neueda

Machine Learning

Introduction to Machine Learning

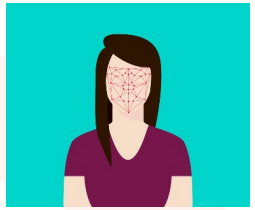
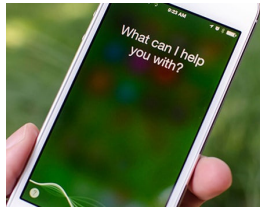
Contents

Introduction

Artificial Intelligence/Machine learning does not only mean robots or Sci-Fi movies!

Machine and deep learning applications are everywhere!

Google search engine, amazon recommender systems, Facebook facial recognition (tagging), Siri


neueda

Introduction

The pace of progress in artificial intelligence (I'm not referring to narrow AI) is incredibly fast. Unless you have direct exposure to groups like Deepmind, you have no idea how fast—it is growing at a pace close to exponential. The risk of something seriously dangerous happening is in the five-year time frame. 10 years at most.

Elon Musk

ChatGPT is incredibly limited, but good enough at some things to create a misleading impression of greatness. it's a mistake to be relying on it for anything important right now. it's a preview of progress; we have lots of work to do on robustness and truthfulness.

Sam Altman, CEO OpenAI. Twitter

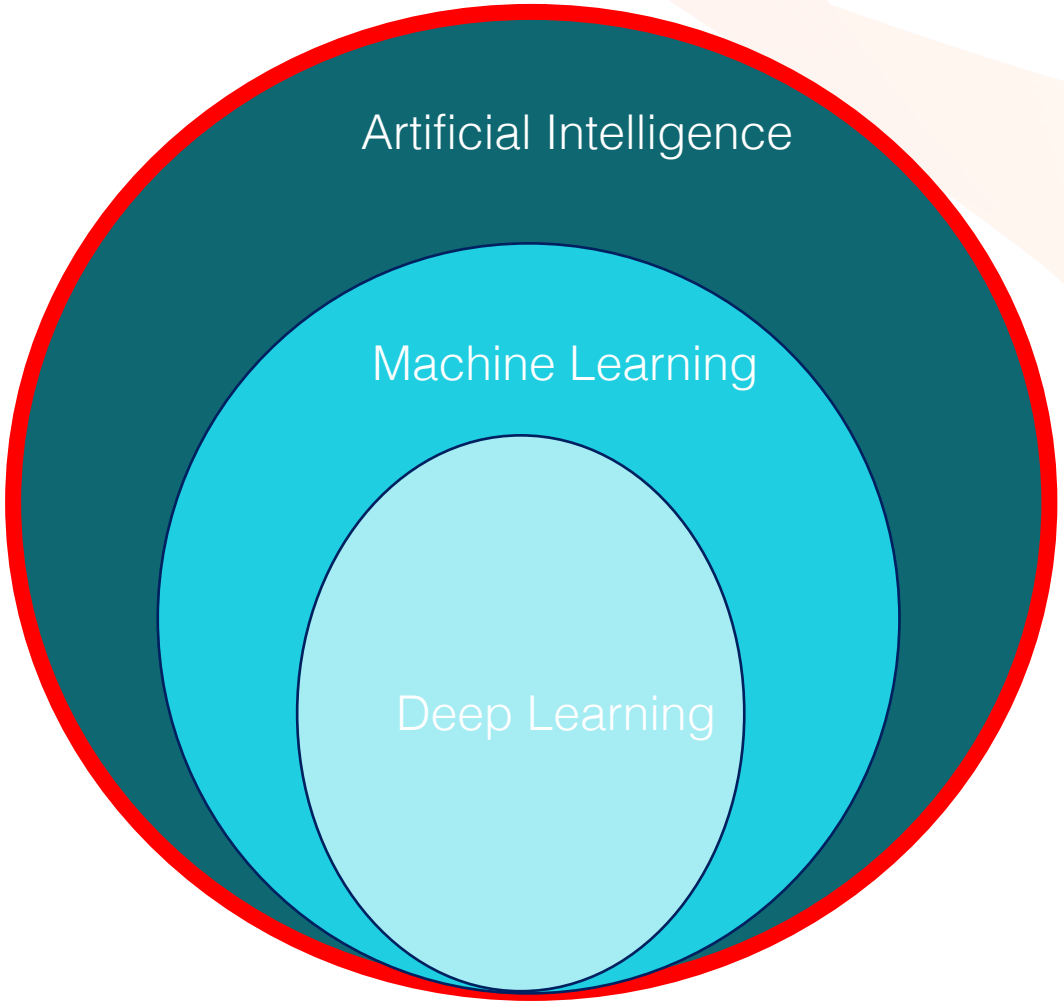
Artificial intelligence is one of the most profound things we're working on as humanity. It is more profound than fire or electricity.

Sundar Pichai CEO Google

Artificial Intelligence

Science that empowers computers to mimic human intelligence

decision making, text processing, and visual perception.



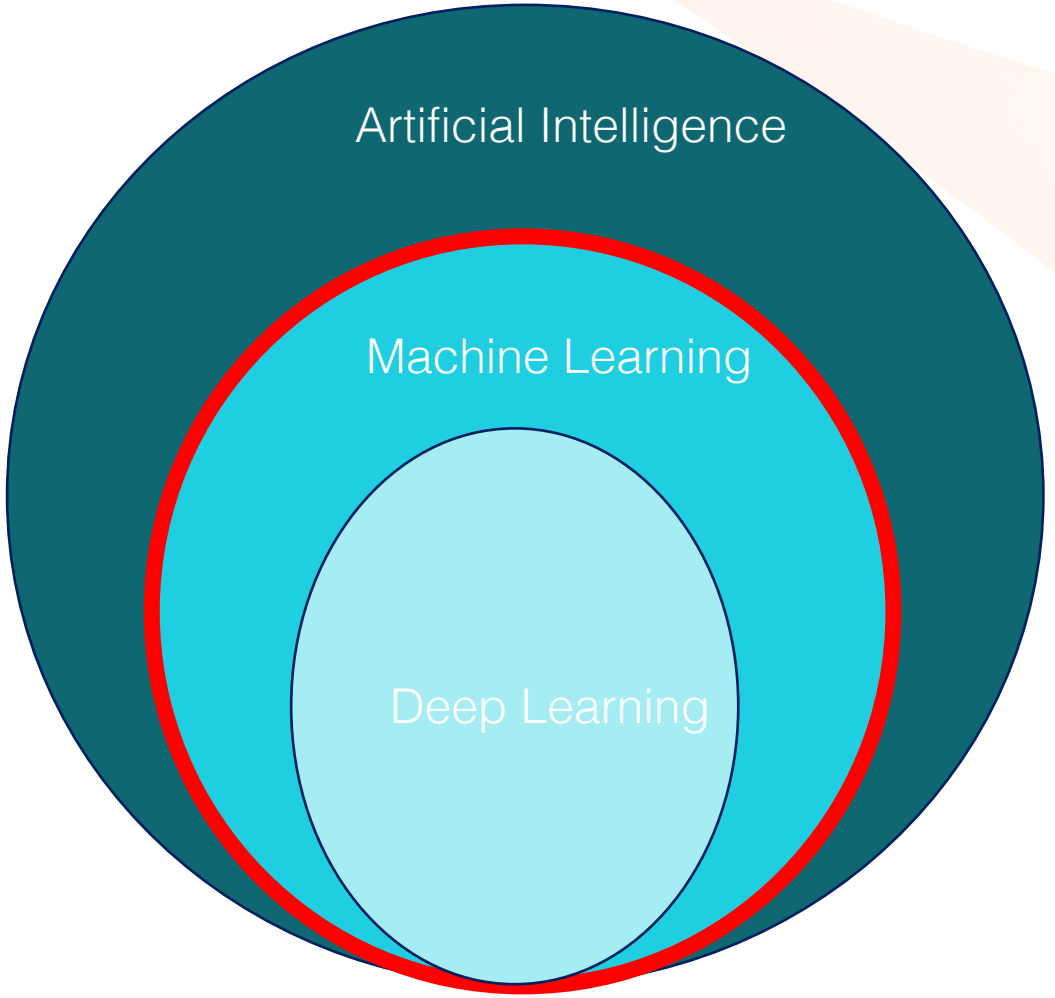
neueda

Machine Learning

A subfield of Artificial Intelligence that enables machines to improve at a given task with experience.

All machine learning techniques are classified as Artificial Intelligence but not all Artificial Intelligence could count as Machine Learning.

e.g. some basic Rule-based engines could be classified as AI but they do not learn from experience therefore they do not belong to the machine learning category.

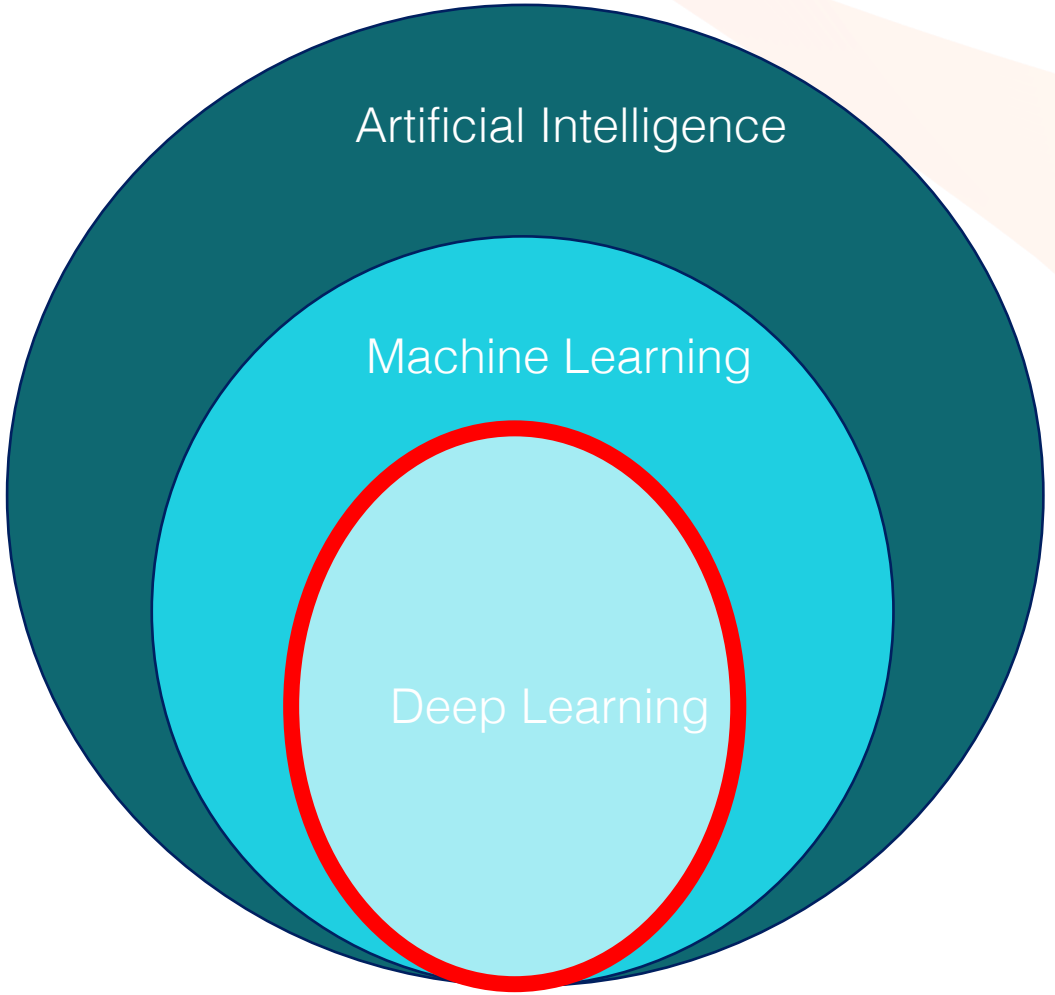


Deep Learning

A specialized field of Machine Learning that relies on training of Deep Artificial Neural Networks (ANNs) using large dataset such as images.

ANNs are information processing models inspired by the human brain.

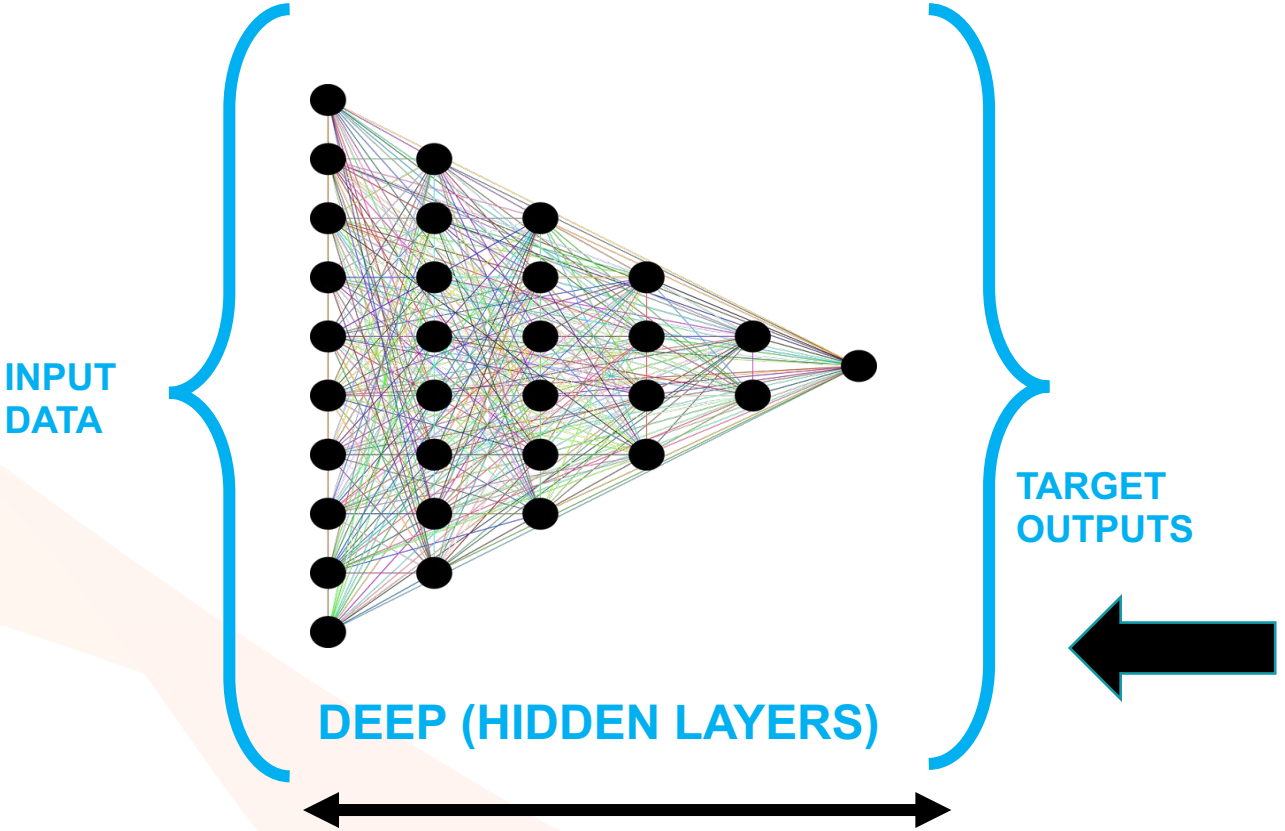
The human brain consists of billions of neurons that communicate to each other using electrical and chemical signals and enable humans to see, feel, and make decision.



Deep Learning

Depth is a measure of how many hidden layers

3 or more layers (including input and output) qualifies as "**deep**" learning.



Fully Connected

Each neuron sends its output to every neuron in the next layer.

Feed Forward

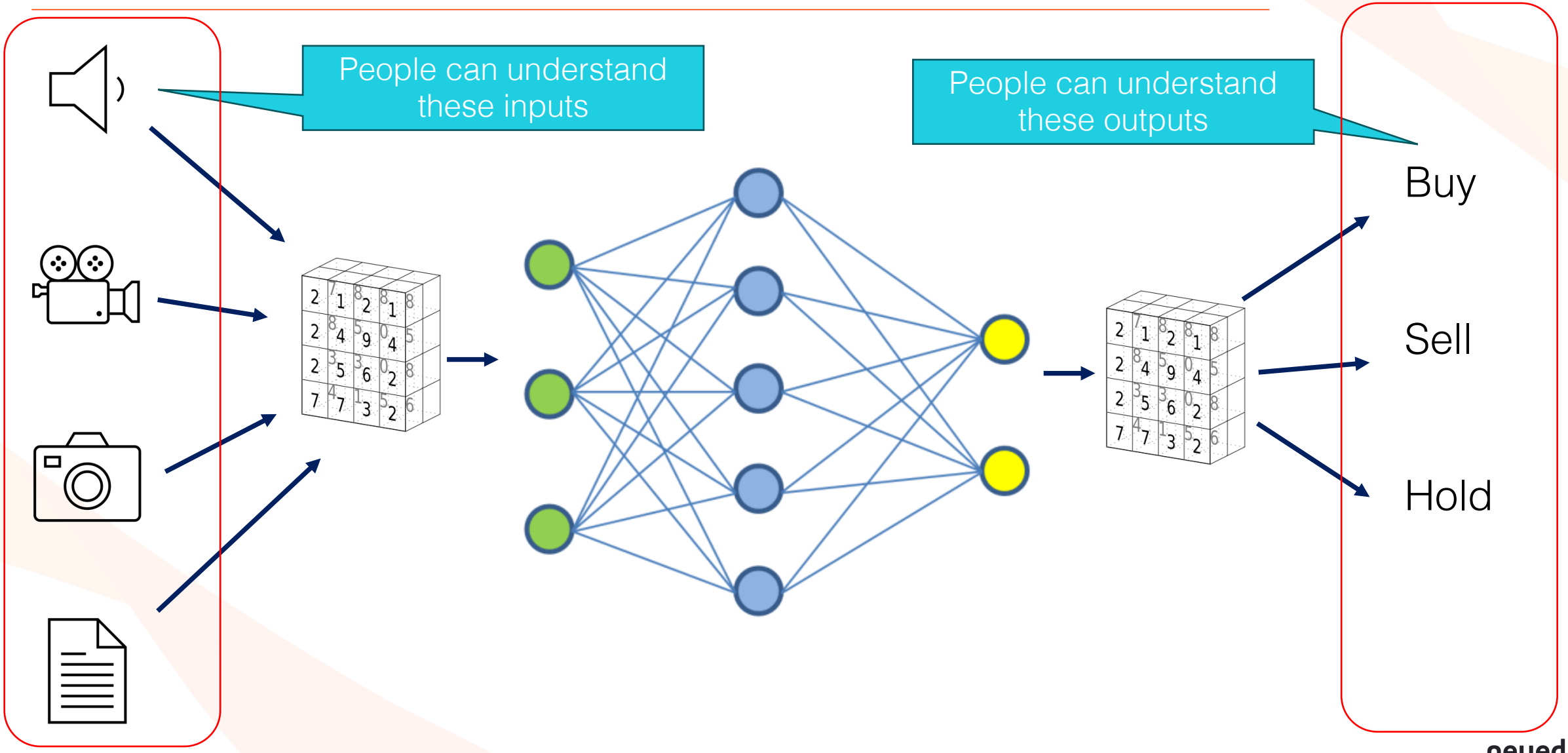
Information flows from input through to output layer

There are no feedback loops

Feed Forward Fully Connected Artificial Neural Network (ANN)

There are many variations of ANN

Deep Learning



Machine Learning vs Deep Learning

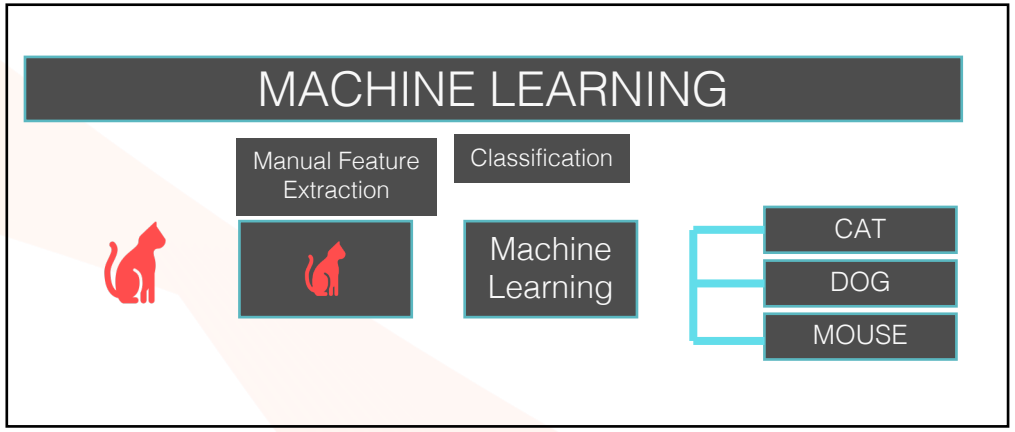
Deep learning can perform automatic feature extraction
Machine Learning cannot

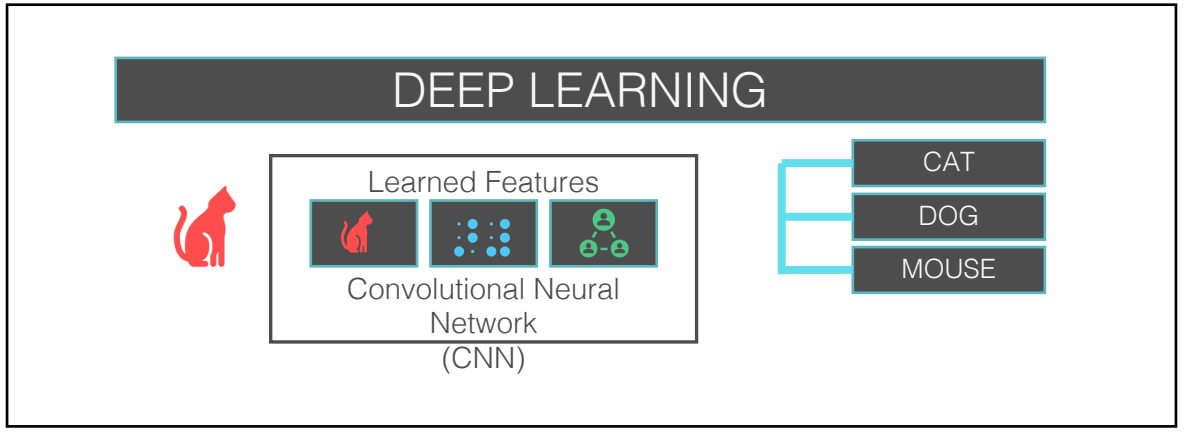
Machine learning Process:

(1) select the model to train,
(2) manually perform feature extraction.

Deep Learning Process:

(1) select the architecture of the network,
(2) features are automatically extracted by feeding in the training data (such as images) along with the target class (label).





neueda

Categories of Machine Learning

Do people help the machines to learn ? How much help do people give the machines ?

- Supervised
- Unsupervised
- **Semi supervised****
- Reinforcement

Can the machine learn incrementally on the fly ?

- Online
- Batch

Do they compare data to known data or detect patterns in data ?

- Instance-based
- Model-based

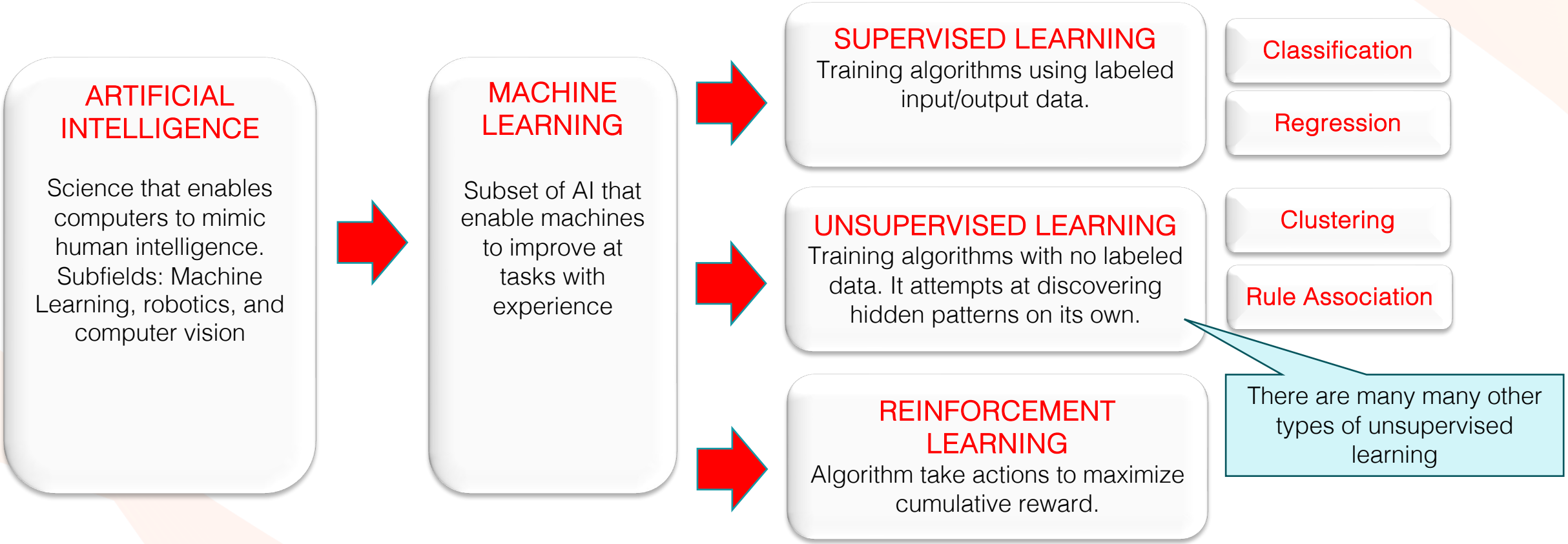
These may be combined in a variety of ways

An ML application might
- Learn on the fly
- Use a deep neural network
- Trained using example data

This would make it online, model-based & supervised

Semi supervised**
Partially labelled data – some labelled, some not
Usually, combination of supervised and unsupervised algorithms

Types of Learning



Supervised Learning

Training data contains the answers - **labeled**

Classification

Predict the **class** of an observation

- Is an email ham or spam
- Is an image a dog or a cat
- Is a transaction fraudulent or legitimate

Regression

Predict a **target** numeric value of an observation given a set of **features**

- Price of a house - given location, square footage, number of bedrooms
- Daily Temperature – given time of year, yesterdays temp
- Percentage increase/decrease of a stock

Class – spam/ham
Target – numeric value being predicted
Features – square footage of house, location, yesterday's temperature
Predictors – a set of features, attributes used to predict a target
Attribute – a single feature, e.g., number of bedrooms

Usually, a feature is an attribute and its value

Supervised Learning

Some regression algorithms can be used for classification

Some classification algorithms can be used for regression

Common supervised learning algorithms include

- k-Nearest Neighbours
- Linear Regression
- Logistic Regression
- Support Vector Machines
- Decision Trees and Random Forests
- Neural Networks

Unsupervised Learning

Training data does not contain the answers - **unlabeled**

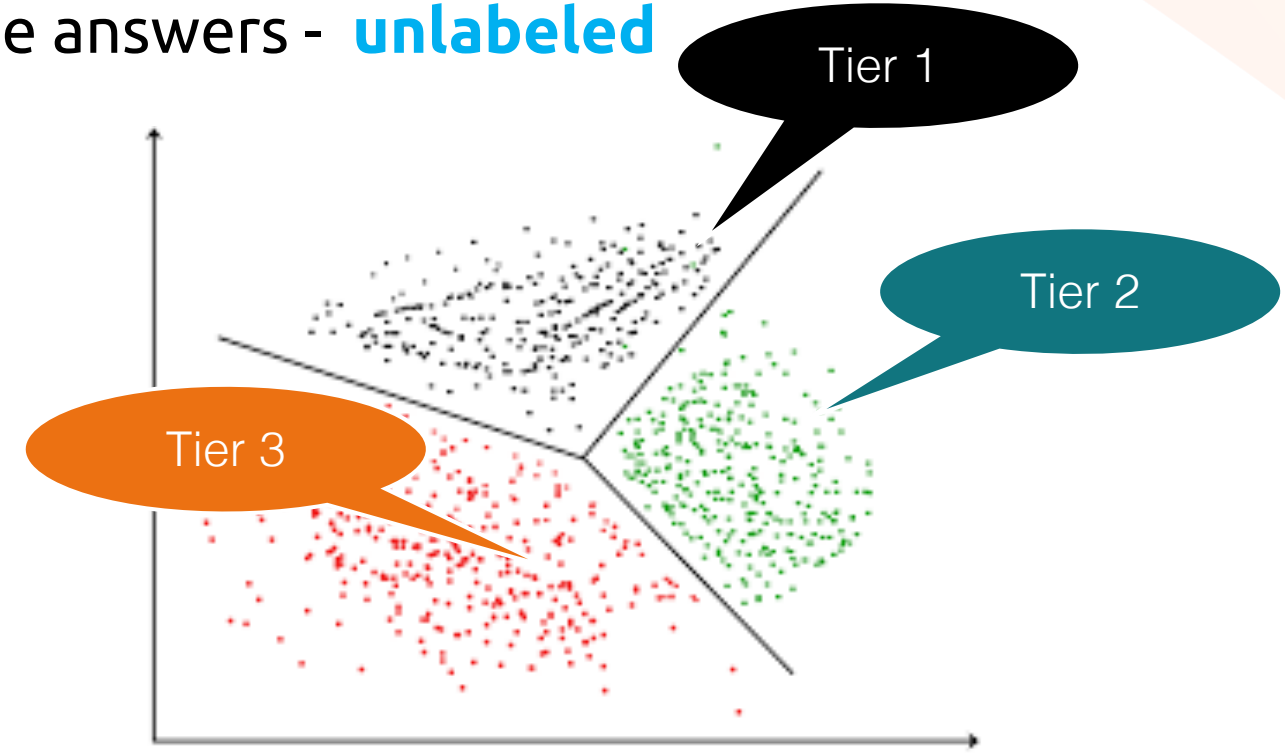
Clustering

Detect groups of similar observations

- K-Means
- Hierarchical Cluster Analysis (HCA)
- Expectation Maximization

e.g.

- 40% of customers are over 65 – Tier1
- 15% are at school – Tier 2
- 45% are working – Tier 3



neueda

Unsupervised Learning

Dimensionality Reduction/ Feature Extraction

- Principal Component Analysis (PCA)
- Kernel PCA
- Locally Linear Embedding (LLE)
- T-distributed Stochastic Neighbour Embedding (t-SNE)

Used to output simplified data without loosing information

Merge several features into one

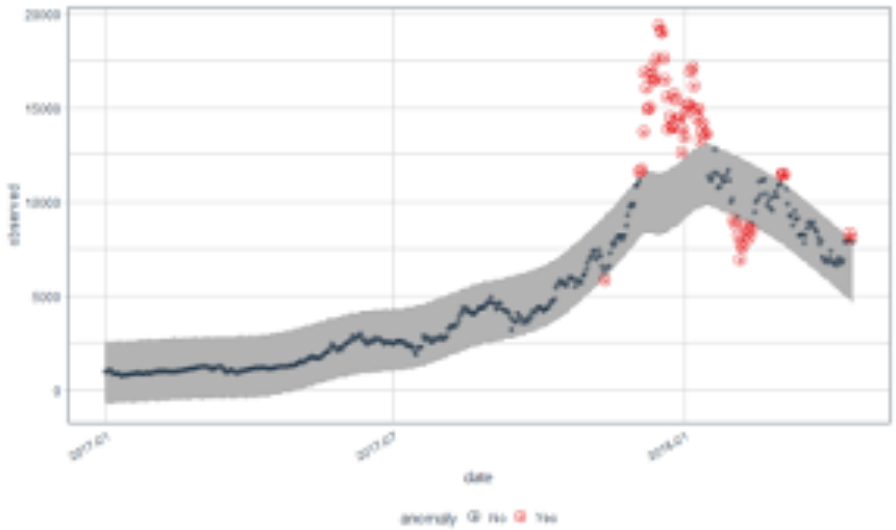
Correlation between cars age and mileage – *wear and tear*

Unsupervised Learning

Anomaly / Outlier Detection

- Principal Component Analysis (PCA)
- Kernel PCA
- Locally Linear Embedding (LLE)
- T-distributed Stochastic Neighbour Embedding (t-SNE)

Credit card fraud
Manufacturing defects
Bad customer review



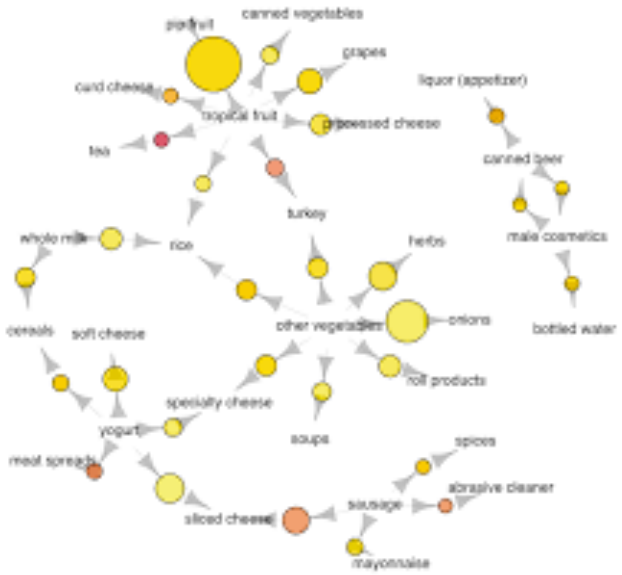
neueda

Unsupervised Learning

Association Rule Learning

- Principal Component Analysis (PCA)
- Kernel PCA
- Locally Linear Embedding (LLE)
- T-distributed Stochastic Neighbour Embedding (t-SNE)

Gain insights into data
- Web site recommendations
- Spell Checkers
- Insurance quotes



neueda

Reinforcement Learning

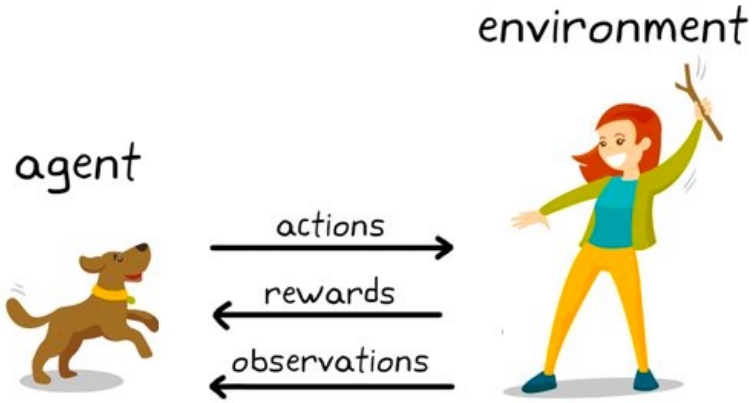
Reinforcement learning allows machines take actions to maximize cumulative reward.

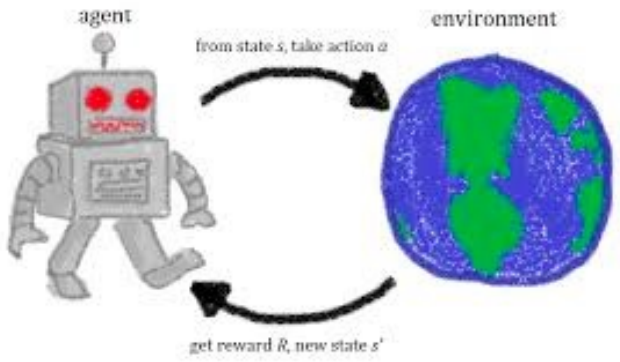
Reinforcement algorithms learn by trial and error through reward and penalty.

Two elements: **environment** and **learning agent**.

The environment rewards the agent for correct actions.

Based on the reward or penalty, agent improves its environment knowledge to make better decision.





Key Words

Trial
&
Error

Batch & Online Learning

Batch

- Cannot learn incrementally
- Trained offline and then launched
- When new data becomes available, stop & retrain
- Often computationally expensive

Online

- System trained in mini-batches
- Small and cheap learning steps
- Good when data arrives in continuous streams (e.g. predicting stock prices)
- Good when data cannot fit into machine (out of core learning)

Instance Based & Model Based Learning

ML systems need to be generalized

After training, how best to generalize to data it has never seen before

Instance-Based

- Compare unlabeled instance vs labelled instance
- Needs a measure of similarity – e.g. how many words in a document
- System learns by heart then generalizes

Model Based

- Build a model from training data
- Use the model to make predictions

ML Example

Does Money make people happy?

OECD

- Organisation for Economic Co-operation and Development (OECD) produce a measure of a country's happiness.
- Factors in many variables – Life expectancy, level of education, air pollution etc

IMF

- Produce GDP information for various countries.

Is it possible to combine both sets of data and determine if money makes people happy

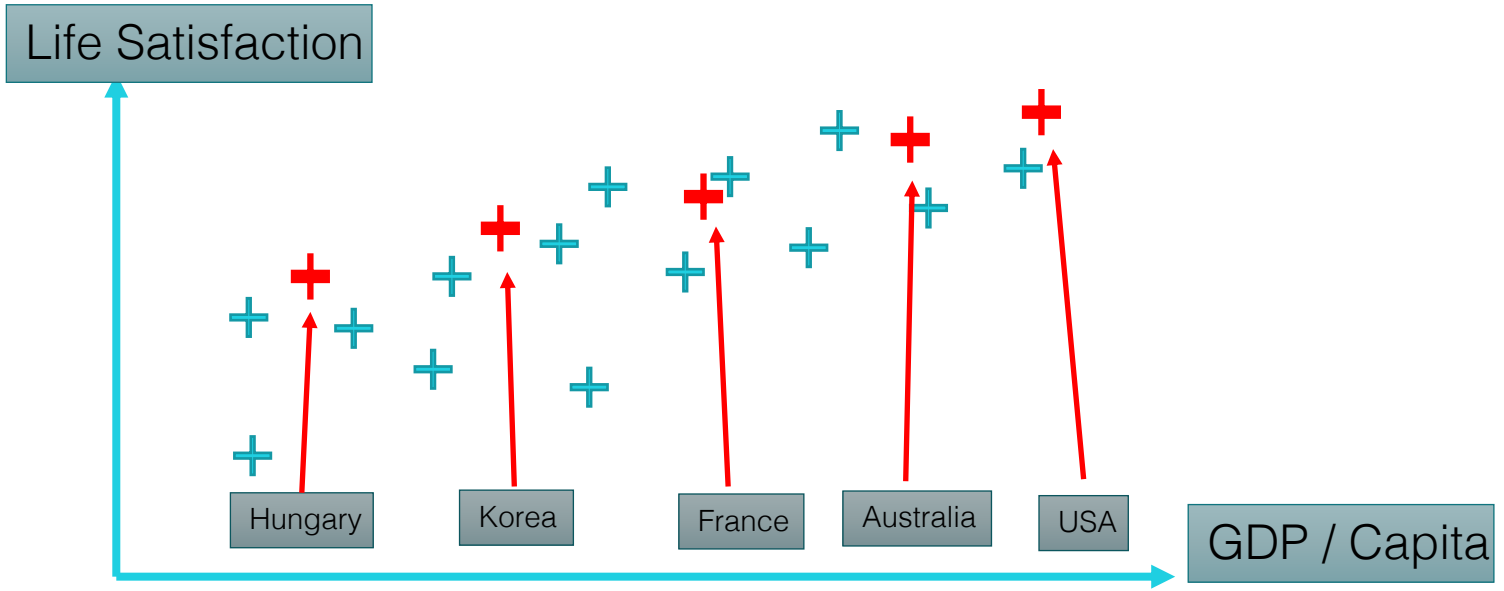
Example – Money & Happiness

The data suggests an increase in satisfaction in line with a countries GDP

An eyeball seems to suggest this is a linear function

Country	GDP per capita	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2

$$life\_satisfaction = b_0 + b_1 \, X \, GDP_{(per\_capita)}$$



neueda

Example – Money & Happiness

Same as straight line equation $y = mx + c$

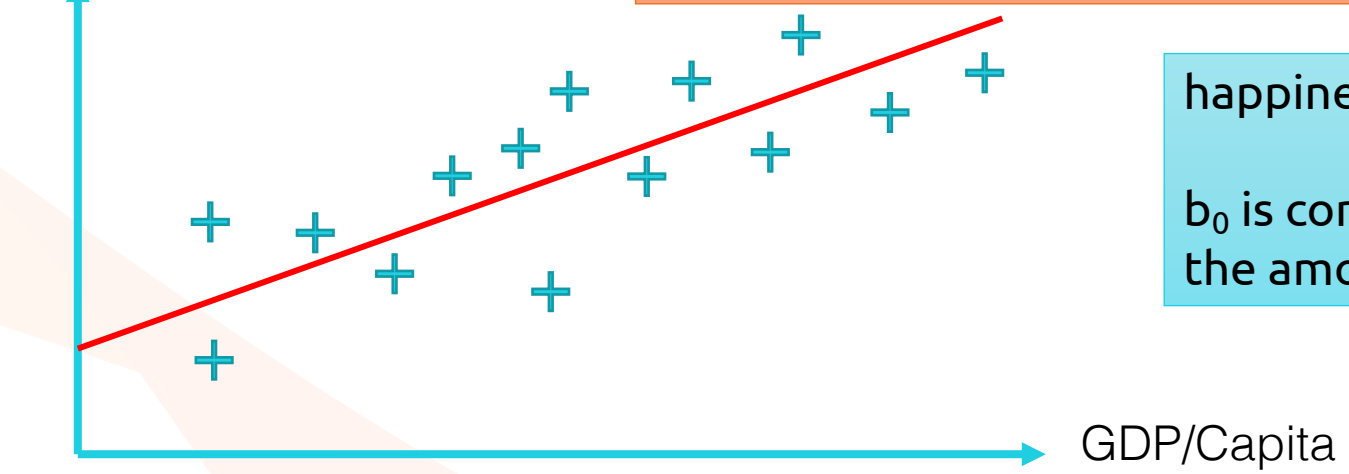
Coefficient
A measure of how sensitive the dependent variable is to unit changes in the independent variable(s)

Dependent Variable
The thing we are trying to explain

Constant

Independent Variable
What causes the dependent variable to change

$$y = b_0 + b_i \times x_i$$



happiness = $b_0 + b_i \times$ experience

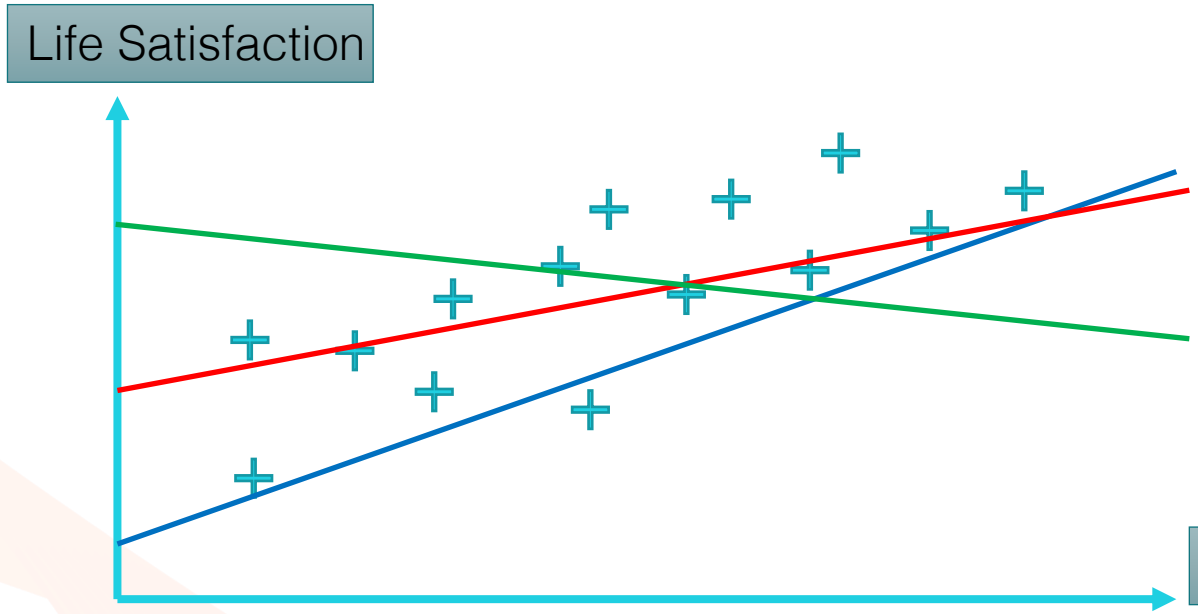
b_0 is constant,
the amount of happiness for having no money

neueda

Example – Money & Happiness

This model has 2 parameters, $\boldsymbol{b_o}$and $\boldsymbol{b_1}$

Its possible to change these parameters to represent any linear function

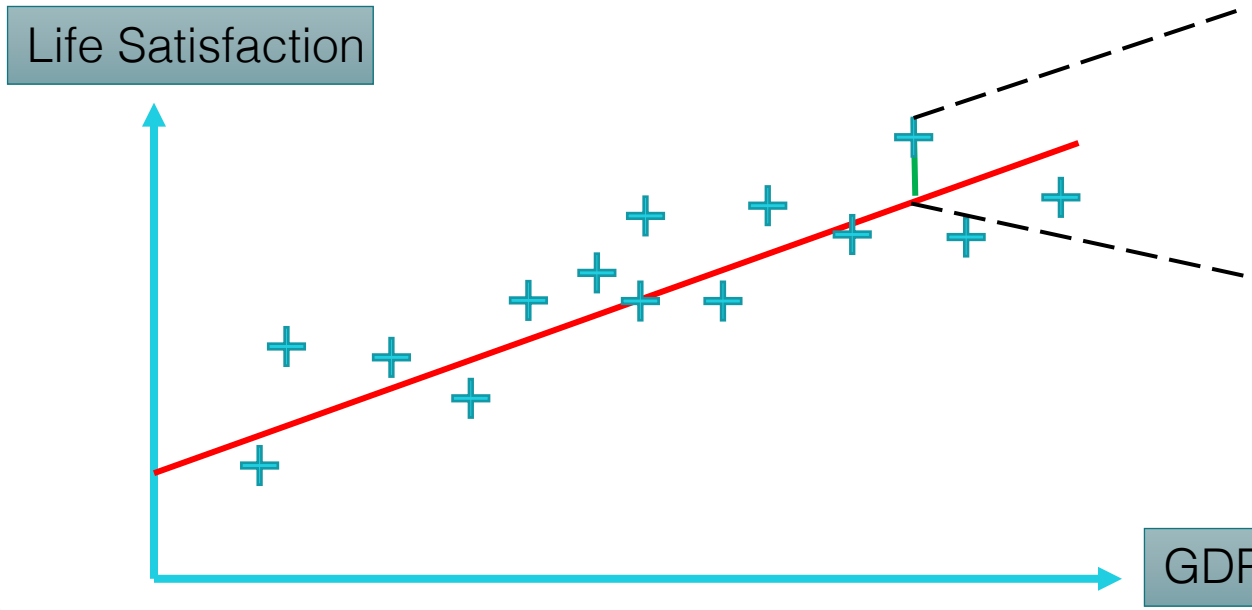


What are the correct values for $\boldsymbol{b_o}$and $\boldsymbol{b_1}$?

Use either

a. a fitness function (how good is it)
b. a cost function (how bad is it)

Example – Money & Happiness



Y_i – what the value actually is.

$Y_i\hat{}$ – what the value should be
The model / predicted value.

$$Y_i - Y_i\hat{} = \text{difference}$$

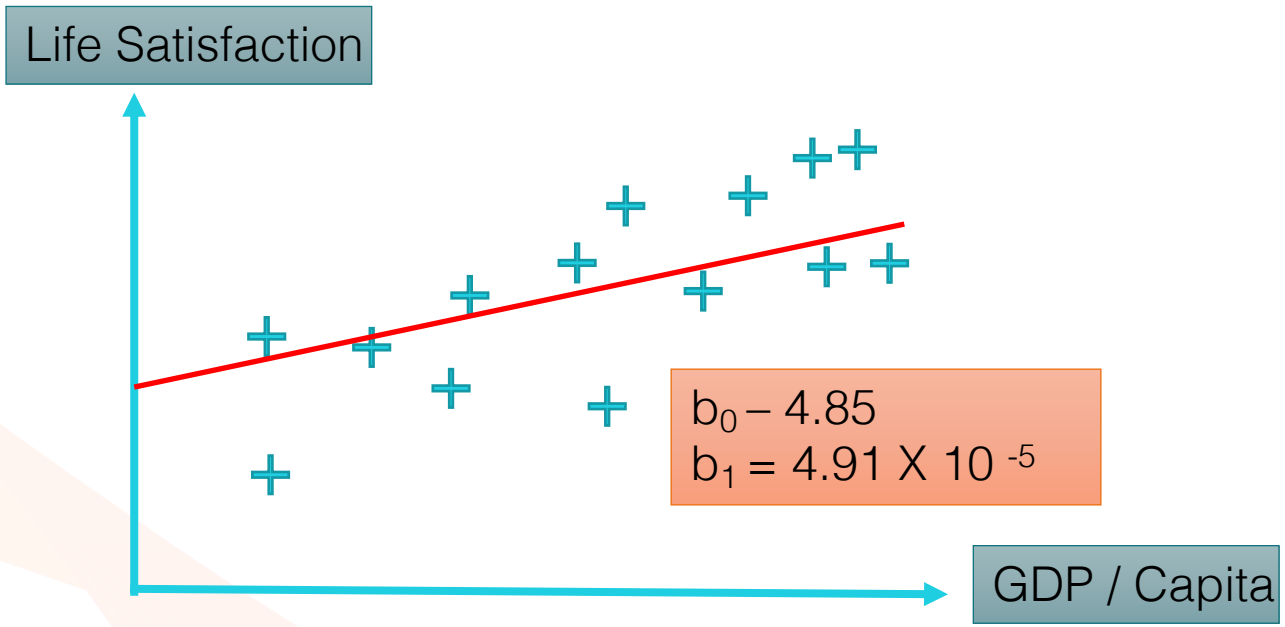
$$\sum(Y_i - Y_i\hat{})^2$$

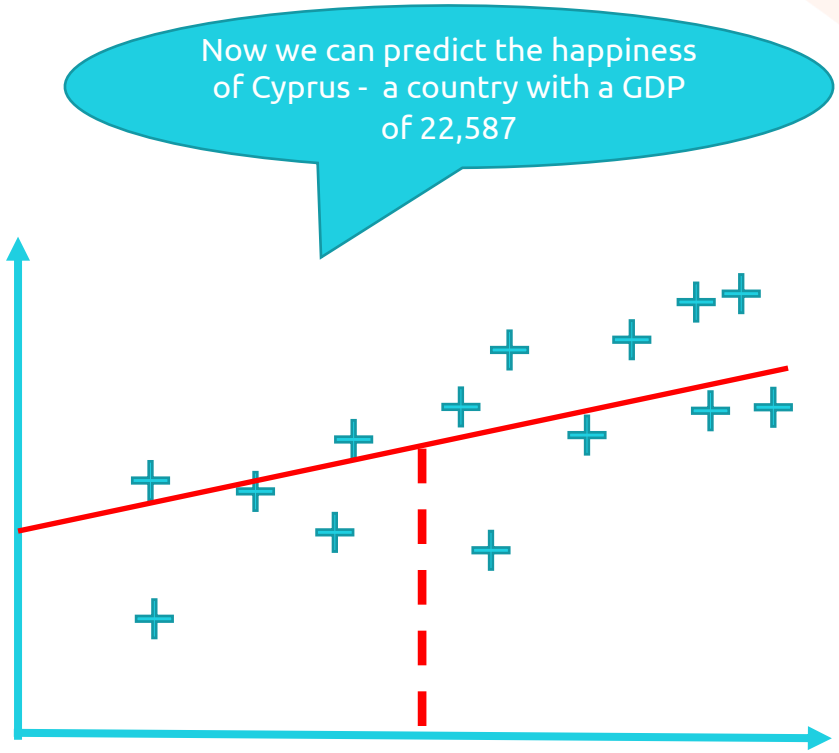
Linear regression based on best fit line
Draw a line, measure the difference between actual and expected
Square the differences and sum.
The line with the smallest sum of squares is the best fit

Example – Money & Happiness

Give the model some data and it finds the values that best fit the data

AKA **training** the model





Example – Money & Happiness

A very popular ML library for python is **Scikit-Learn**

Free, works seamlessly with **pandas** and **numpy**

Supports many common ML models

Some knowledge of python and basic data science is required

- Slicing, DataFrames, variables, string manipulation etc

Example – Money & Happiness

Import the python packages

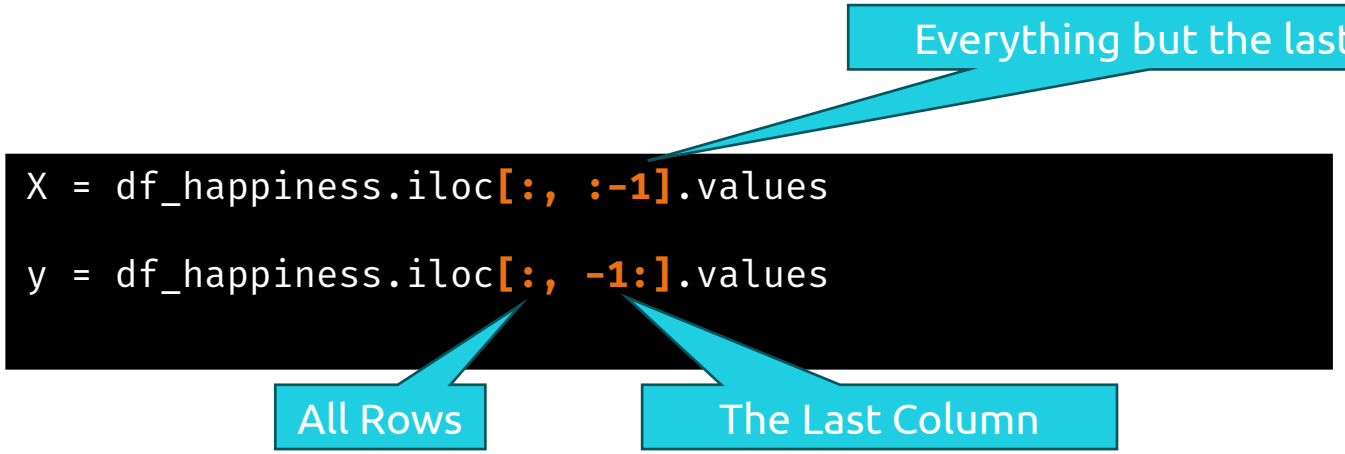
```
import pandas as pd

import sklearn.linear_model
```

Import the data

```
df_happiness = pd.read_csv(filepath_or_buffer='../Data/life_satisfaction.csv',
                           index_col='Country')
```

Example – Money & Happiness



Naming Conventions

Uppercase X - a **matrix**

Lowercase y – a **vector**

```
model = sklearn.linear_model.LinearRegression()

model.fit(X, y)
```

Example – Money & Happiness

Make a prediction

e.g. the happiness of Cyprus, given its GDP/Capita ($22,587)

A list of inputs

A list of outputs

```
df_gdp = pd.read_csv(filepath_or_buffer = "../data/gdp.csv", index_col='Country')

gdp_cyprus = df_gdp.loc['Cyprus']

pred = model.predict([gdp_cyprus])

print(pred[0])
```

Example – Money & Happiness

See Notebook 01A – Money_And_Happiness for examples

Complete Lab 1A - Money_And_Happiness