Machine Learning Approaches for advanced Industrial Applications

Allmin Susaiyah

Scientist, Philips Research, Eindhoven, Netherlands

Marie Curie Fellow

Educational Background

Bachelor of Engineering
(Electrical and
Electronics Engineering)

Master of Science
(By Research)
Image Processing and ML

Exchange Student
Signal Processing and
Computer Vision

Professional Background

Medaino Healthcare
DSP Manager

Philips Innovation Campus

Philips Research,
Bangalore,
Senior Research Engineer

Philips Research
Eindhoven
The Netherlands
(Headquarters)

Areas of Research

- Non-Obtrusive Vital Sign Monitoring
- Medical Imaging Decision Support Systems
  - Histopathology Diagnosis
  - Lung Cancer Diagnosis
  - Liver Cancer Diagnosis
  - Stroke Diagnosis
  - Super Resolution
- Data Science
  - Missing Value Imputation
  - Predictive Maintenance
  - Insight Mining

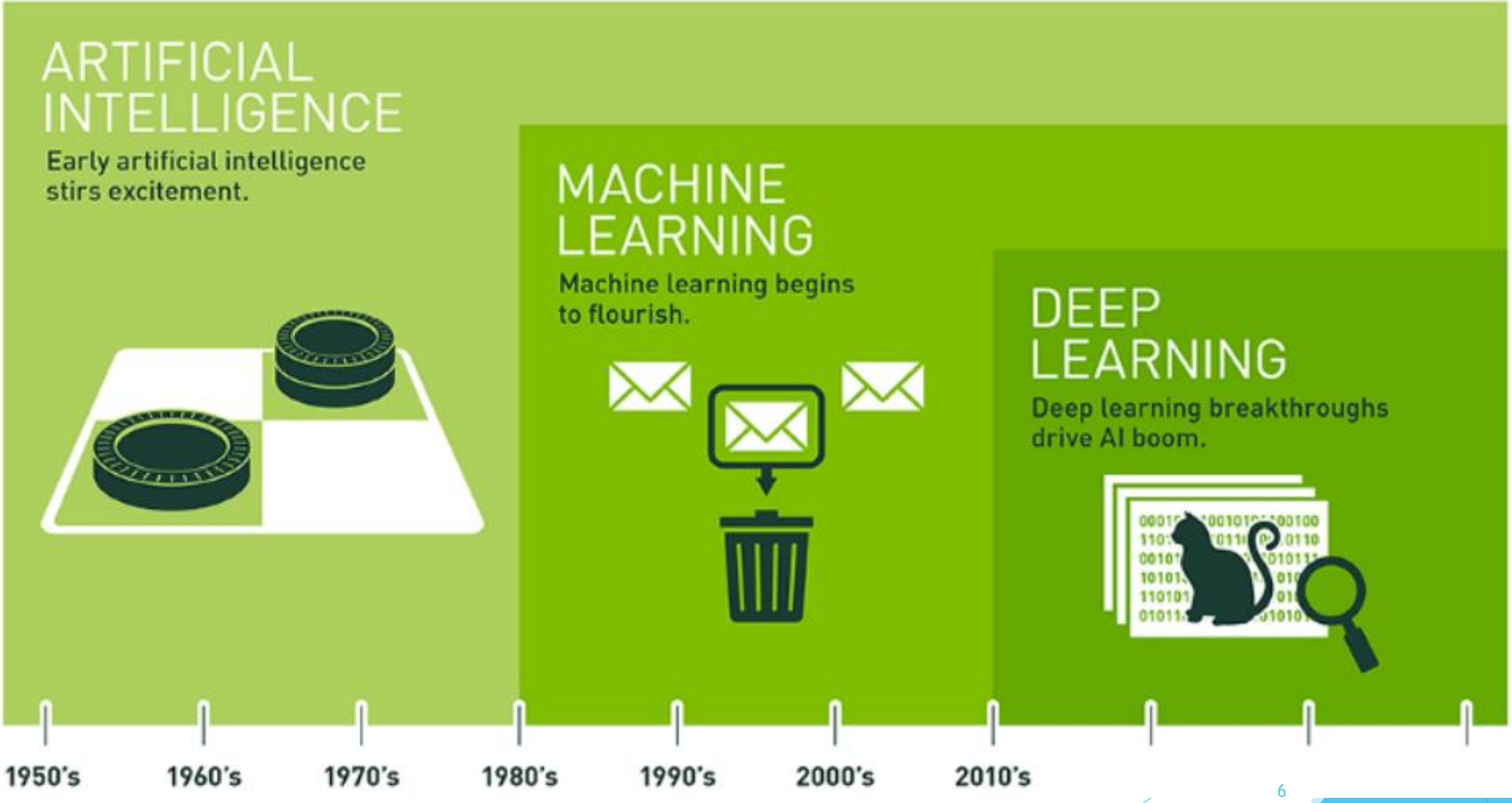
Contents

- Introduction:
  - in-depth exposure to ML approaches for advanced industrial applications will be provided.
  - Much focus will be on ML approaches on Images with a brief overview of other applications.
  - The talks will have three sessions as listed below. Prior knowledge on ML or python is not required. Basic programing skills on any language will be helpful but not mandatory.
- **Phase1:**
  - Session1 - (18th Nov, 11:30AM to 1:00PM IST): Introduction to ML and Python
  - Session2 - (18th Nov, 2:30 PM to 4:00PM IST): Traditional Machine Learning and Deep Learning
  - Session3 - (20th Nov, 2:30 PM to 4:00PM IST): Image Classification using Deep Learning
- **Phase2:**
  - Session1- (9th Dec, 11:30AM to 1:00PM IST): Introduction to ML and Python
  - Session2- (9th Dec, 2:30 PM to 4:00PM IST): Traditional Machine Learning and Deep Learning
  - Session3- (11th Dec, 2:30 PM to 4:00PM IST): Image Classification using Deep Learning
- **Phase3:**
  - Session1- (6th Jan, 11:30AM to 1:00PM IST): Introduction to ML and Python
  - Session2- (6th Jan, 2:30 PM to 4:00PM IST): Traditional Machine Learning and Deep Learning
  - Session3- (8th Jan, 2:30 PM to 4:00PM IST): Image Classification using Deep Learning

Session1
Introduction to Machine Learning and Python

repo: https://github.com/allmin/mlworkshop

Artificial Intelligence and its evolution



Explicit Rule Based | Implicit Data Driven

ML in Industries..

Healthcare

- Clinical Decision Support
- Prediction, Prevention, Wellness, Aging, and Rehabilitation
- Drug discovery
- GENOME Analysis

Media and e-commerce

- Animating faces
- Music composition
- News Report generation
- Animating NPCs in video Games

Transportation

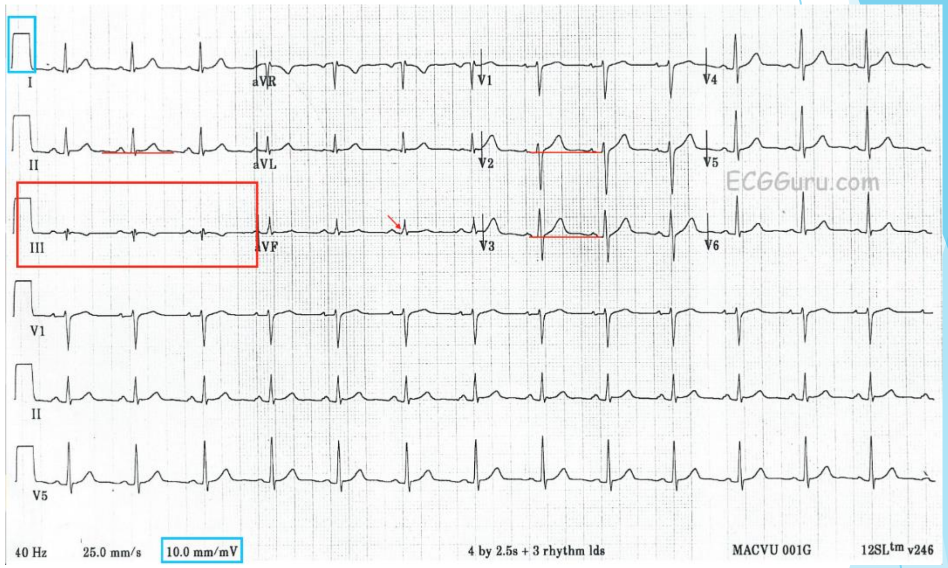
- Self-driving
- Automatic Transmission
- Self-parking
- Traffic Management
- self-driving truck platoons

Other

- Voice Assistance
- Business
- Aviation
- Safety and Security

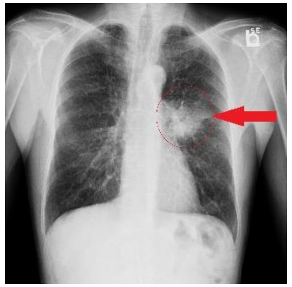
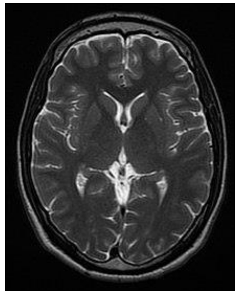
ML in the Healthcare Industry

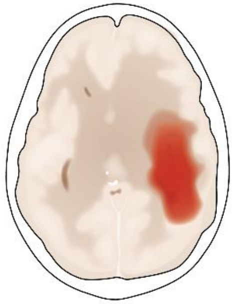
- Signals
  - Detection of Abnormalities in Biosignals
    - Predictive
    - Reactive
  - Signal Denoising
  - Signal Modelling and Synthesis
  - Treatment planning
  - Fall detection



ML in the Healthcare Industry

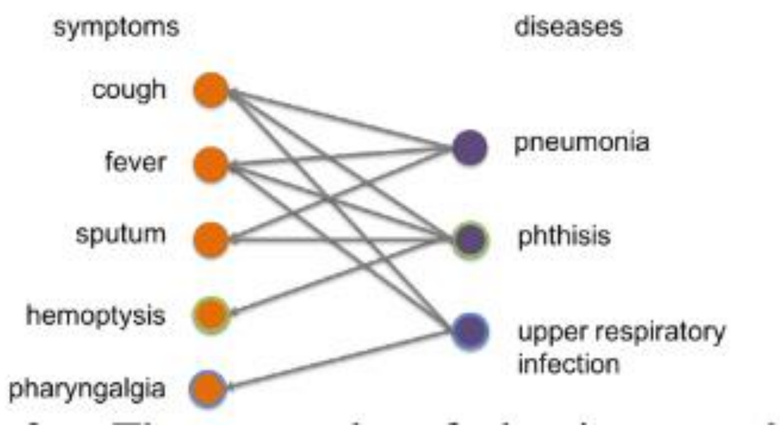
- Images
  - Diagnosis
    - Imaging Modalities: CT, MRI, US
    - Classification
    - Delineation
  - Telemedicine
    - Compression
  - Therapy
    - Tracking Progress


ML in the Healthcare Industry

- Text
  - Clinical Reports
  - Insights from Electronic Medical Records
  - Patient history analysis
  - Population health management
  - Case retrival
- Try out:
  - https://app.inferkit.com/demo

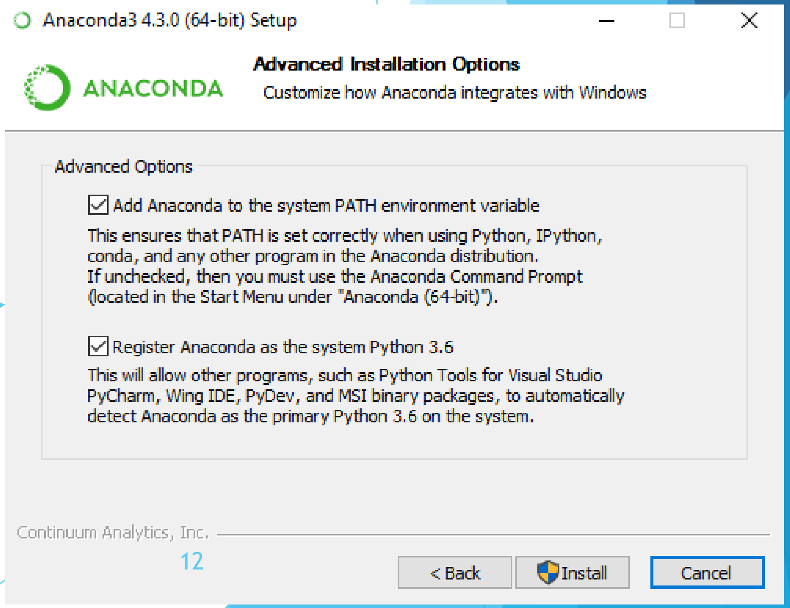


Python

- High-level and General Purpose programing language
- Interpreter Based
- Support Object Oriented Programing
- Easy to understand and write code
- Freely available libraries for most scientific and mathematical computing
- The To-go Programing Language for ML
- Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify

Installing Anaconda

- Log into a windows 64 bit system as admin
- Install and open gitbash from https://git-scm.com/download/win
  - cd C:/
  - git clone https://github.com/allmin/mlworkshop.git
- Download
  - https://repo.continuum.io/archive/Anaconda3-4.3.0-Windows-x86_64.exe
- Navigate to the Anaconda3-4.3.0-Windows-x86_64.exe installer
  - Right click and run as administrator
  - Click yes on the prompt
  - Click **Next -> I Agree -> All Users -> C:\Anaconda3-> Next**
    - Make sure both check boxes are ticked
  - Click Install



Testing if installation was successful

- Open command prompt (winkey + R, type cmd, return)
- type
  - conda info
- You must get a list of conda information like below

```
Current conda install:

               platform : win-64
          conda version : 4.3.29
       conda is private : False
      conda-env version : 4.3.29
    conda-build version : not installed
         python version : 3.6.1.final.0
       requests version : 2.14.2
```

- If you get it, your Anaconda installation was successful!!

Opening Workshop Materials

- Open jupyter notebook by using following command
  - Open anaconda prompt from start menu
  - cd C:/mlworkshop
  - jupyter-notebook
  - Web browser opens with a webpage
  - Click on PythonBasics.ipynb

Now, let us see python in action..

Session2- Traditional Machine Learning and Deep Learning

ML Practices

- Typical ML Workflow
  - Data Preprocessing
  - Model Learning
  - Model Validation
  - Model Deployment

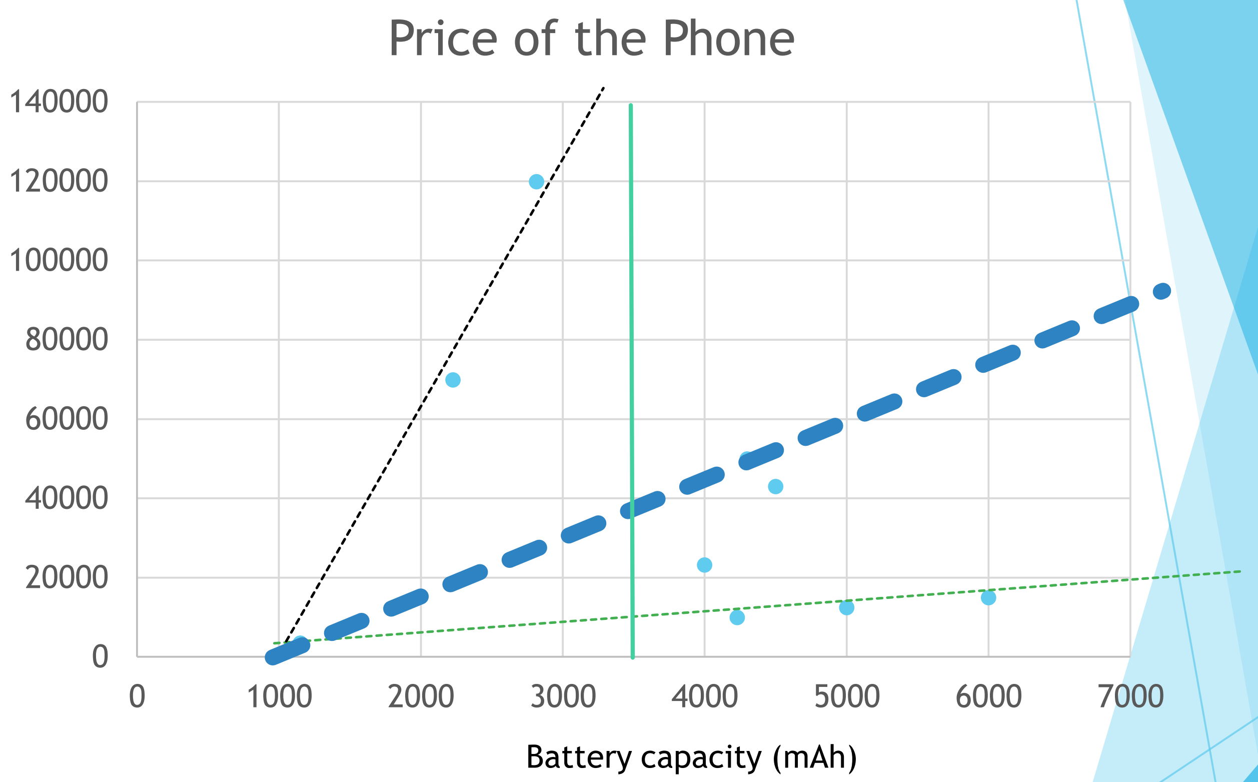
SUPERVISED LEARNING:
Data + Labels

UNSUPERVISED LEARNING:
Data Only

Supervised Algorithm 1

- Linear Regression

mobile	Battery Capacity (mAh)	Price of the Phone
Galaxy F41	6000	15000
Vivo V20	4000	23241
OnePlus 8T	4500	43000
Oppo A15	4230	9990
Iphone 12 Mini	2227	69900
Nokia 225	1150	3499
Realme 7i	5000	12447
Iphone 12 Pro	2815	?
LG Velvet	4300	?



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

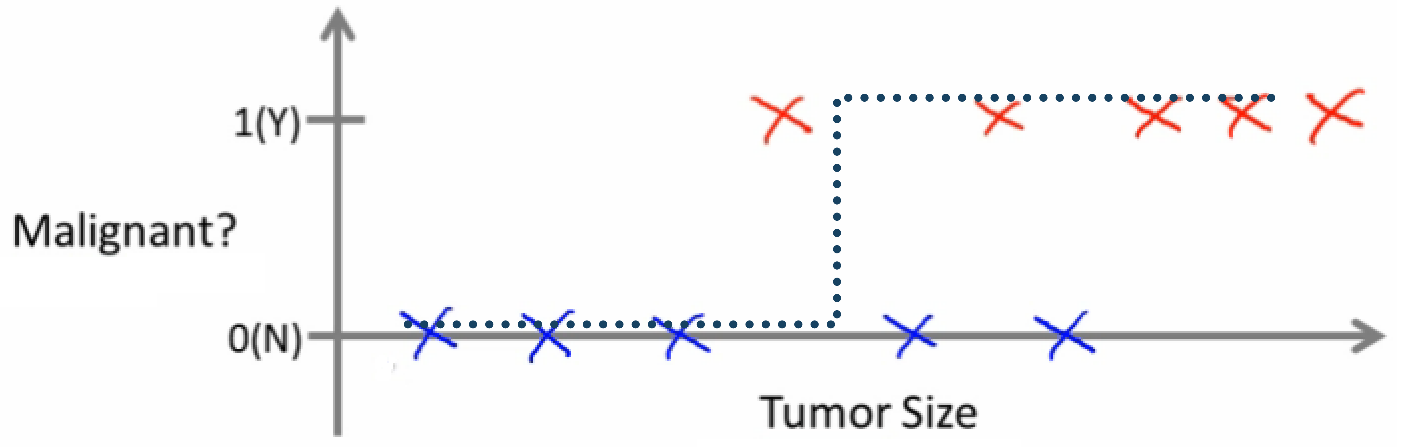
$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\underset{\theta_1}{\text{minimize}}\ J(\theta_1)$$

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

Supervised Algorithm 2

- Logistic Regression



$\cancel{h_\theta(x) = \theta_0 + \theta_1 x}$

$h_\theta(x) = g((\theta^T x))$

$g(z) = 1/(1 + e^{-z})$

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

$$\underset{\theta_1}{\text{minimize}}\ J(\theta_1)$$

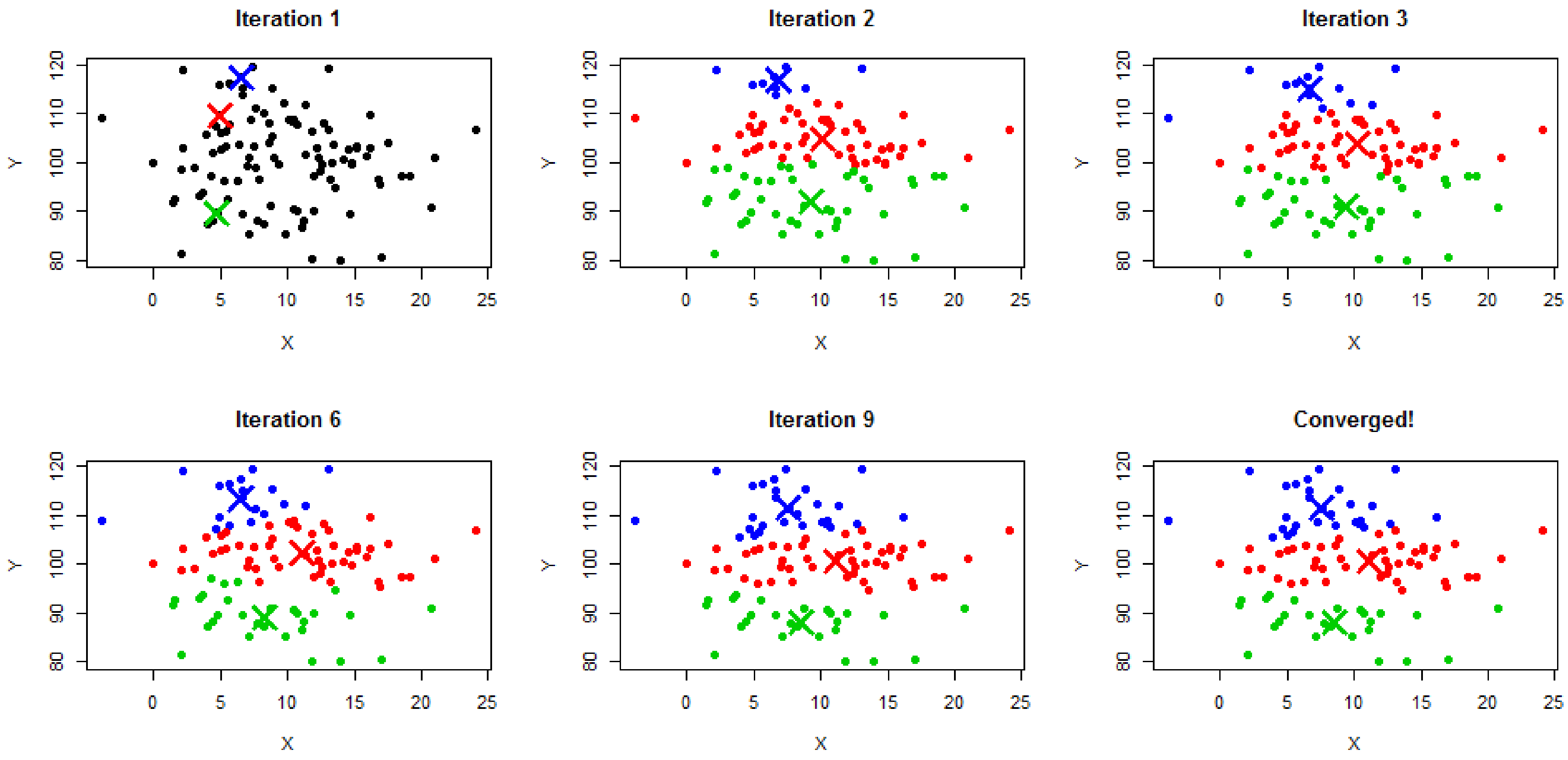
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

ML JARGON

- Feature set:
  - Input Data
- Labels, Ground Truth, Annotation :
  - Target Data
- Classifier, Regression :
  - An algorithm used for learning predicting whether the given sample is class A, B or …..
- Dimensionality Reduction:
  - If too many features, we derive fewer features from original feature set
- Feature Selection:
  - If too many feature, we ignore irrelevant or insignificant features
- Overfitting or Biased:
  - This is when you learn something **too specific** to your data set and therefore fail to generalize well to unseen data
- Under fitting or regularized:
  - Opposite of overfitting

Unsupervised Algorithm – K Means

- Data Driven
- No Labels
- Iterative process



COMMON ML Techniques

- SVM : Support Vector Machine
- RF : Random Forest
- ANN : Artificial Neural Networks
- Re-inforcement Learning
- Genetic Learning
- Active Learning
- Generative Adversarial Networks

BASIC IMPLEMENTATION OF ML IN PYTHON

	ip1	op
Training	6000	15000
	4000	23241
	4500	43000
	4230	9990
	2227	69900
Validation	1150	3499
	5000	12447
Deployment	2815	?
	4300	?

```
import numpy as np
X = np.array([[6000], [4000], [4500], [4230], [2227]]) #training
y = np.array([15000, 23241, 43000, 9990, 69900])

from sklearn.linear_model import LinearRegression
clf = LinearRegression()
clf.fit(X, y)
coef = clf.coef_
print(coef)

X_ = np.array([[1150], [5000]]) #validation
y_ = np.array([3499, 12447])
y_predicted = clf.predict(X_)
print(y_predicted, y_)
from sklearn.metrics import mean_absolute_error
error = mean_absolute_error(y_, y_predicted)
print('error:',error)

print(clf.predict([[2815]])) #deployment
print(clf.predict([[4300]]))
```

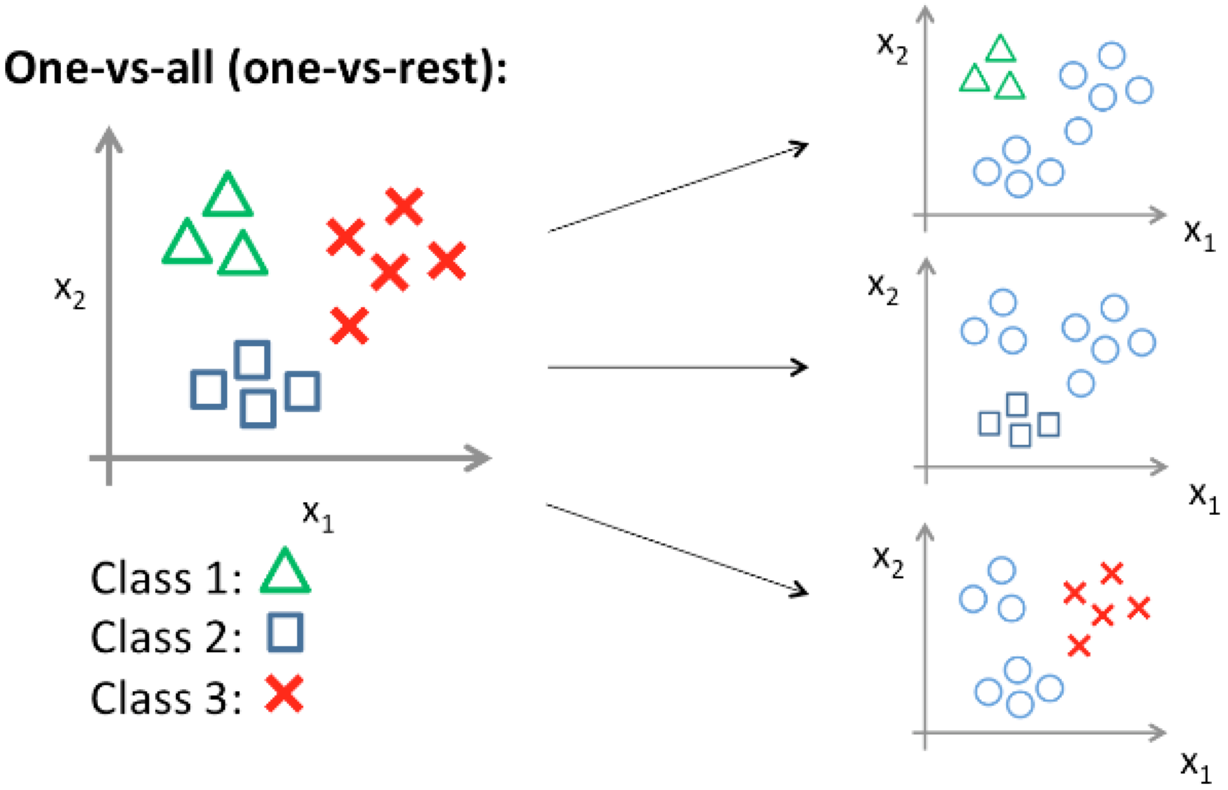
Feature Selection

- Genetic Algorithm
- Information Gain

Feature Reduction

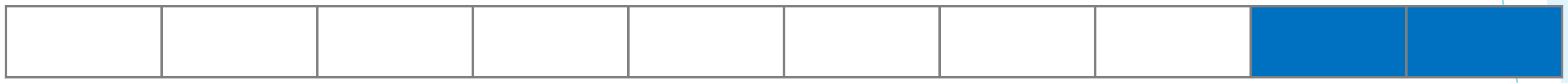
- Principal Component Analysis

One Versus All Classification

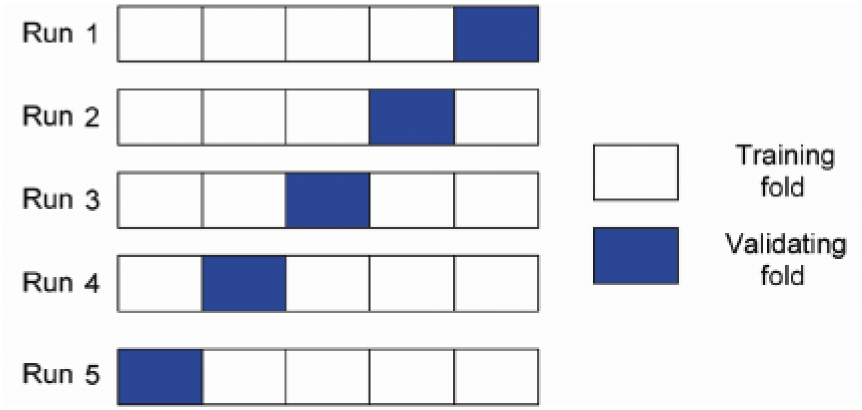


Cross Validation

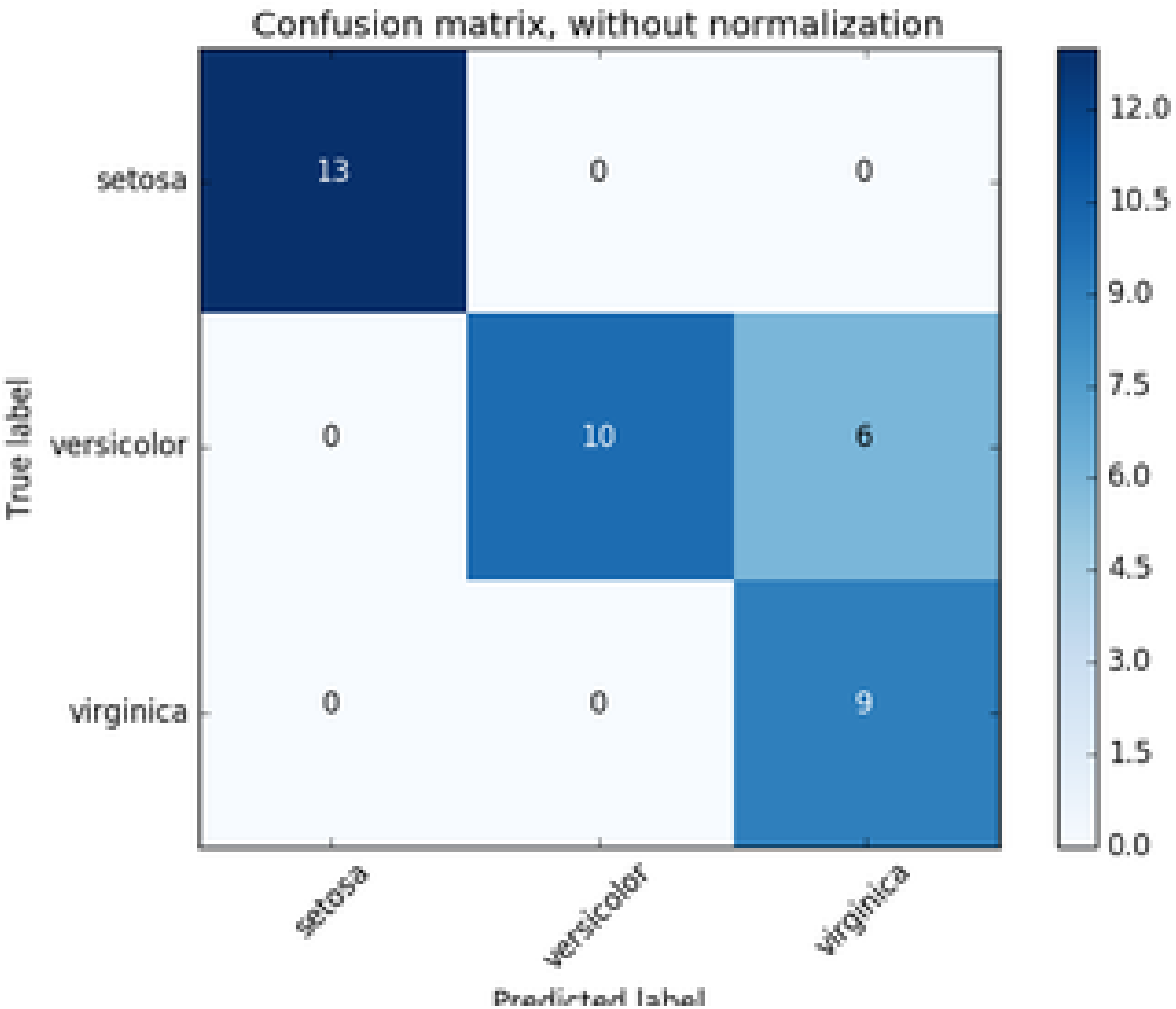
- %split (80 - 20 split)



- K - Fold Cross Validation (5 fold)

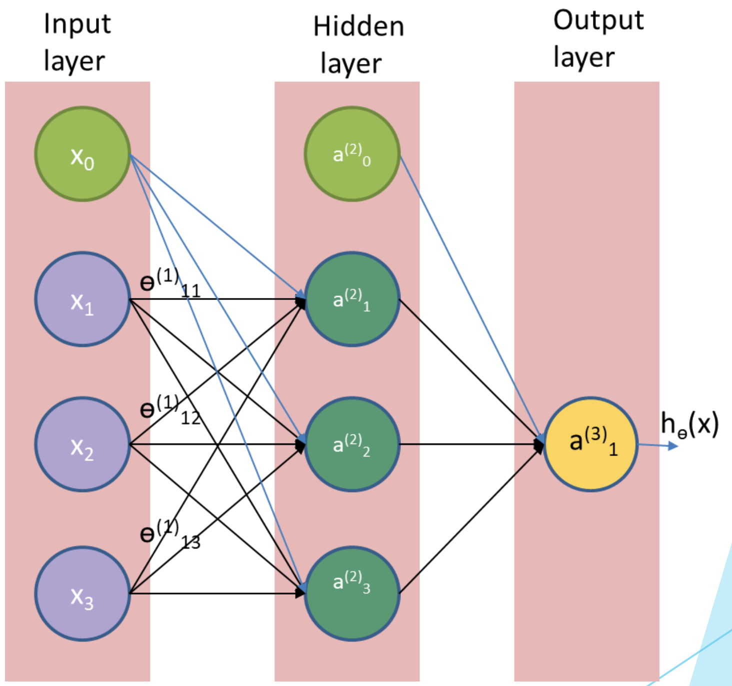


Confusion Matrix



Deep Learning

- Learning algorithm inspired from the biological nervous system
- Ability to learn and make decisions with the help of data and labels
- Fast and Scalable
- In theory, capable of learning any decision making skill
- No need to hard code
- More and more hidden layers -> deeper network (Deep Learning)
- Not much importance to data preprocessing
- Number of input neurons is equal to number of features
- Number of output neurons is equal to number of classes
- g(.) is sigmoid activation function



$a^{(2)}_1 = g((\theta^{(1)}_1)^T X)$

where,

$$\theta^{(1)}_1 = \begin{bmatrix} \theta^{(1)}_{11} \\ \theta^{(1)}_{12} \\ \theta^{(1)}_{13} \end{bmatrix}$$

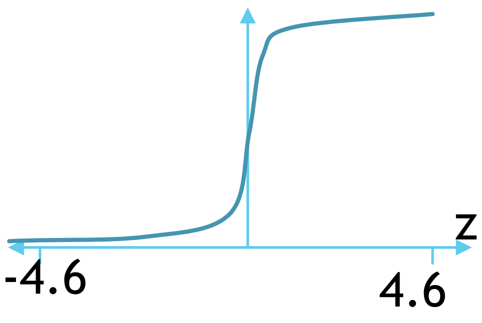
$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$g(z) = \left(\frac{1}{1+e^{-z}}\right)$$

Simple Operations using ANN

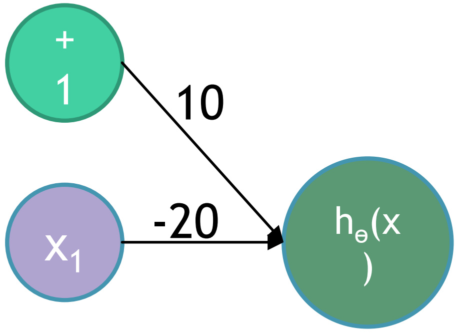
Sigmoid Activation Function

$g(z)=(\frac{1}{1+e^{-z}})$



- Same function used in logistic regression
- When z is beyond the range [-4.6,4.6] it is approximately equal to zero or one
- Other activation functions include Relu, tan, Leaky Relu

NOT

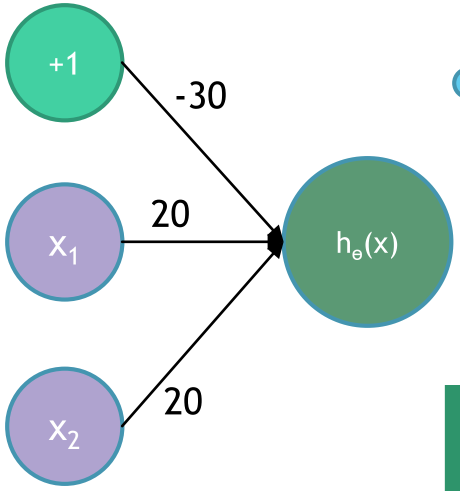


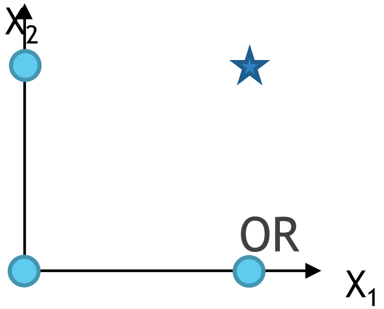
PREDICTION

x_1	$h_\theta(x)$
0	≈ 1
1	≈ 0

Simple Operations using ANN

AND

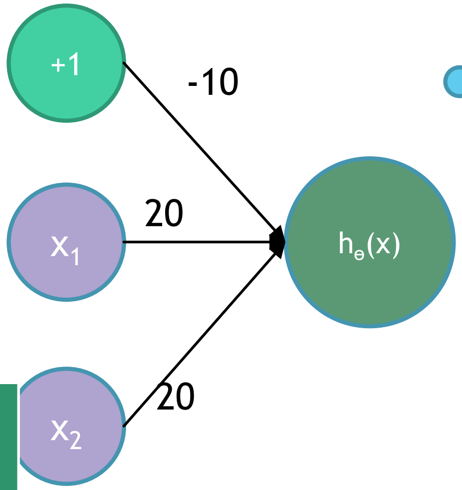


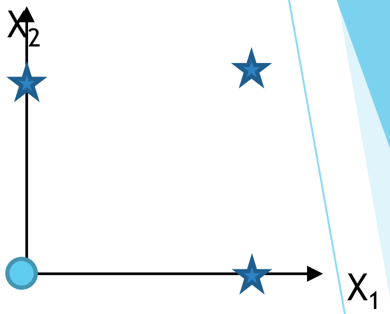


PREDICTION

x_1	x_2	$h_\Theta(x)$
0	0	≈0
0	1	≈0
1	0	≈0
1	1	≈1

OR





PREDICTION

x_1	x_2	$h_\Theta(x)$
0	0	≈0
0	1	≈1
1	0	≈1
1	1	≈1

Deep Learning Techniques

- Signals:
  - Multi layer perceptron
- Images:
  - Convolutional neural networks
    - LeNet
    - ResNet
    - GoogLeNet
    - AlexNet
  - U-Net
  - Y-Net
- Text:
  - BERT
  - ELMO
  - ROBERTA
  - GPT-3

Now, let us build some ML models..

Thank you ☺