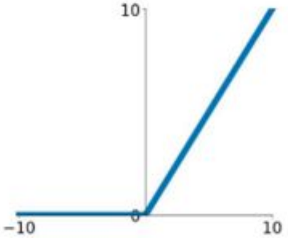
tanh

$\tanh(x)$



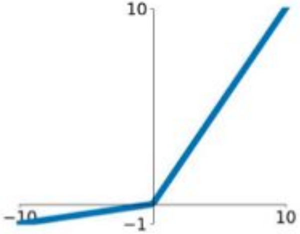
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

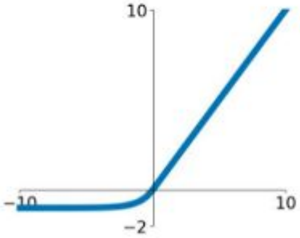


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



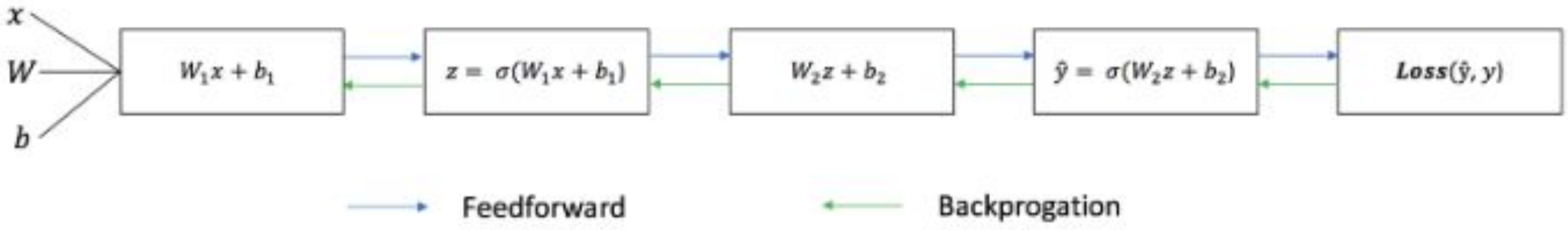
Training a Neural Network[7]

Training a Neural Network

Step 1: Feedforward inputs through the whole network

Step 2: Compute the loss

Step 3: Backpropagate to update the weights



Backpropagation[8]

Refer Link: https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

Unsupervised Learning

- K-Means Clustering
- Hierarchical Clustering
- Dimensionality Reduction
  - PCA
  - T-SNE

Reinforcement Learning

- Exploration / Exploitation
- Markov Decision Process
- Policy Gradient
- Q - Learning
- Deep Reinforcement Learning

Some Tips

- Papers
  - ArXiv Sanity Preserver
- Twitter
  - Arxiv daily
  - Ian Goodfellow
  - Siraj Raval
  - Submarine
  - OpenAI
  - DeepMind
- Reddit
- YouTube
  - 2 Minute Paper
  - Siraj Raval
- Medium
  - Towards Data Science
- Courses
  - Fast.ai
  - Deeplearning.ai
  - Machine learning Andrew Ng

Thanks!

Contact Us

Aman Agarwal

- Email: 15bce006@nirmauni.ac.in
- LinkedIn: aman-agarwal-743548137
- GitHub: First-Of-His-Name
- Twitter: @aman_ag11

Aditya Mishra

- Email: 15bce003@nirmauni.ac.in
- GitHub: aditya985

References

1. https://www.dropbox.com/s/e38nil1dnl7481q/machine_learning.pdf?dl=0
2. https://medium.com/machine-learning-for-humans/supervised-learning-740383a2feab
3. https://iamtrask.github.io/2015/07/27/python-network-part2/
4. https://www.jeremyjordan.me/nn-learning-rate/
5. https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc
6. https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052
7. https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6
8. https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/