Artificial Intelligence - An Overview

NYP NANYANG THE INNOVATIVE POLYTECHNIC

School of Information Technology

MINDEF SINGAPORE

What are we covering

- Introduction to Artificial Intelligence
- Approaches to AI
- AI Playground Online
- Types of Machine Learning

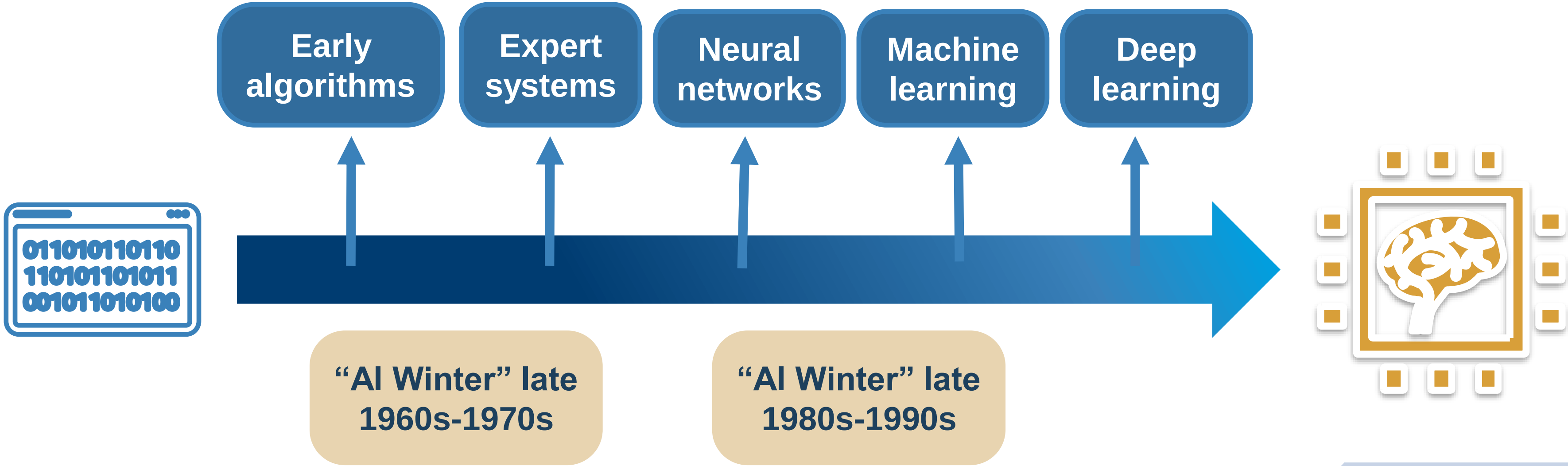
What is Artificial Intelligence

AI Programs are programs that can mimic what human can do and historically what computer could not do very well
(by AI Singapore)

AI is around us

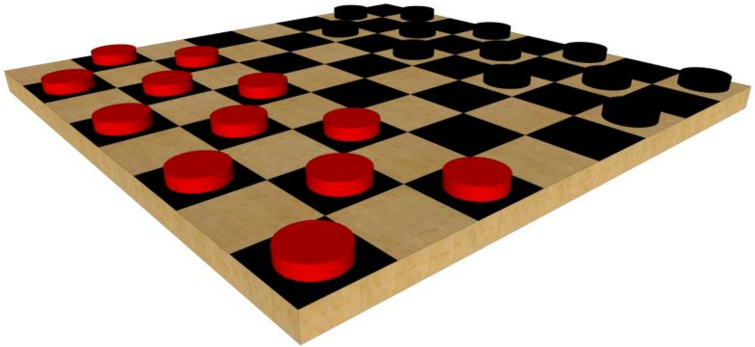
History of AI

AI has experienced several hype cycles, where it has oscillated between periods of excitement and disappointment.



1950S: EARLY AI

- 1950: Alan Turing developed the Turing test to test a machines ability to exhibit intelligent behavior.
- 1956: Artificial Intelligence was accepted as a field at the Dartmouth Conference.
- 1957: Frank Rosenblatt invented the perceptron algorithm. This was the precursor to modern neural networks.
- 1959: Arthur Samuel published an algorithm for a checkers program using machine learning.



The First "AI Winter"

- 1966: ALPAC committee evaluated AI techniques for machine translation and determined there was little yield from the investment.
- 1969: Marvin Minsky published a book on the limitations of the Perceptron algorithm which slowed research in neural networks.
- 1973: The Lighthill report highlights AI's failure to live up to promises.
- The two reports led to cuts in government funding for AI research leading to the first "AI Winter."

John R. Pierce, head of ALPAC

1980’s AI Boom

- Expert Systems - systems with programmed rules designed to mimic human experts.
- Ran on mainframe computers with specialized programming languages (e.g. LISP).
- Were the first widely-used AI technology, with two-thirds of "Fortune 500" companies using them at their peak.
- 1986: The “Backpropogation” algorithm is able to train multi-layer perceptrons leading to new successes and interest in neural network research.

Early expert systems machine

Another AI Winter (late 1980's – early 1990s)

- Expert systems' progress on solving business problems slowed.
- Expert systems began to be melded into software suites of general business applications (e.g. SAP, Oracle) that could run on PCs instead of mainframes.
- Neural networks didn't scale to large problems.
- Interest in AI in business declined.

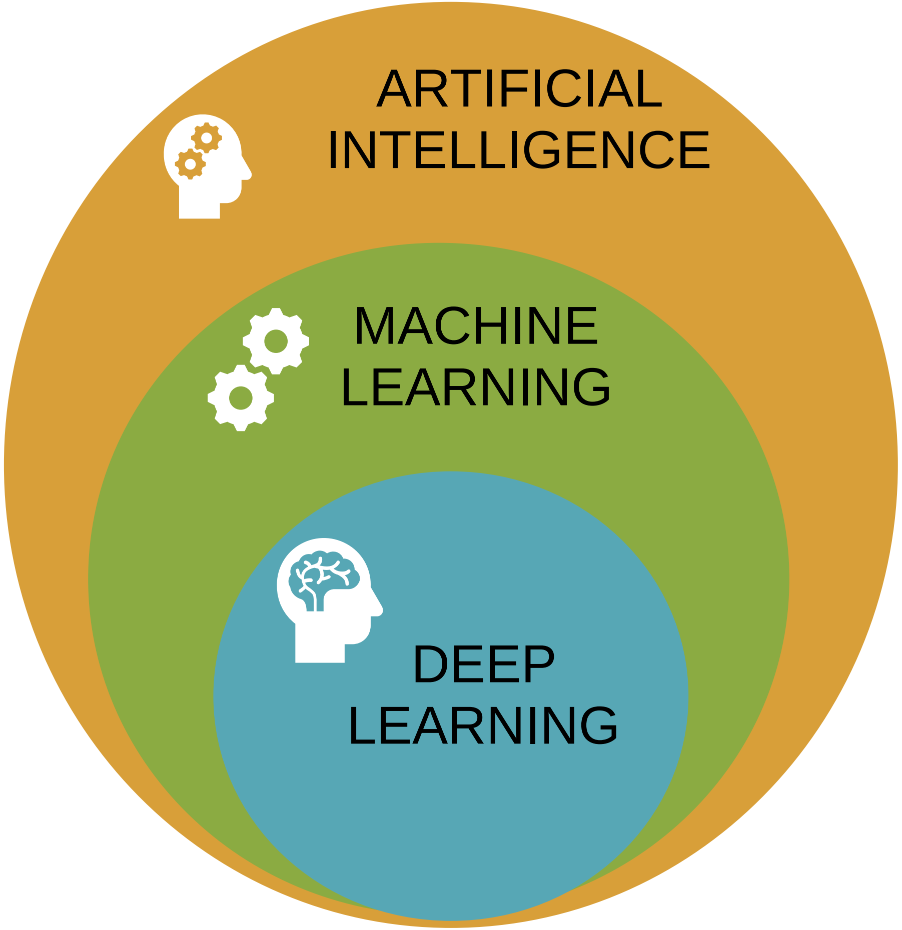
Late 1990's to early 2000's: Classical Machine Learning

- Advancements in the SVM algorithm led to it becoming the machine learning method of choice.
- AI solutions had successes in speech recognition, medical diagnosis, robotics, and many other areas.
- AI algorithms were integrated into larger systems and became useful throughout industry.
- The Deep Blue chess system beat world chess champion Garry Kasparov.
- Google search engine launched using artificial intelligence technology.

IBM supercomputer

2006: Rise of Deep Learning

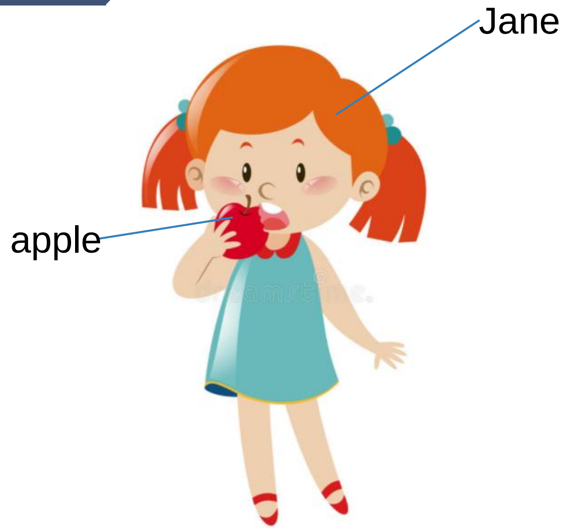
- 2006: Geoffrey Hinton publishes a paper on unsupervised pre-training that allowed deeper neural networks to be trained.
- Neural networks are rebranded to deep learning.
- 2009: The ImageNet database of human-tagged images is presented at the CVPR conference.
- 2010: Algorithms compete on several visual recognition tasks at the first ImageNet competition.


ARTIFICIAL INTELLIGENCE
MACHINE LEARNING
DEEP LEARNING

Approaches to AI

Approaches to AI

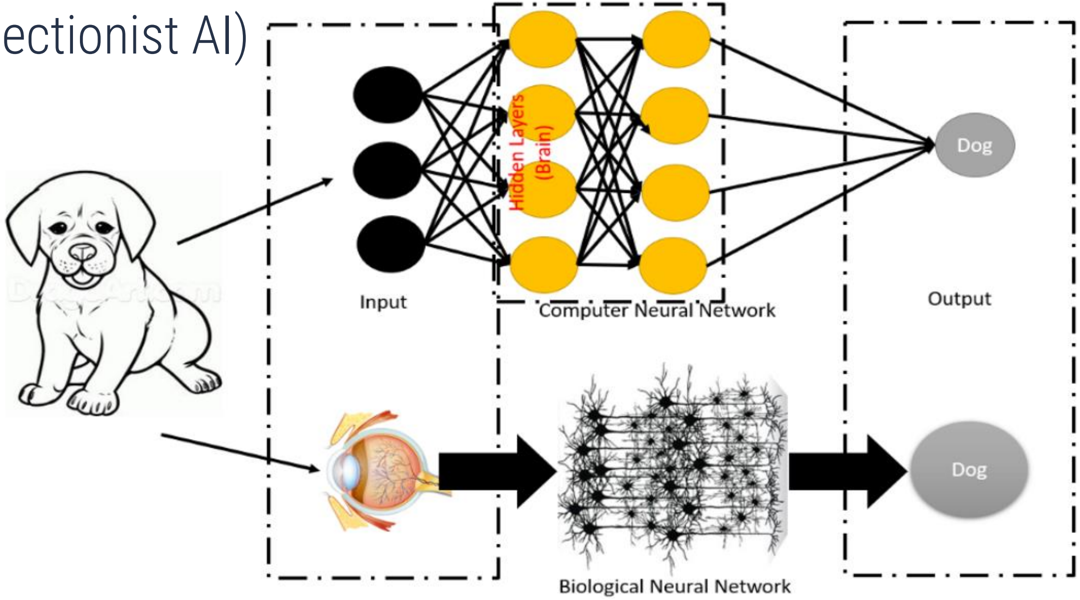
- Symbolic AI (aka Good Old Fashion AI)
  - No Massive Amount of data
  - No training
  - Represent problems using symbols
  - Uses logic as problem solving technique



eat(jane, apple)

Approaches to AI

- Non-Symbolic AI (aka Connectionist AI)
  - No symbol
  - perform calculation
- Examples
  - neural networks
  - deep learning
  - genetic algorithms



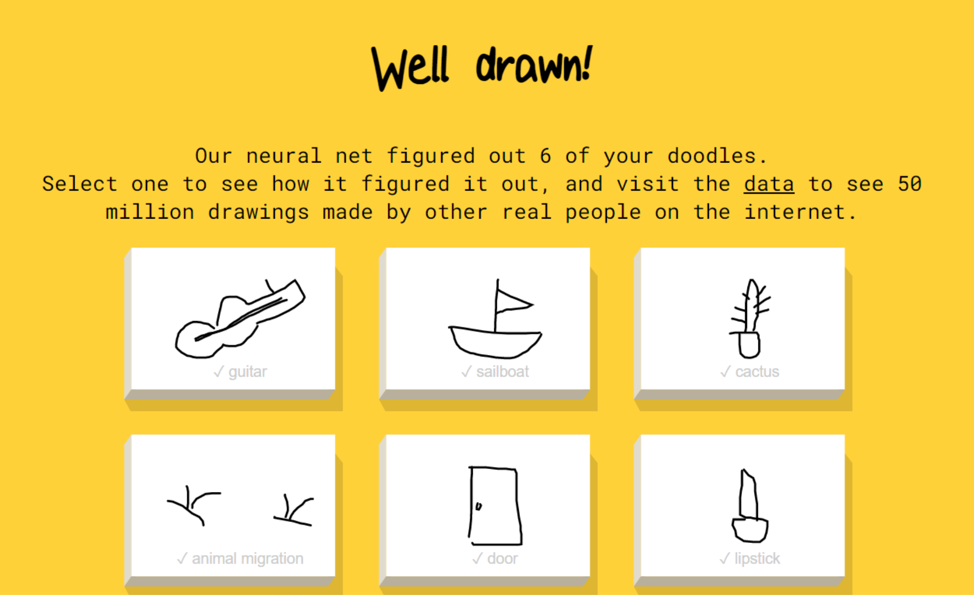
Advantages and Disadvantages of Symbolic AI and Non-symbolic AI

	Symbolic AI	Non-symbolic AI
Advantages	Does not require large amount of data Reasoning process can be easily understood, we can understand how a certain conclusion is reached	Can deal with combinations of attributes such as an image. Noise tolerant
Disadvantages	The rules and knowledge has to be hand coded.	Difficult to understand how the system came to a conclusion. This is particularly important when applied to critical applications such as self-driving cars, medical diagnosis among others

AI Playgrounds Online

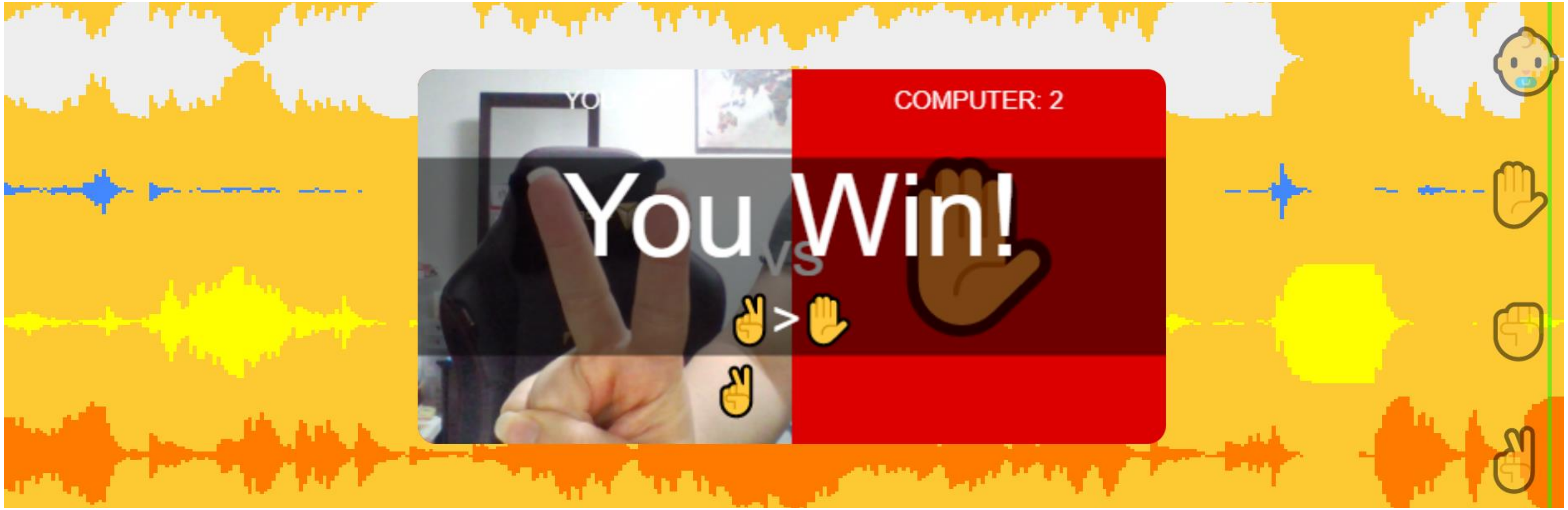
Quick Draw

- https://quickdraw.withgoogle.com/



Rock-Paper-Scissor

- https://tenso.rs/demos/rock-paper-scissors/

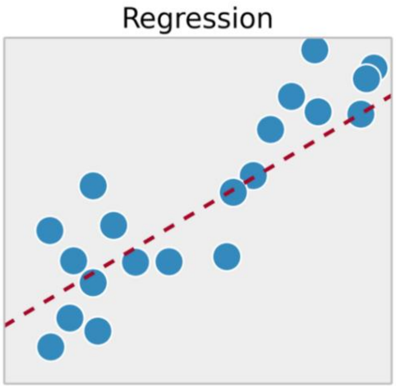


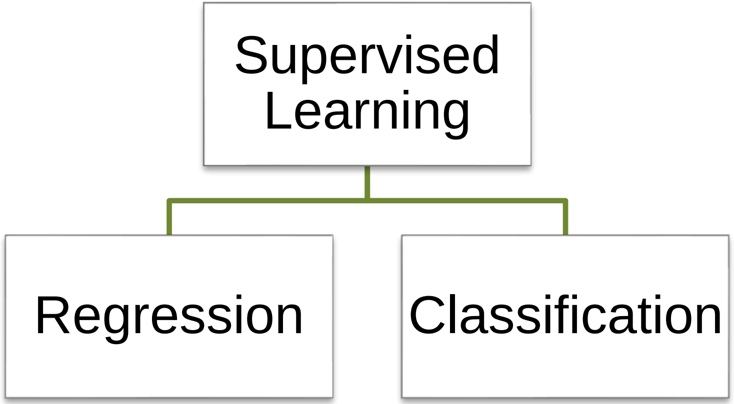
Types of Machine Learning

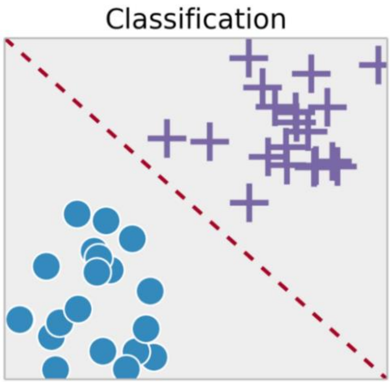
Types of Machine Learning

- Machine Learning
  - Supervised Learning
  - Unsupervised Learning
  - Semi-supervised Learning
  - Reinforcement Learning

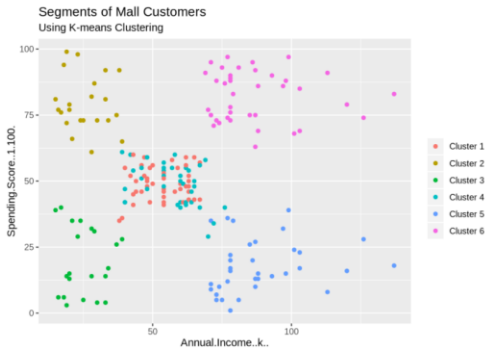
Supervised Learning

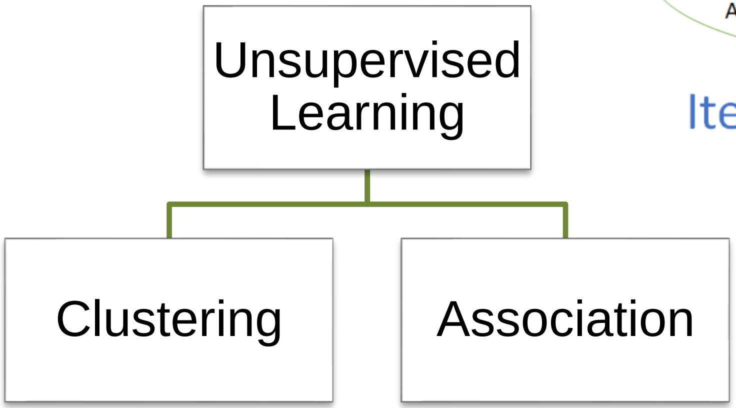


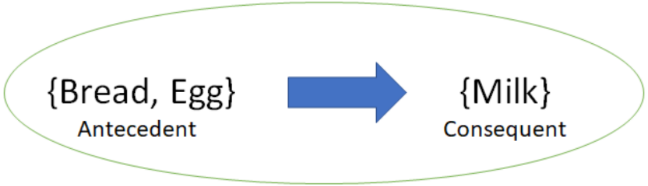




Unsupervised Learning

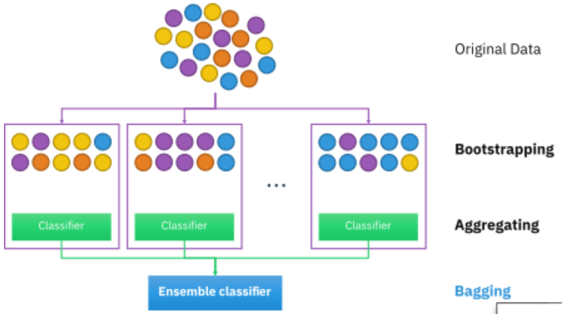






Itemset = {Bread, Egg, Milk}

Semi-supervised Learning

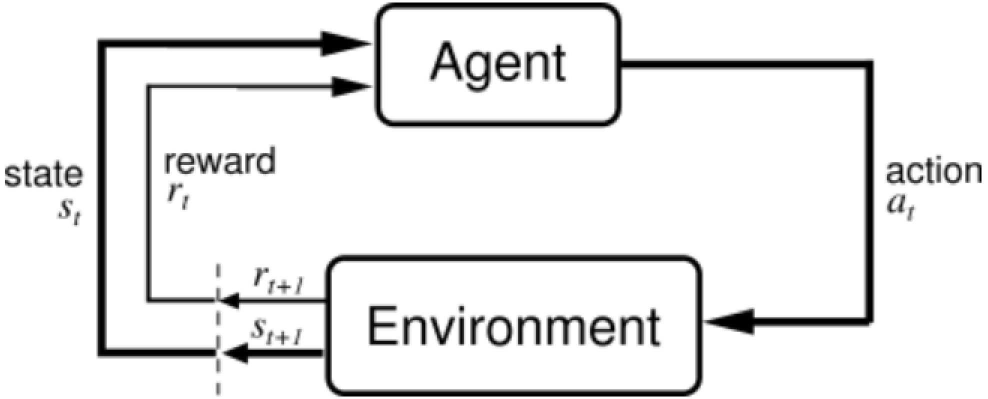


Semi-supervised Learning

Classification

Clustering

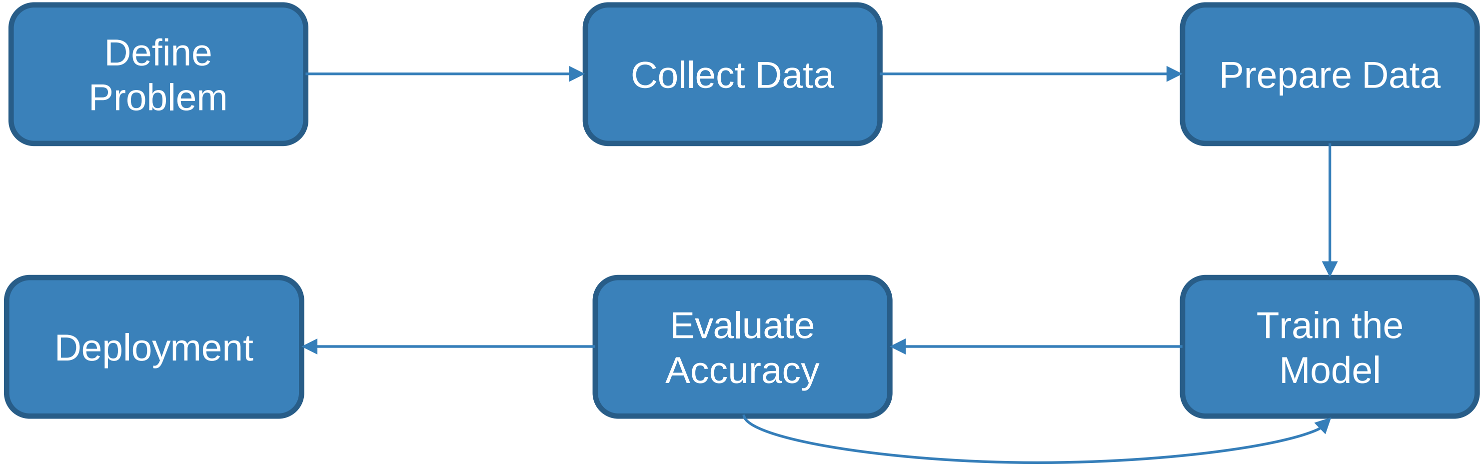
Reinforcement Learning



Deep Reinforcement Learning

Machine Learning Development Workflow

Machine Learning Development Workflow



Define Problem

- Is this picture a cat or a dog? ← Classification
- Detecting credit card fraud ← Anomaly Detection
- Predict the next quarter sales number ← Regression
- Customer segmentation ← Clustering

Collect Data

Data Collection

- Real-time data (IoT System)
- Data collection via forms
- Public dataset
  - Kaggle / UCI/ Data Gov SG
- Other data source
  - enterprise process data etc

Data Format

- Data file (e.g. CSV)
- Image files (e.g. jpg)
- Database
- etc

Prepare Data

Problems with data collected

- Missing data
- Noisy data
- Inconsistent data
- Unstructured text data

Prepare Data

Types of Data

- Numeric (income, age, etc)
- Categorical (gender, nationality, etc)
- Ordinal (Low, Medium High)

Pre-Process Method

- Conversion of data
- Ignore the missing data
- Filling in the missing data
- Outliers detection

Train the model (Supervised Learning)

Classification

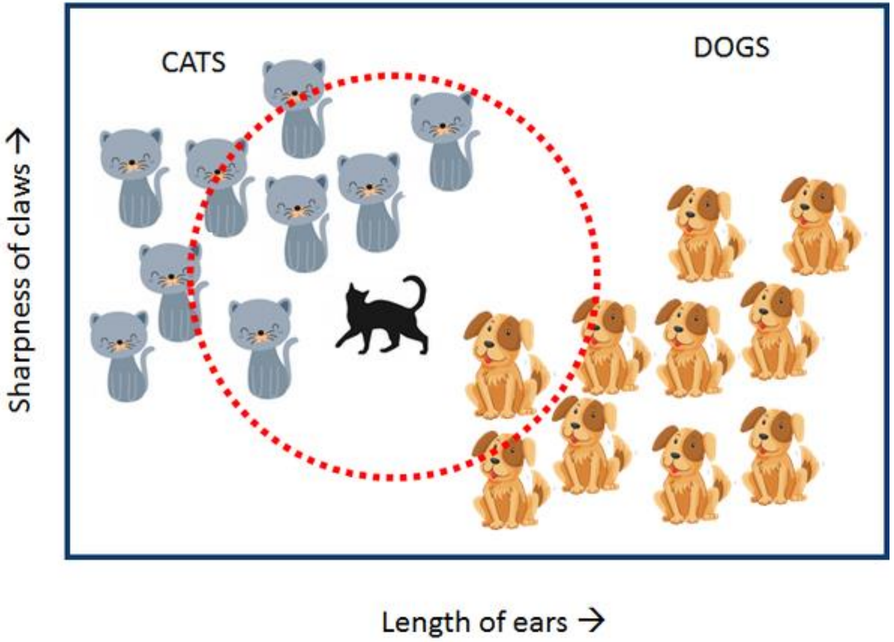
- K-Nearest Neighbour (KNN) *
- Naive Bayes *
- Decision Trees/Random Forest
- Support Vector Machine
- Logistic Regression

Regression

- Linear Regression *
- Support Vector Regression
- Decision Tress/Random Forest
- Gaussian Progresses Regression

K-Nearest Neighbour

Birds of a feather flock together



- Compute the distance between the unknown image and all the images
- Choose the nearest k images (in this example k = 5) by using the 5 shortest distance away
- Check how many cats and how dogs are there in this 5 of them.
- If there are more cats, the unknown image will be classify as a cat
- If there are more dogs, the unknown image will be classify as a dog

Naive Bayes

- Probabilistic machine learning model
- Bayes theorem

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

- A : hypothesis
- B : evidence

Day	outlook	temperature	play
1	sunny	hot	no
2	sunny	hot	no
3	overcast	hot	yes
4	rainy	mild	yes
5	rainy	cool	yes
6	rainy	cool	no
7	overcast	cool	yes
8	sunny	mild	no
9	sunny	cool	yes
10	rainy	mild	yes
11	sunny	mild	yes
12	overcast	mild	yes
13	overcast	hot	yes
14	rainy	mild	no

P(Play = Yes) = 9/14
P(Play = No) = 5/14

Today is a sunny & cool day. Play golf??

NAIVE BAYES EXAMPLE

Today is a sunny & cool day.
Play golf??

outlook	Yes	No	P(Yes)	P(No)	P
sunny	2	3	2/9	3/5	5/14
overcast	4	0	4/9	0/5	4/14
rainy	3	2	3/9	2/5	5/14

temperature	Yes	No	P(Yes)	P(No)	P
hot	2	2	2/9	2/5	4/14
mild	4	2	4/9	2/5	6/14
cool	3	1	3/9	1/5	4/14

Probability that we can play golf

- **P(Outlook = Sunny | Play = Yes) = 2/9**
- **P(Temperature = Cool | Play = Yes) = 3/9**
- **P(Play = Yes) = 9/14**

Probability that we cannot play golf

- **P(Outlook = Sunny | Play = No) = 3/5**
- **P(Temperature = Cool | Play = No) = 1/5**
- **P(Play = No) = 5/14**

P(sunny, cool) = 5/14 x 4/14 = 0.102

Example

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

Probability that we can play golf

- P(Outlook = Sunny | Play = Yes) = 2/9
- P(Temperature = Cool | Play = Yes) = 3/9
- P(Play = Yes) = 9/14

Probability that we cannot play golf

- P(Outlook = Sunny | Play = No) = 3/5
- P(Temperature = Cool | Play = No) = 1/5
- P(Play = No) = 5/14

P(sunny, cool)
= 5/14 x 4/14 = 0.102

P(Can Play Golf | Sunny, Cool)
= P(Sunny|**Yes**) x P(Cool|**Yes**) x P(Play = **Yes**) / P(Sunny, Cool)
= (2/9 x 3/9 x 9/14) / 0.102 = 0.4669

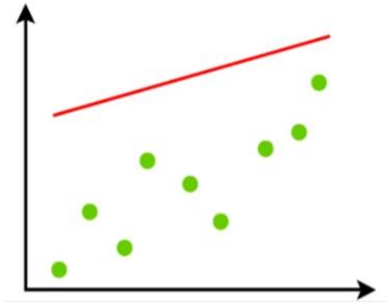
P(Cannot Play Golf | Sunny, Cool)
= P(Sunny|**No**) P(Cool|**No**) P(Play = **No**) / P(Sunny, Cool)
= (3/5 x 1/5 x 5/14) / 0.102 = 0.4202

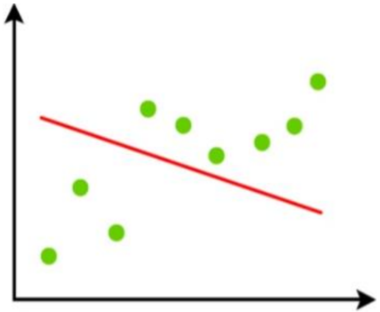
Answer is YES!

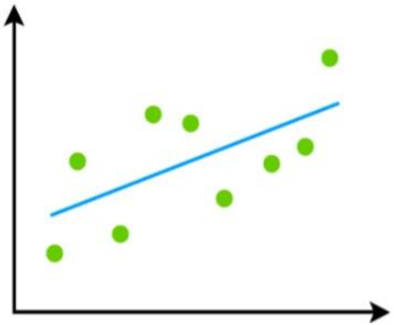
Linear Regression

Finding the best fit line:

The best fit line can be found by minimizing the distance between all the data points and the distance to the regression line. Ways to minimize this distance are sum of squared errors, sum of absolute errors etc.







Choose the Algorithm to train the model (UnSupervised Learning)

Clustering

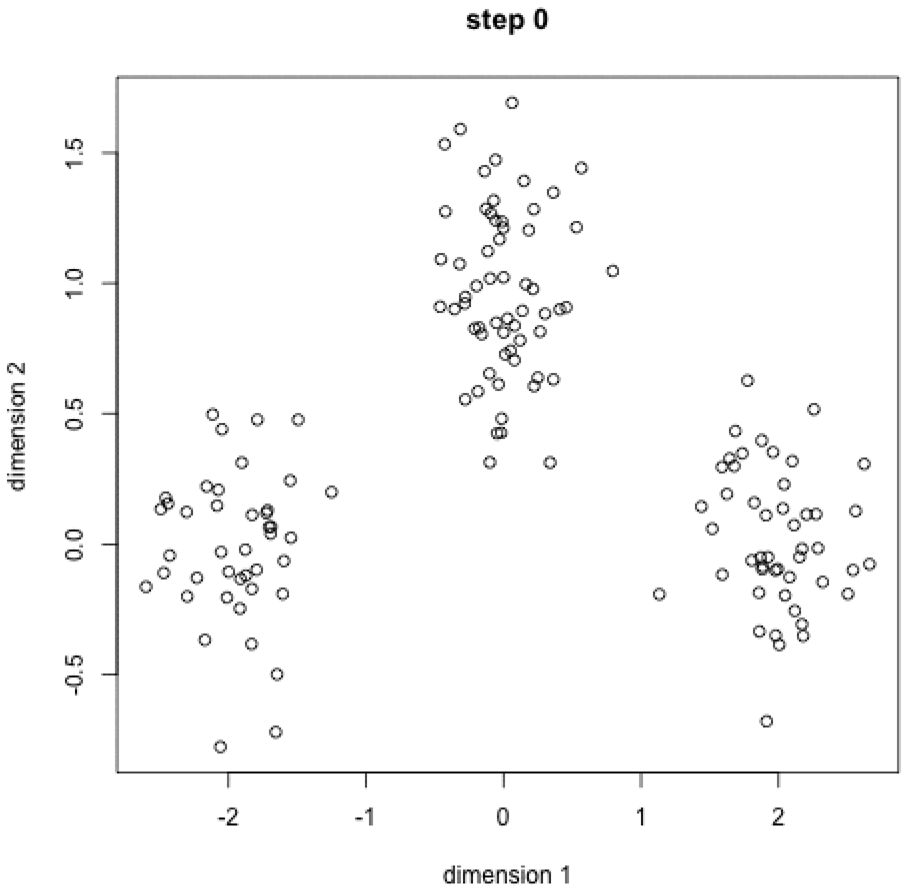
- K-Means *
- Hierarchical Clustering
- Anomaly Detection
- Gaussian mixtures

Association

- Apriori *

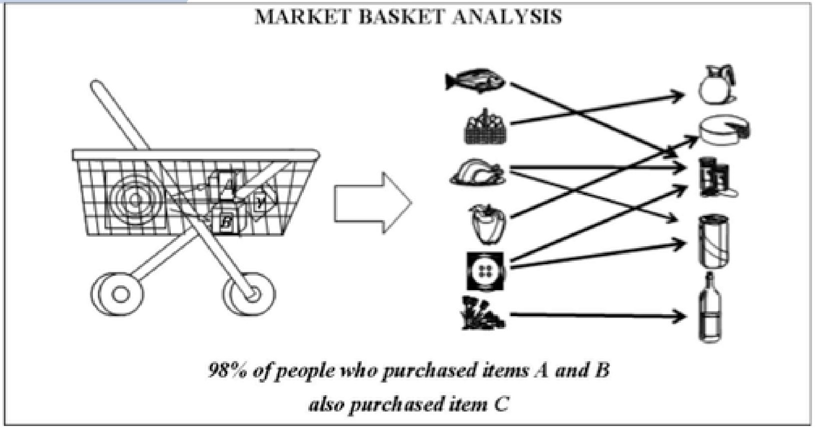
K-Means

- Select k centroid
- Randomly initialise their respective centroid
- Go through each data points to classify it by computing the distance between that point and each centroid. Choose the centroid that is closest
- Recompute the centroid by taking the means of all the vectors in the group.
- Repeat for a few iterations of until the centroid don't change much



Apriori

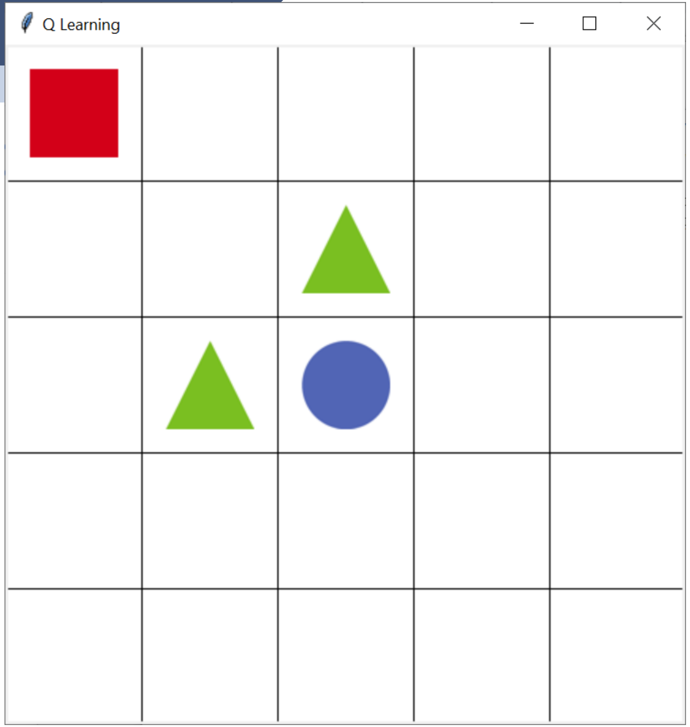
- Market Basket Analysis
- Recommender System
- Retail Store Planning



Reinforcement Learning

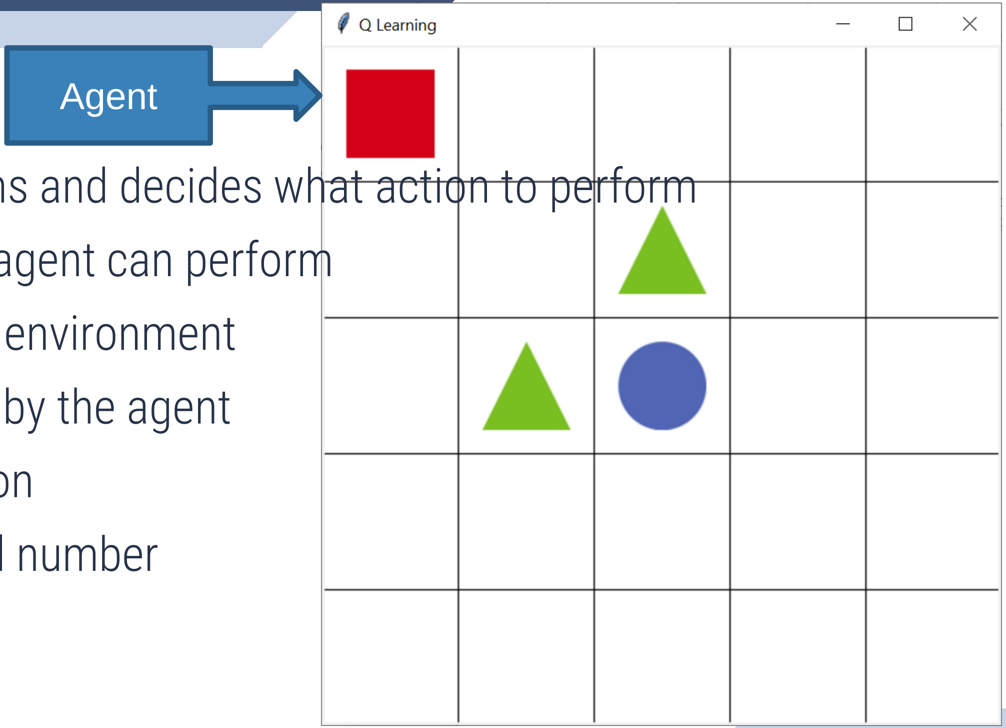
- Blue Circle = Win
- Green Triangle = Lose

- Win = Reward
- Lose = Punishment (negative reward)



Reinforcement Learning – Key Terminologies

- Agent – learner/decision maker
- Environment – the place agent learns and decides what action to perform
- Action – the set of actions that the agent can perform
- State – the state of the agent in the environment
- Rewards – for each action selected by the agent
- Policy – the decision-making function
- Value – mapping each states to real number



Deep Reinforcement Learning

AI pilot shoots down F16 Top Gun to win first ever USAF dogfight simulator competition as human pilot says he can't cope with the robot's aggressive tactics and warns 'the things we do as fighter pilots aren't working'

[Watch DARPA's AI vs. Human in Virtual F-16 Aerial Dogfight (FINALS) - YouTube]

Evaluate Accuracy (Classification)

Training Data (70%)	Validation Data (10%)	Testing Data (20%)

- **Training dataset:** the set of data used to fit the model
- **Validation dataset:** the set of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters
- **Test dataset:** The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset

Evaluate Accuracy (Classification)

n=165	Predicted: NO	Predicted: YES
Actual: NO	True Negative 50	False Positive 10
Actual: YES	False Negative 5	True Positive 100

Accuracy

= (True Positives +True Negatives) / (Total number of classification)

= (100 + 50) / 165

= 0.909

Precision = $\frac{True\ Positive}{True\ Positive + False\ \boldsymbol{Positive}}$

= 100/(100+10)

= 0.909

Evaluate Accuracy (Classification)

n=165	Predicted: NO	Predicted: YES
Actual: NO	True Negative 50	False Positive 10
Actual: YES	False Negative 5	True Positive 100

Recall = $\frac{True\ Positive}{True\ Positive + False\ Negative}$

= 100 / (100+5)

= 0.952

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

= 2 $\times$ $\frac{0.909 \times 0.952}{0.909 + 0.952}$

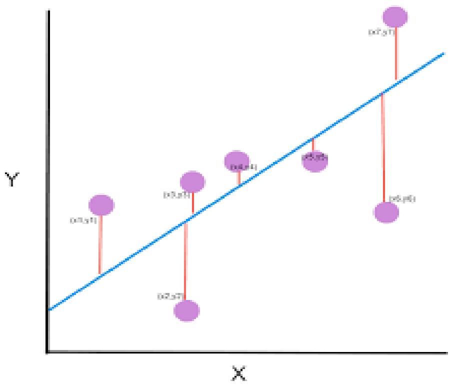
= 0.930

Evaluate Accuracy (Regression)

- Mean Absolute Error
- Mean Square Error
- Root Mean Square Error

Evaluate Accuracy (Regression)

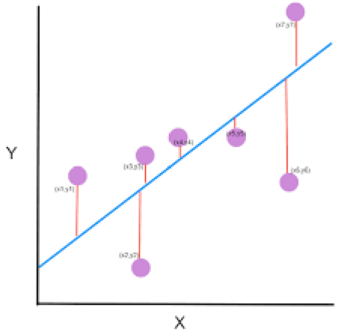
- Mean Absolute Error



$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n} |y_j - \hat{y}_j|$$

Evaluate Accuracy (Regression)

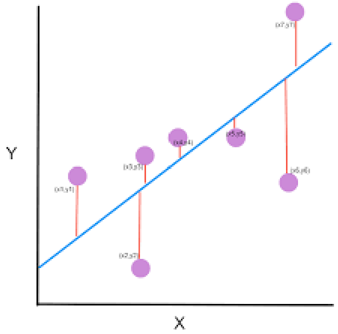
- Mean Square Error



$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

Evaluate Accuracy (Regression)

- Root Mean Square Error



$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Evaluate Accuracy (Regression)

		Model 1: y = 2.7x + 1.5					Model 2: y = 1.2x + 2.3				
X	**Actual Y**	**Model 1 Y**	**Error**	**MAE**	**MSE**	**RMSE**	**Model 2 Y**	**Error**	**MAE**	**MSE**	**RMSE**
0	2	1.5	0.5	0.5	0.25		2.3	-0.3	0.3	0.09	
1	3.7	4.2	-0.5	0.5	0.25		3.5	0.2	0.2	0.04	
2	5.1	6.9	-1.8	1.8	3.24		4.7	0.4	0.4	0.16	
3	7.4	9.6	-2.2	2.2	4.84		5.9	1.5	1.5	2.25	
5	13.4	15	-1.6	1.6	2.56		8.3	5.1	5.1	26.01	
				1.32	**2.228**	**1.493**			**1.5**	**5.71**	**2.39**
				MAE	**MSE**	**RMSE**			**MAE**	**MSE**	**RMSE**

Break?

Workshop

Predictive Modelling

- In this morning, we will be looking at the following predictive modelling
  - Malware Prediction
  - Phishing Prediction

Tools & Datasets

- Tools
  - SIT GPU Cloud
  - Juypter Notebook
  - Python Programming
- Datasets
  - Malware
  - Phishing

Approaches to Malware Detection & Analysis

- Static Analysis : examine the codes without executing the program
- Dynamic Analysis : execute the program in an controlled environment

Malware Prediction

Collect APK data

https://www.unb.ca/cic/datasets/andmal2017.html

Disassembler

Decompile APK files to get the following ->

Feature Extraction

Files after decompilation

- Lib
- Res
- Assets
- classes.dex
- resource.arsc
- **AndroidManifest.xml**

Malware - Feature

- android.permission.ACCESS_ALL_DOWNLOADS
- android.permission.ACCESS_BLUETOOTH\
- android.permission.ACCESS_CACHE_FILESYSTEM
- android.permission.WRITE_EXTERNAL_STORAGE

- And the list goes on in our lab1 example with ~1000 features

Phishing Prediction - Features

- Iframe
- PopUpWindow
- double_slash_redirecting
- having_IP_Address
- having_@_Symbol
- etc

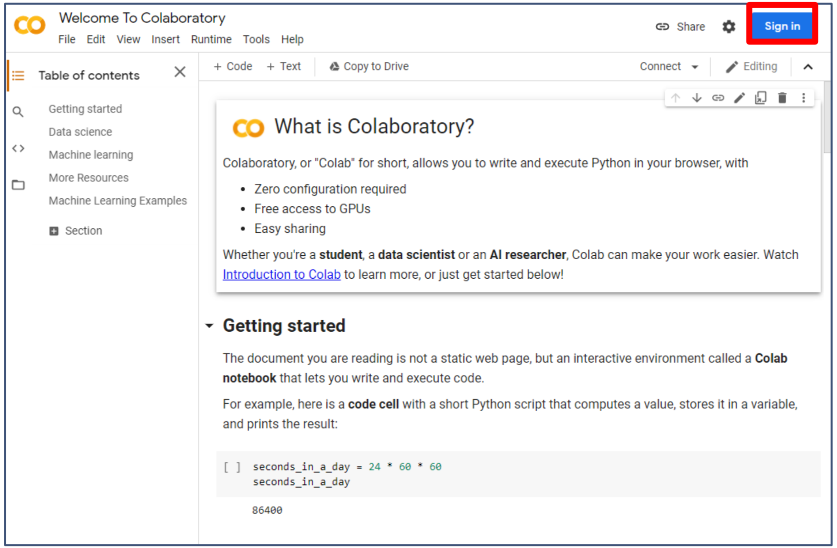
https://www.kaggle.com/akashkr/phishing-website-dataset https://github.com/alanchow85/AIUP2

https://github.com/nyp-sit/aiup

Let’s get started~

Using Google Colab

- https://colab.research.google.com
- Login using
  - Email: ______________________
  - Password: ___________________



THANK YOU!

Any questions?