Big data science Day 2

F. Legger - INFN Torino

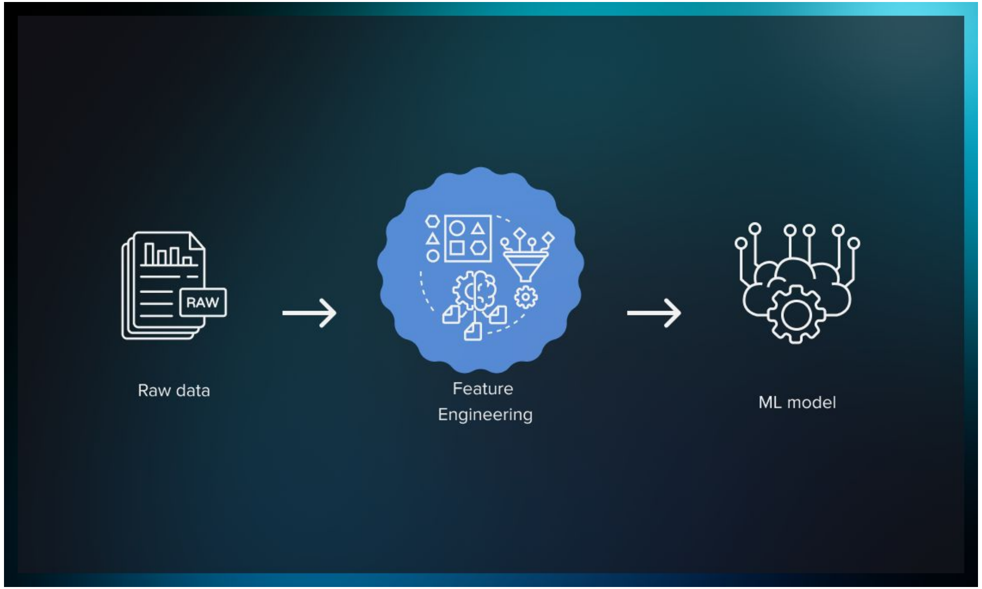
https://github.com/Course-bigDataAndML/MLCourse-2223

Yesterday

- Big data
- Analytics
- Distributed Computing infrastructure

Today

- **Machine learning**
  - Feature engineering
  - Supervised models
  - Unsupervised models



A PROPOSAL FOR THE

DARTMOUTH SUMMER RESEARCH PROJECT

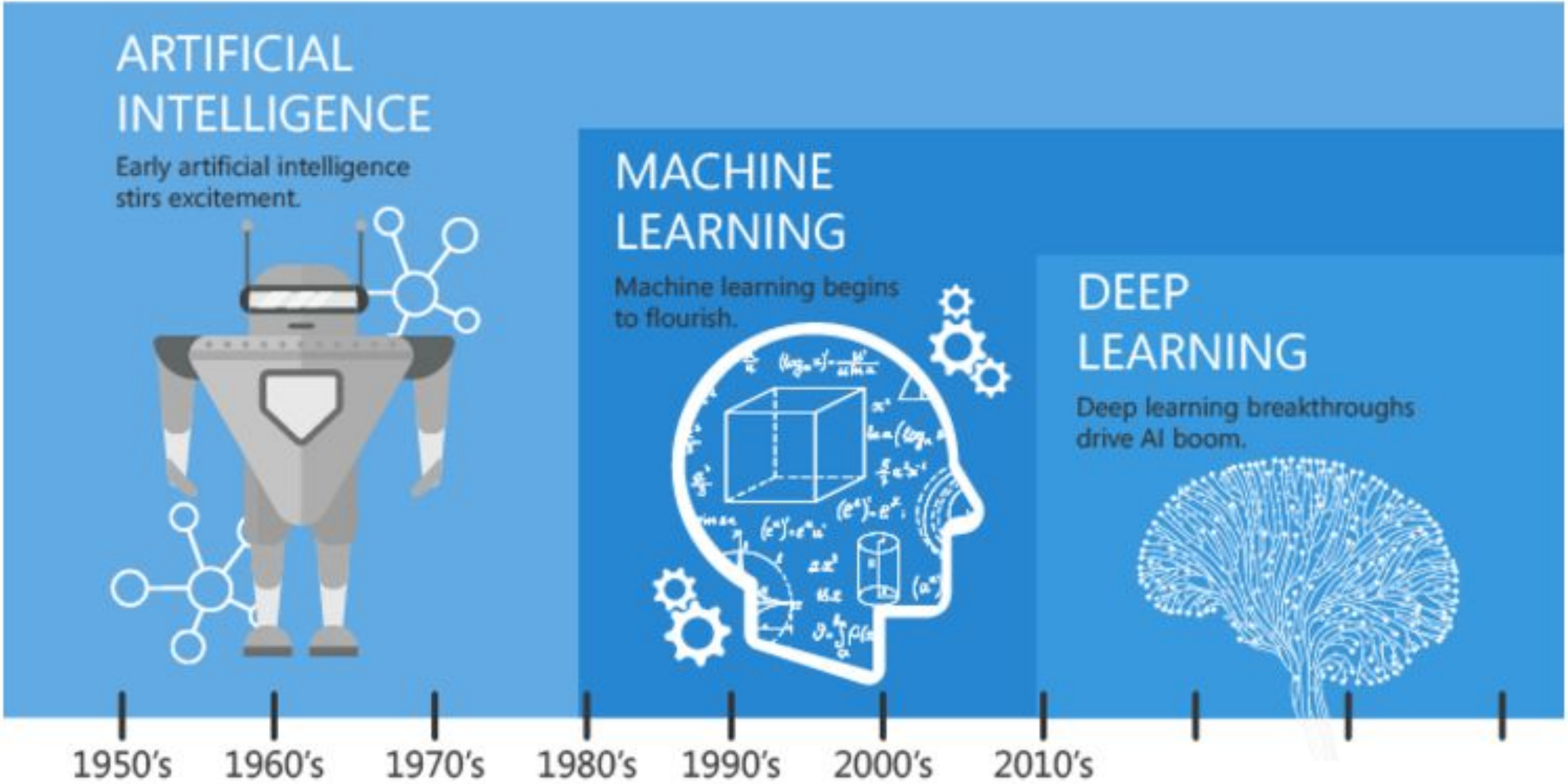
ON ARTIFICIAL INTELLIGENCE

J. McCarthy, Dartmouth College
M. L. Minsky, Harvard University
N. Rochester, I. B. M. Corporation
C. E. Shannon, Bell Telephone Laboratories

August 31, 1955

Our ultimate objective is to make programs that learn from their experience as effectively as humans do

[John McCarthy, 1958]


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

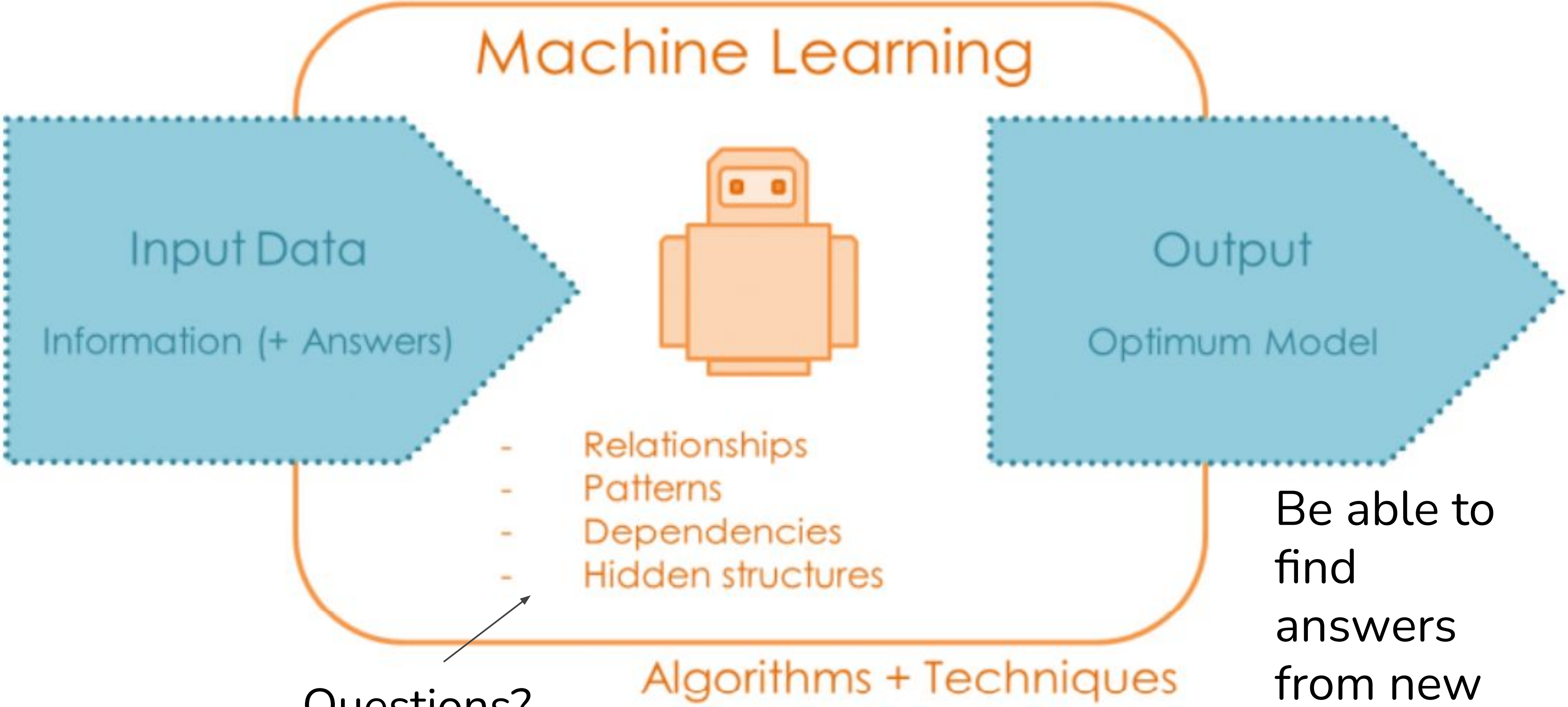
Machine Learning

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead [Wikipedia]

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at task in T, as measured by P, improves with experience E

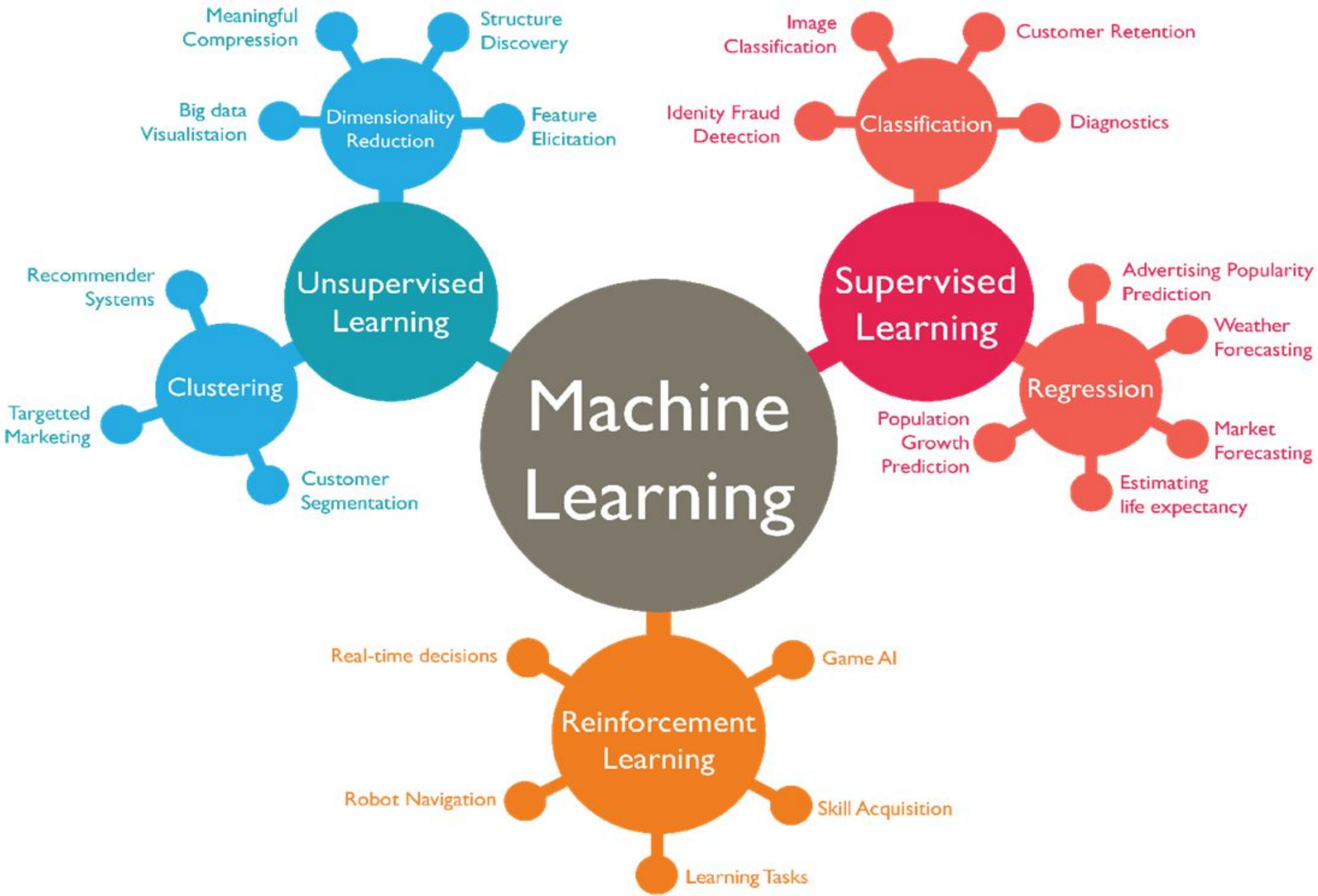
[Tom Mitchell, 1997]

Machine Learning is the science of getting computers to act without being explicitly programmed [Andrew Ng]



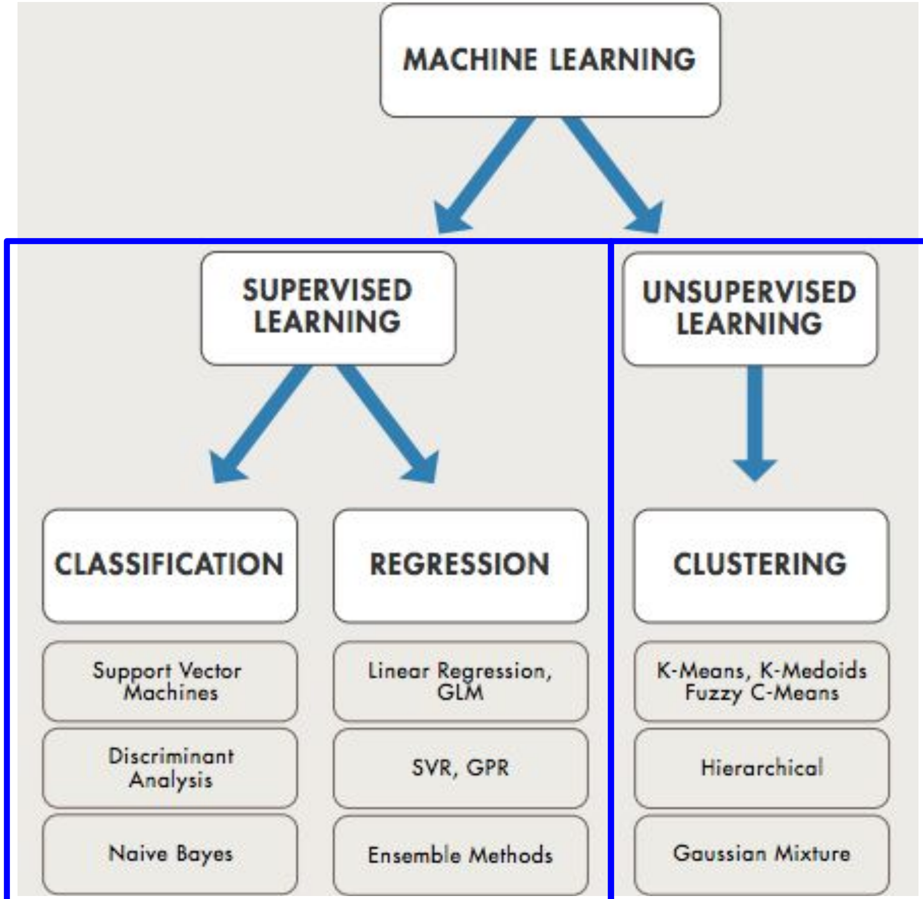
Questions?

Be able to find answers from new data


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

Are input data labelled?

sensor1	sensor 2	sensor3	label
0.3	0.2	0.6	0
0.3	0.2	0.6	0
0.3	0.2	0.6	0
0.3	0.2	0.6	0
0.3	0.2	0.6	1
0.3	0.2	0.6	1



Discrete labels

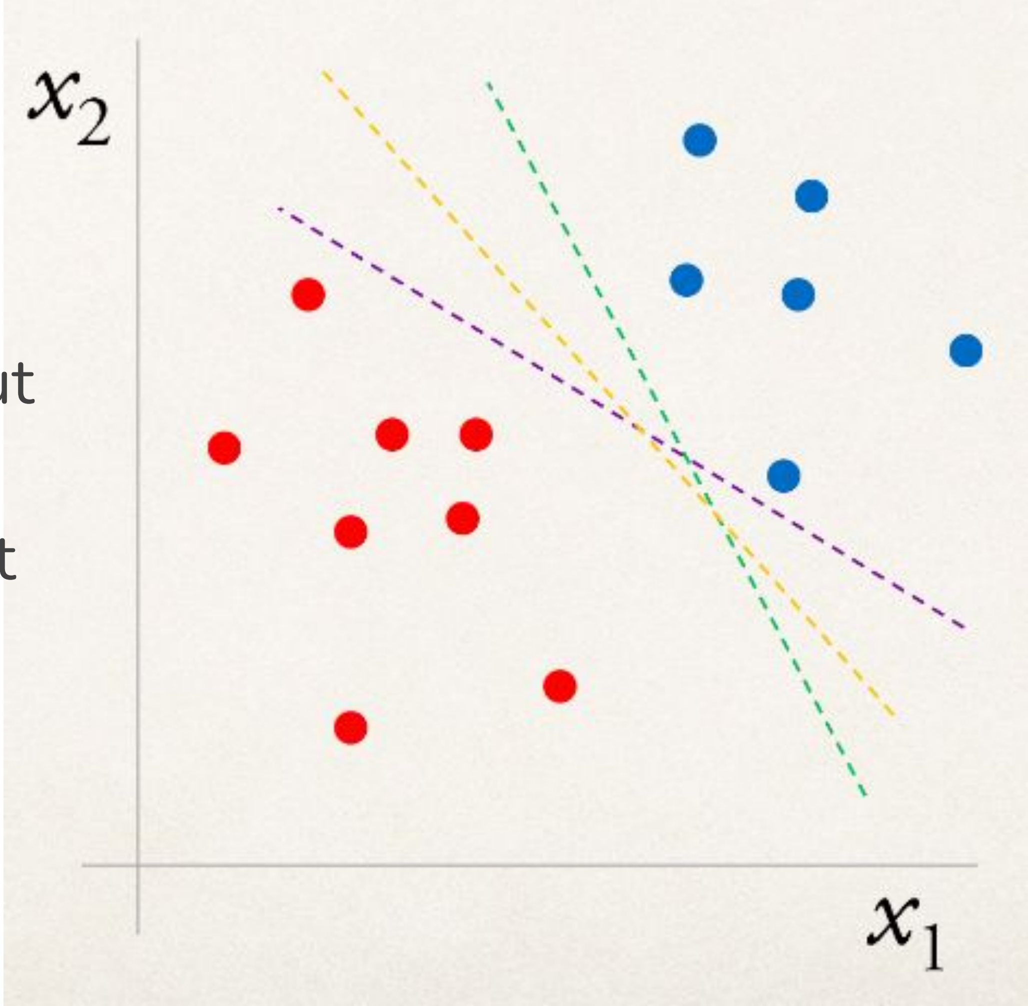
Continuous labels

No labels

Classification

Supervised, discrete labels

- Predict one or more output class
  - Businesses who target customers: good vs bad, stay or leave
  - **Signal vs background**

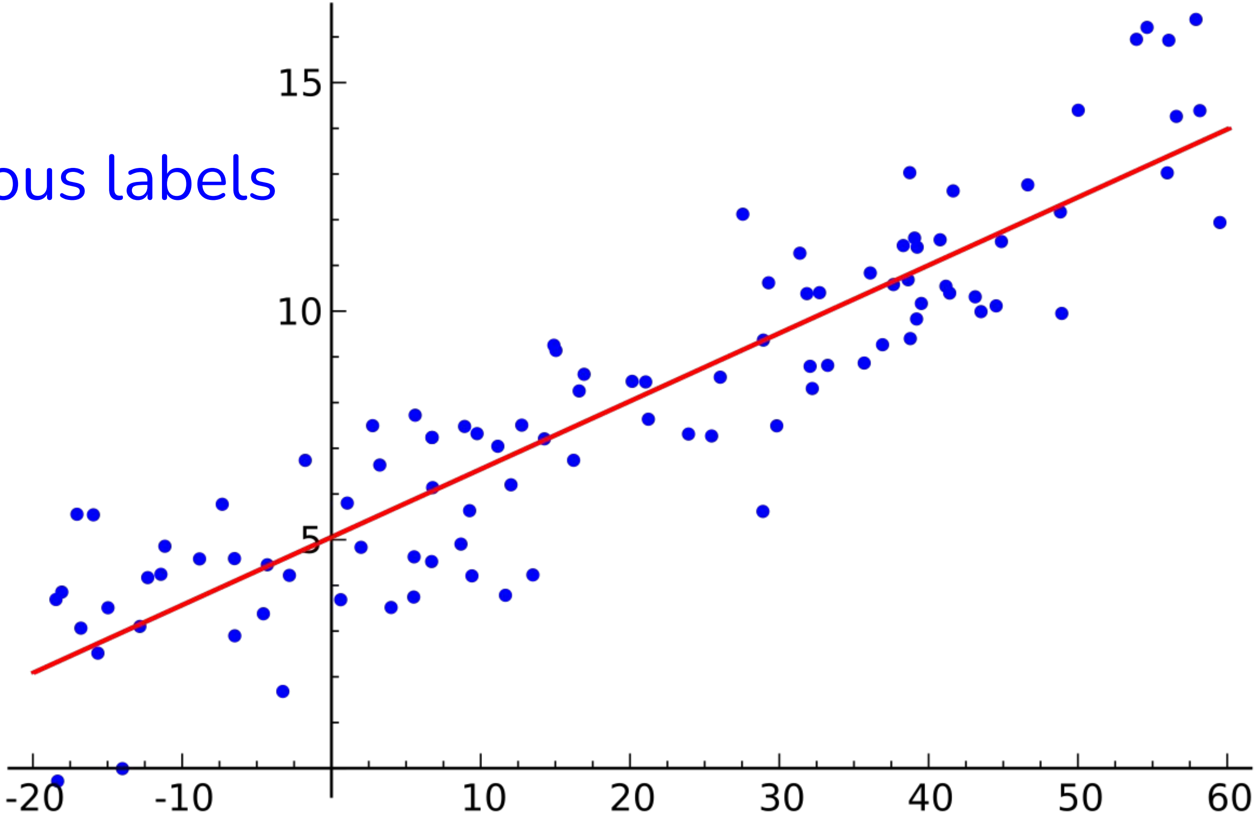


hands-on

Regression

Supervised, continuous labels

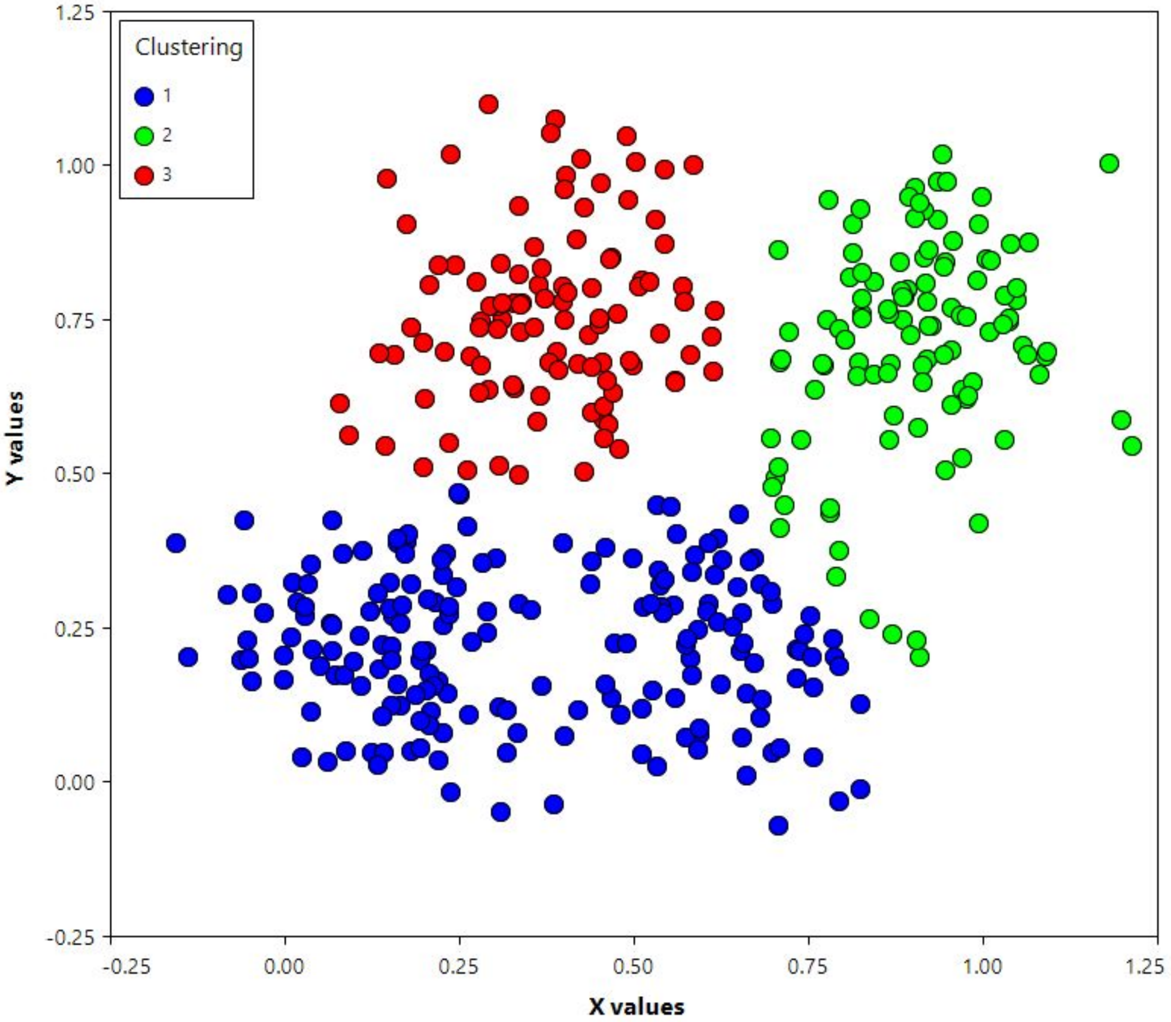
- Businesses who predict customer behavior: e.g. house prices, ...



Clustering

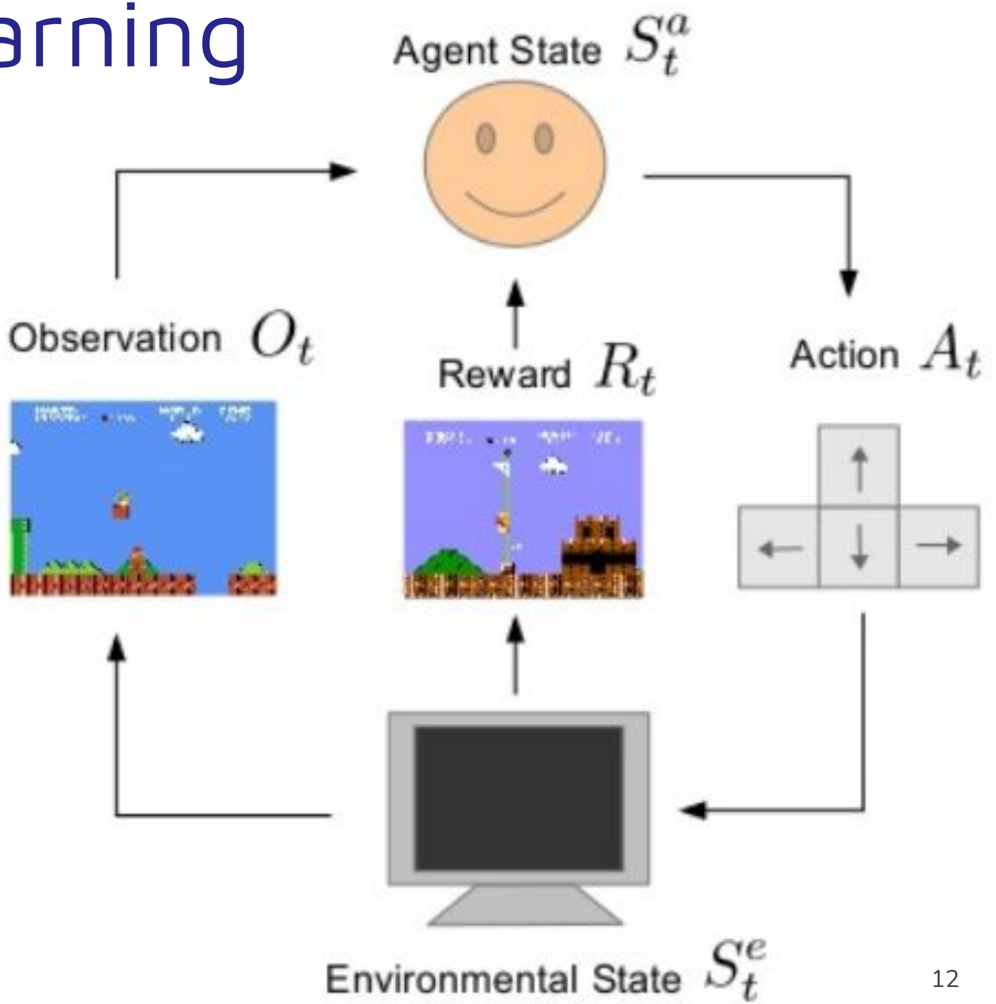
Unsupervised

- Businesses who identify customer categories
- Light vs heavy flavour jets
- ....

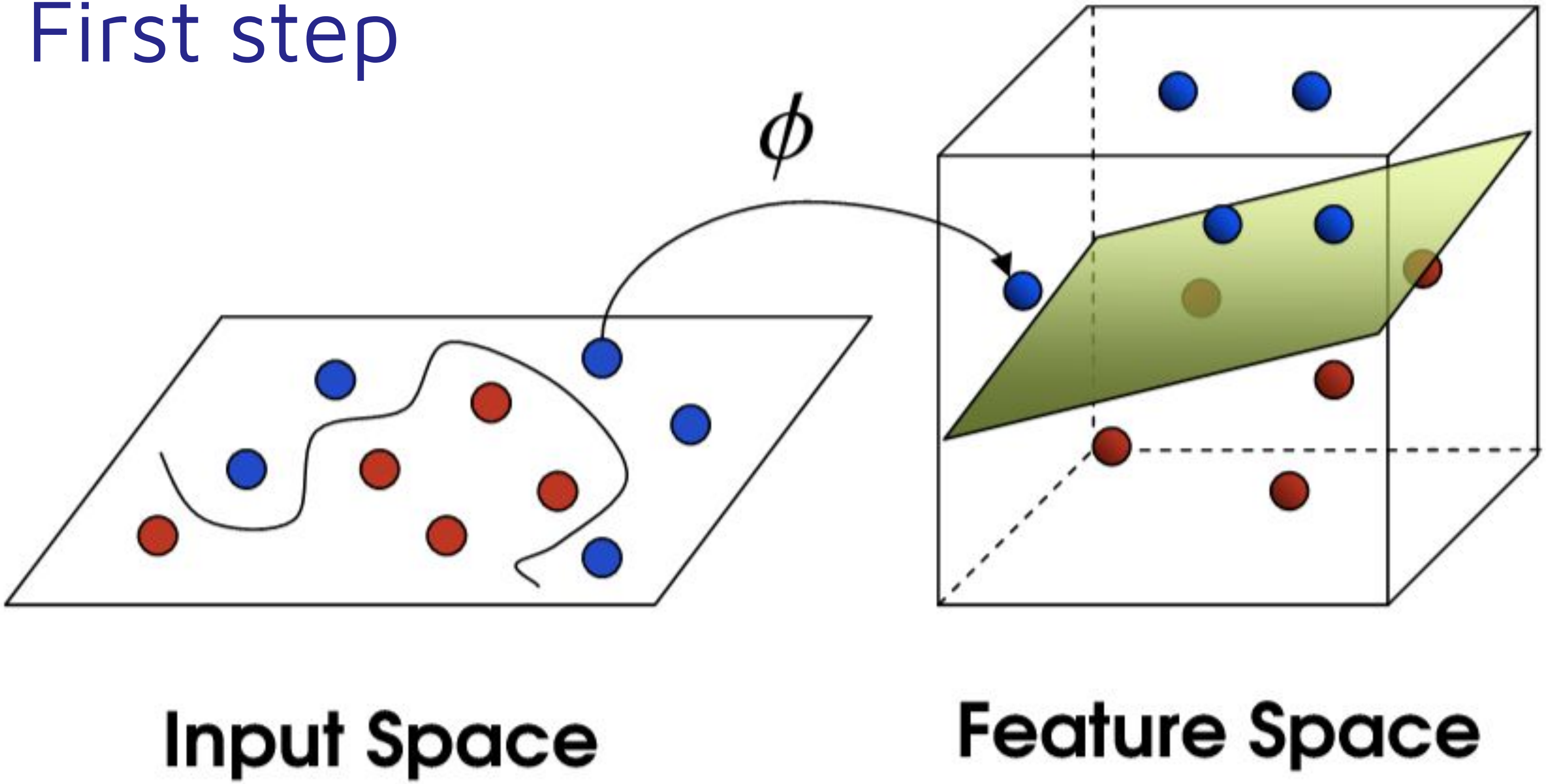


Reinforcement learning

- getting an agent to act in the world so as to maximize its rewards
- sparse and time delayed labels (**rewards**)



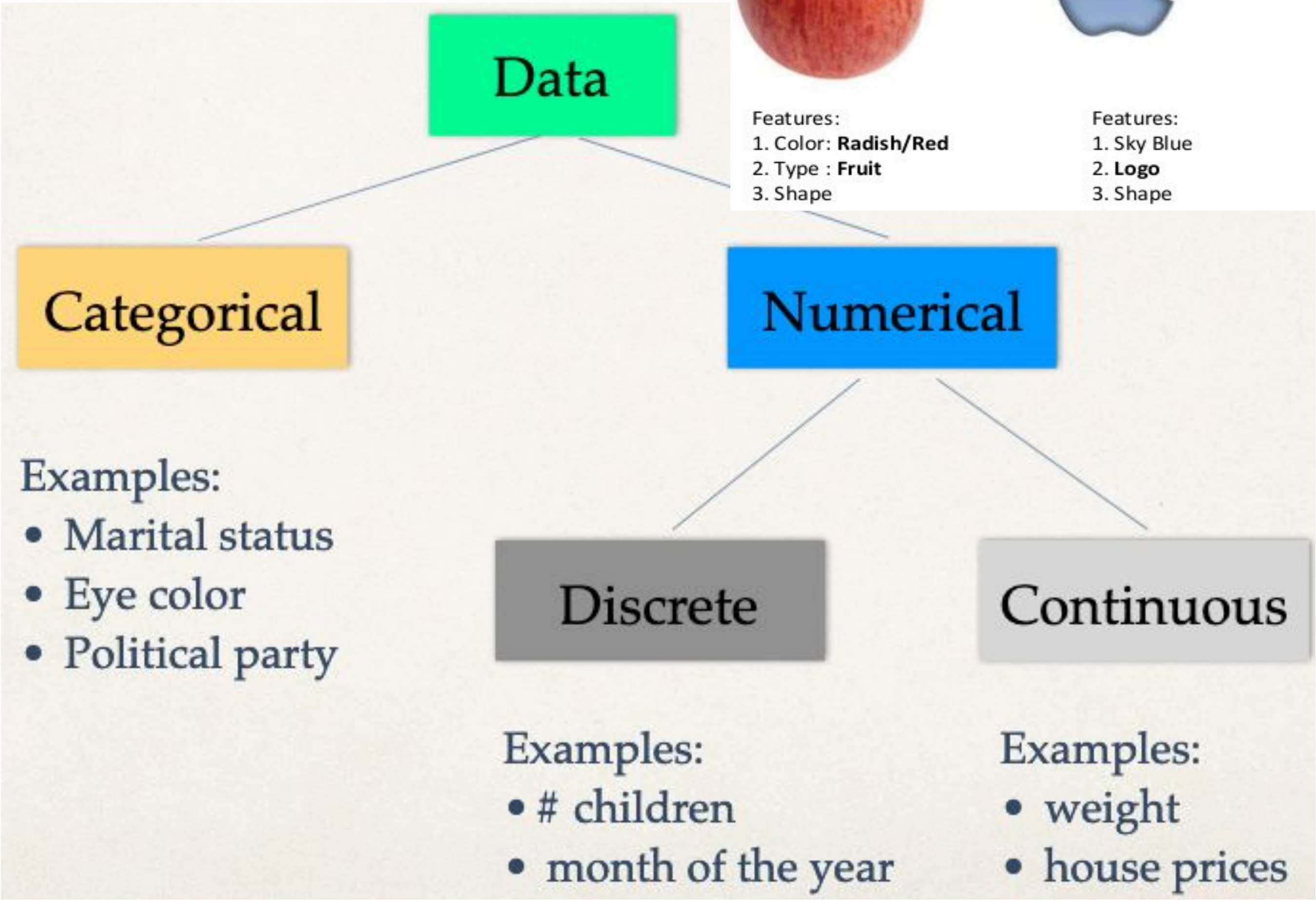
First step



Raw data → **Preprocessing**

Data types

Typically strings

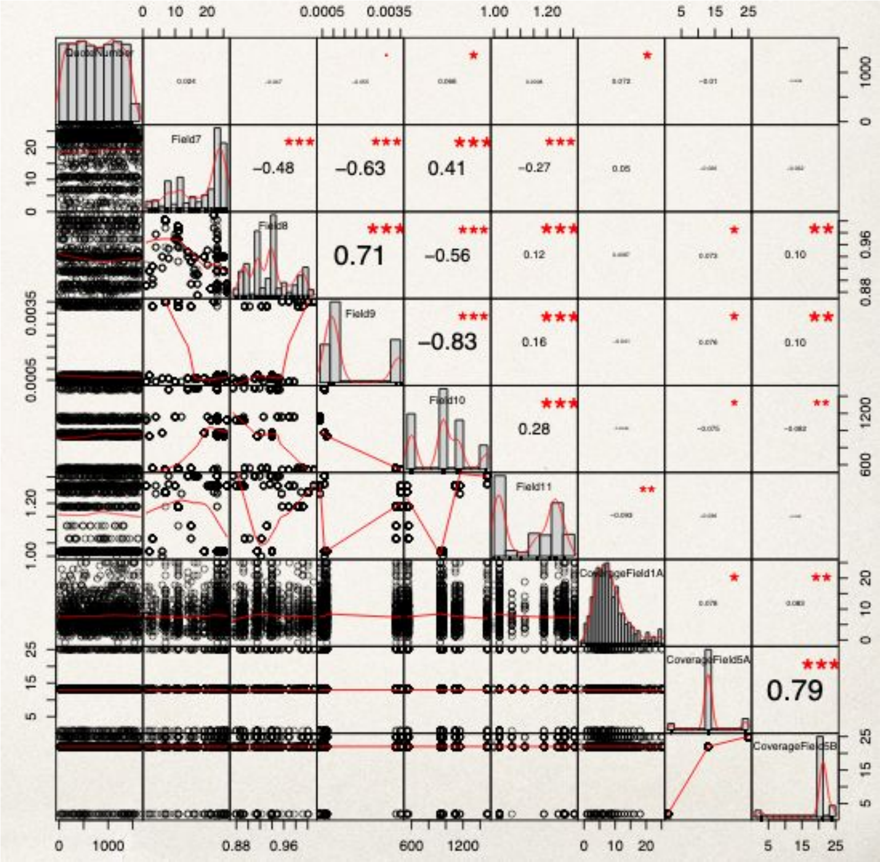


Data preprocessing

- Most of the time will be spent in this step
- Data clean-up, data transformation, feature engineering
  - **data transformation**
    - scaling and normalization
    - encoding, aggregation features, log-transformation (to remove outliers)
  - **data visualization, exploration**
  - **data augmentation, bucketing, binning, ...**
  - **dimensionality reduction**
- Your programming skills will be required here: **R, Python,** ...

Data visualization

- Graphical representation may reveal important features of the data
  - find correlations, identify range, etc.
- Identify features which may require transformations, e.g. see outliers or skewness (asymmetry in probability distribution) in data
- It helps to identify a strategy how to deal with different features



Data transformation

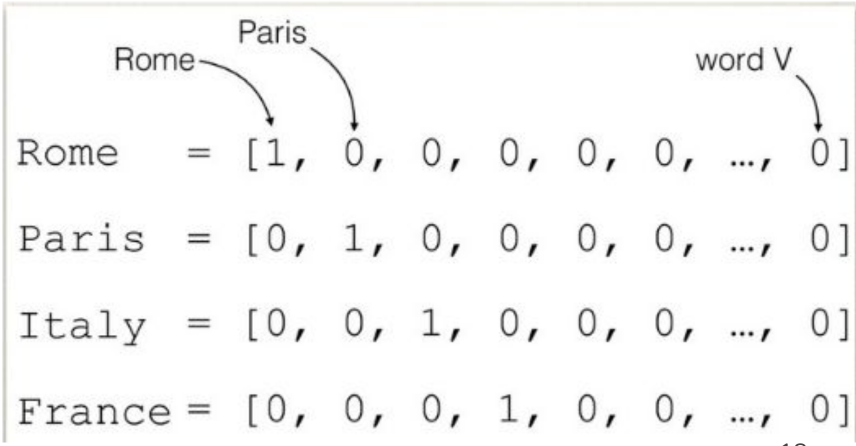
$$x' = \frac{x - \bar{x}}{\sigma}$$

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- Data **transformation and *aggregation***: log, sum of values, average
- **Scaling**: a technique to scale data to a given range [0,1] or any other range
- **Normalization/Standardization:** a technique to scale data to mean with zero and and unit-variance
- **Augmentation:** a technique to create additional data based on input sample which slightly differ from it, e.g. image rotation, flip, scale, crop, etc.
- **Bucketing/Binning**: a technique to place similar values into buckets/bins

One-hot encoding

- Technique to handle **categorical** data
- “One-Hot” refers to a state in electrical engineering where all of the bits in a circuit are 0, except a single bit with a value of 1 (“hot”)
- It represents a categorical column as a vector of words
- You need to define the word vector for the full set of data (train + test datasets)
    - Issues with NULL or missing data
        - delete rows with missing data
        - input data for missing values
    - Problematic with high cardinality



Leave-one-out encoding

Mean of [1,1,0]

mean_y* random

- Effective by high cardinality
- Y is what we want to predict
- Encode UserID:
    - Train dataset:
        - Take mean of Y's for all rows with same UserID except the one you want to encode
        - multiply random noise
    - Test dataset
        - No Y, just use frequency of UserID

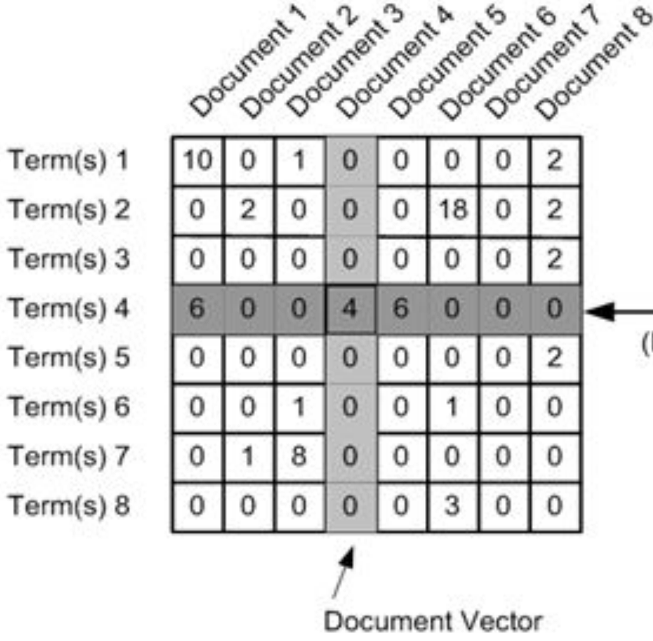
Split	UserID	Y	mean_y	random	newID
Train	A1	~~0~~	0.667	1.05	0.70035
Train	A1	1	0.333	0.97	0.32301
Train	A1	1	0.333	0.98	0.32634
Train	A1	0	0.667	1.02	0.68034
Test	A1	-	0.5	1	0.5
Test	A1	-	0.5	1	0.5
Train	A2	0			

Word embedding

- A way to capture multi-dimensional relationships between categories
- you define a dimension of word vector up-front
  - it projects categorical variables into another phase space, e.g. days may be sunny or rainy, season or off season, Sunday and Saturday may have similar effect while other days may be treated independently
- Use neural networks or other ML algorithms to train the model to find the best representation of embedded variables

Frequency based word embedding

- **Count Vector**
  - Corpus C of D documents {d1,d2…..dD} and N unique tokens (words) in C
  - The N tokens will form our dictionary and the size of the Count Vector matrix M will be given by D X N. Each row in the matrix M contains the frequency of tokens in D(i)
- **TF-IDF Vector**
  - Similar to Count vector, but frequency is calculated with respect to all documents

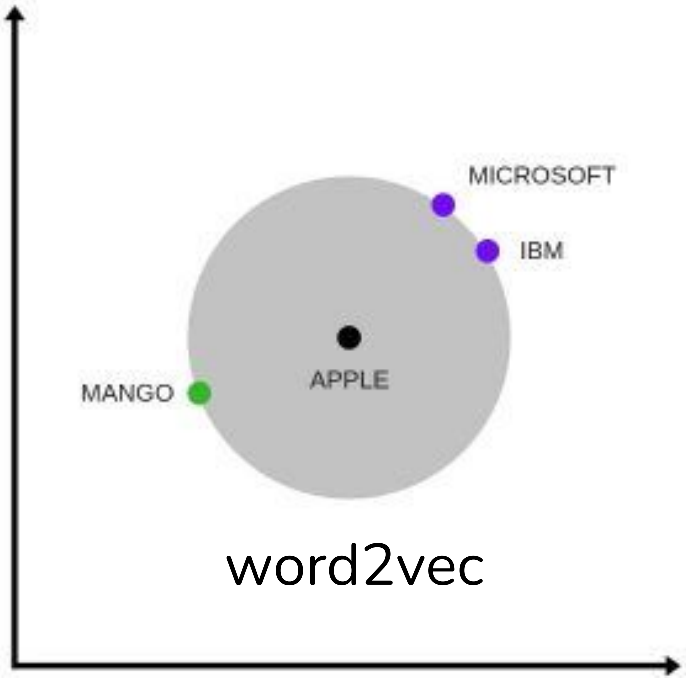


	Document 1	Document 2	Document 3	Document 4	Document 5	Document 6	Document 7	Document 8
Term(s) 1	10	0	1	0	0	0	0	2
Term(s) 2	0	2	0	0	0	18	0	2
Term(s) 3	0	0	0	0	0	0	0	2
Term(s) 4	6	0	0	4	6	0	0	0
Term(s) 5	0	0	0	0	0	0	0	2
Term(s) 6	0	0	1	0	0	1	0	0
Term(s) 7	0	1	8	0	0	0	0	0
Term(s) 8	0	0	0	0	0	3	0	0

- **Co-Occurrence Vector**
  - Based on frequency of words appearing together (for example, it is)

Prediction based word embedding

- **Word2vec** based on neural networks
  - **Continuous Bag of words (CBOW)**: predicts the probability of a word given a context
  - **Skip-Gram model:** predicts the context given a word



Feature selection

- Low vs high level variables