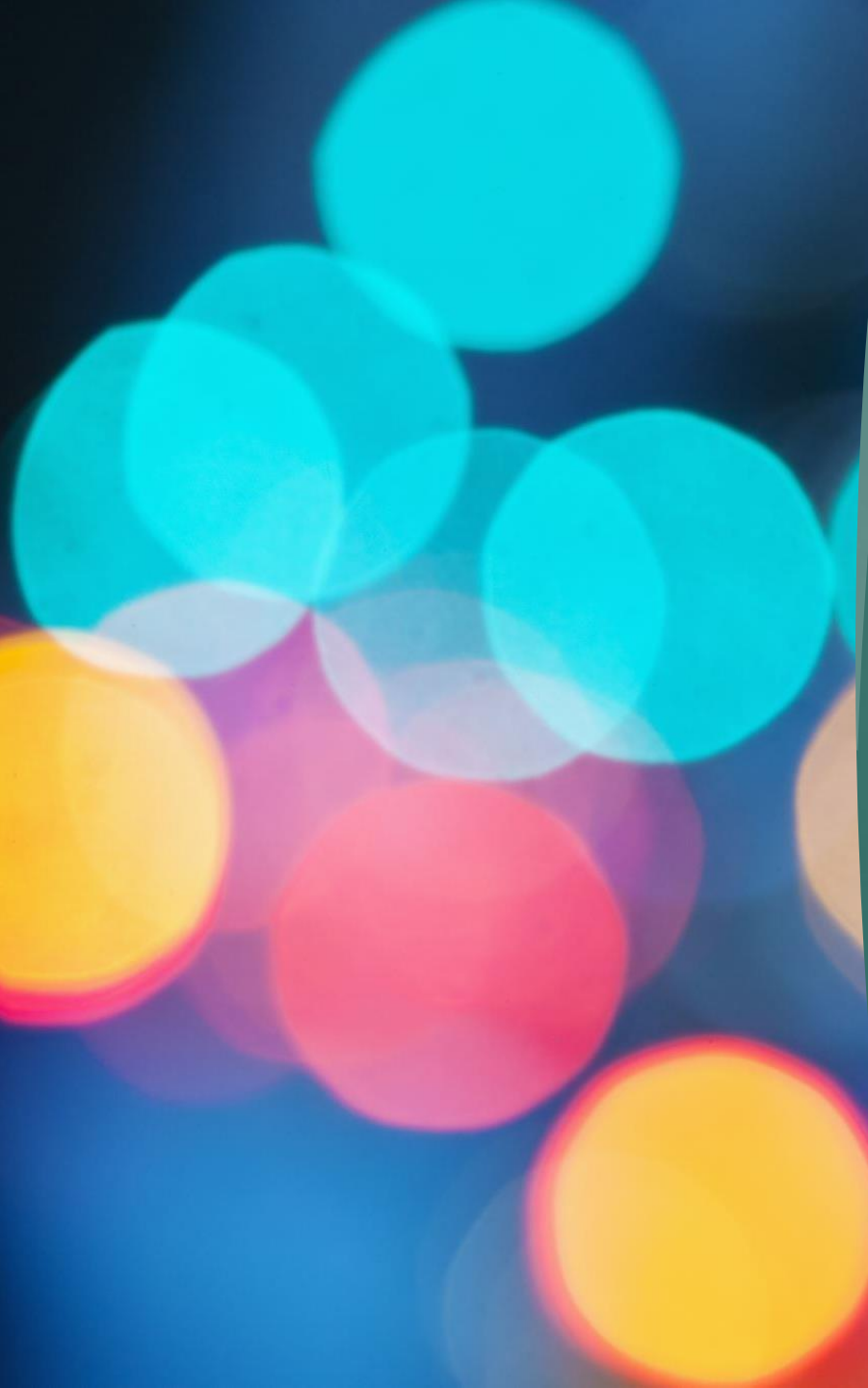


Introduction to Machine Learning and Data Analytics

TRI-DUC NGHIEM

DUC.NGHIEM@PIPEPREDICT.COM

Overview

Machine Learning

Supervised learning

Unsupervised learning

Semi-supervised learning

Reinforcement learning

Data analytics

What to do?

Big data?

Practice

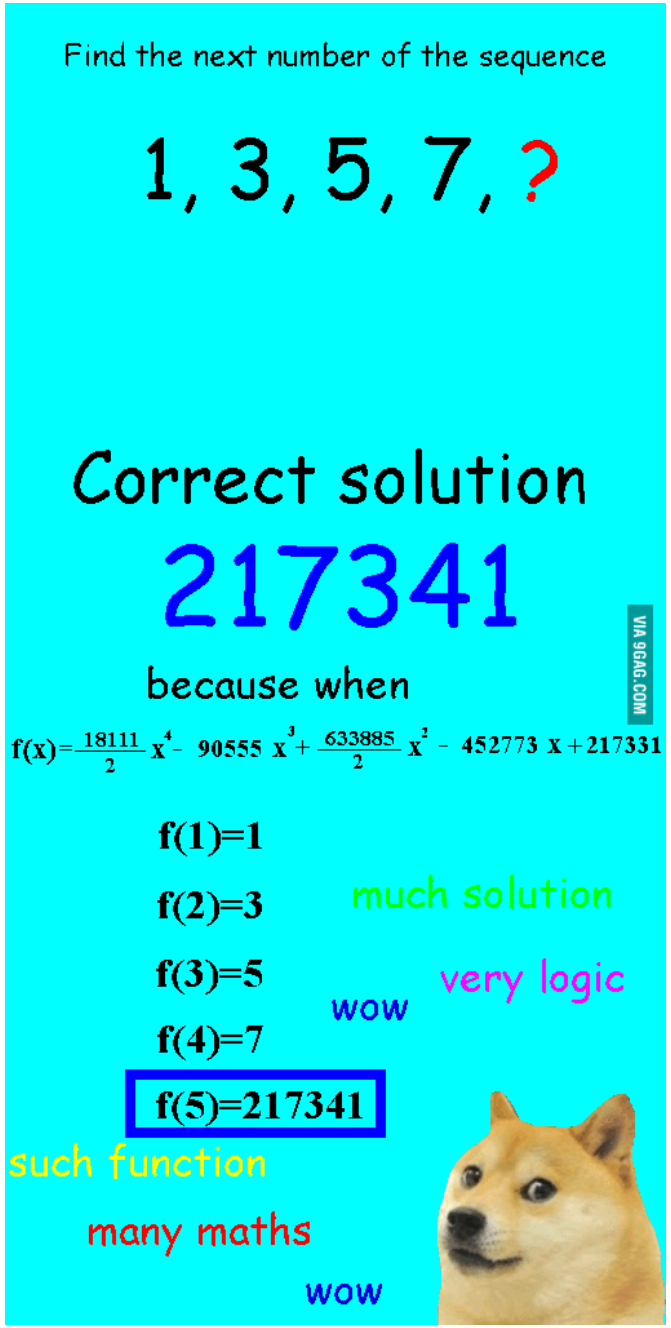
Playing around with scikit-learn, matplotlib

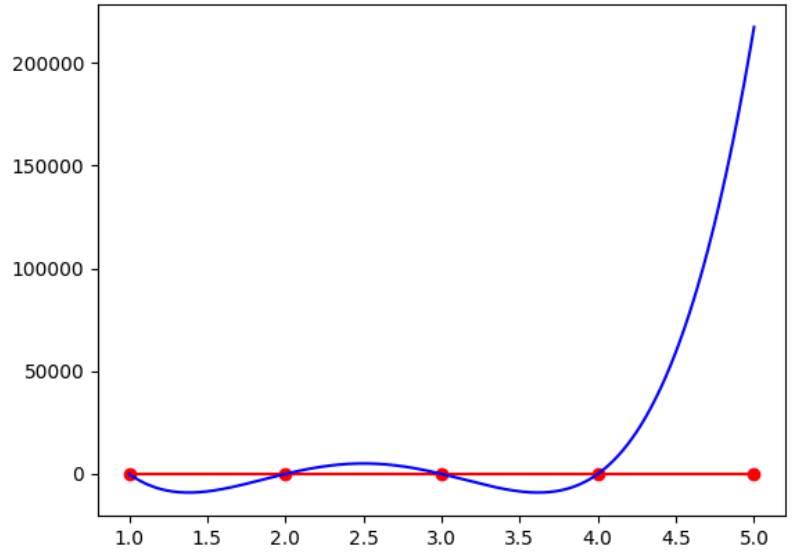
Part I: Machine Learning

- Example:
  - Given a series of integer numbers:
    - 1, 3, 5, 7, …

    What is the next number?

    A linear model (series of odd numbers) will give us 9, but a polynomial function (degree of 4 as below) will give us: 217341





Part I: Machine Learning

- And more:
  - $R = \frac{U}{I}$
  - $F = G\frac{(m_1 \times m_2)}{r^2}$
    - (It's true until Einstein proved it's wrong)

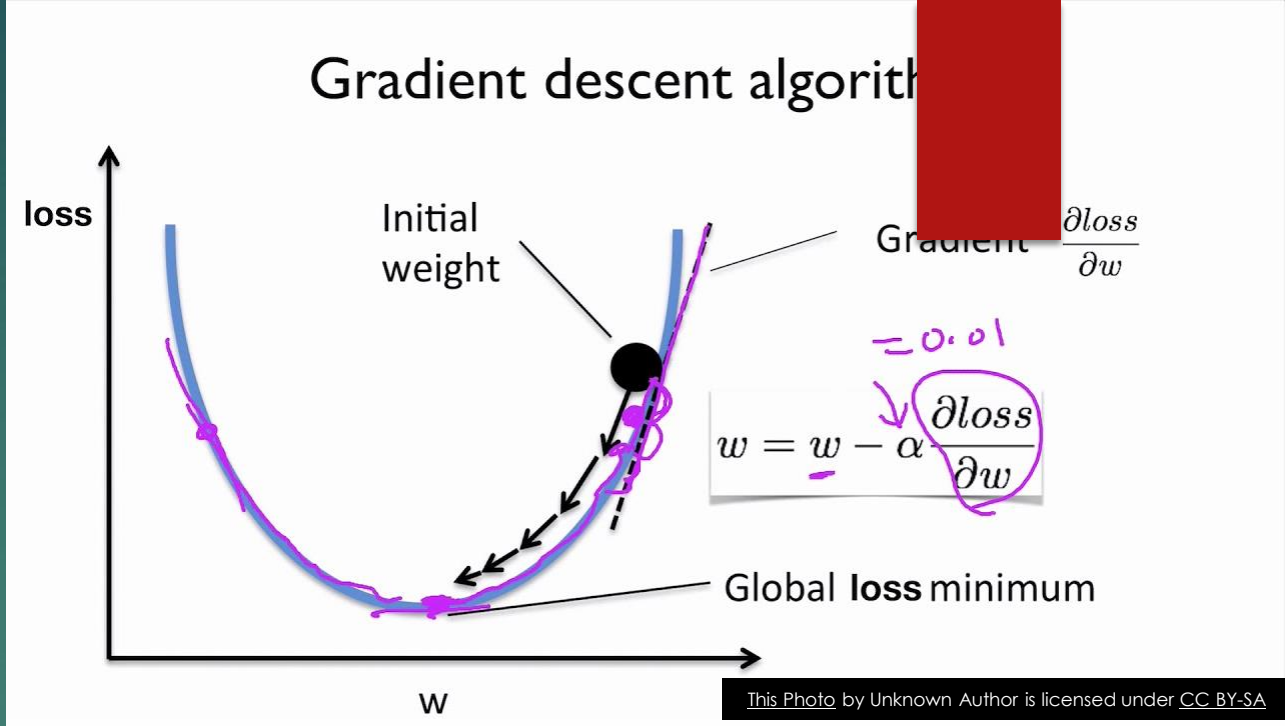
"All models are wrong, but some are useful"

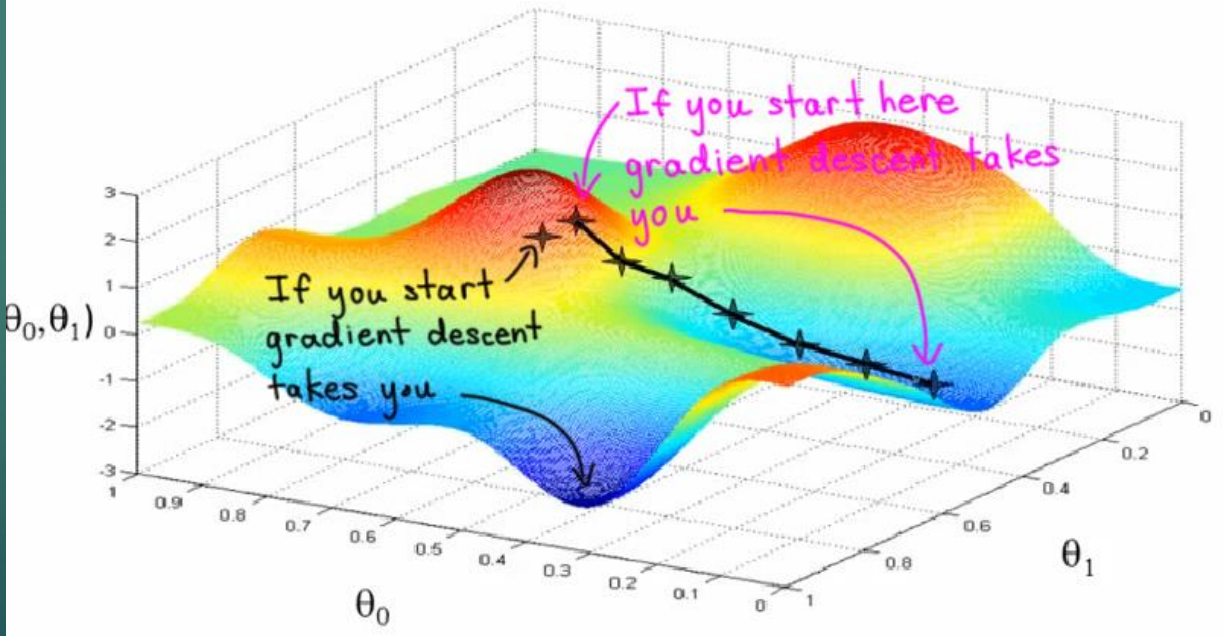
George Box

I.I Supervised Machine Learning

- Observation:
  - (x_i, y_i) where $i = 1..n$
- Estimate F(x) = y with set of parameters and hyper-parameters
- Such that the loss over the observation set is minimum:
  - Minimize $loss = \sum_i L(F(x_i), y_i)$
- Example:
  - In the first example: $y = w_0 + \mathrm{w}_1 \times x + w_2 \times x^2 + w_3 \times x^3 + w_4 \times x^4$
    - Hyper-parameters: degree of the polynomial: 4
    - Parameters: $w(w_0, w_1, w_2, w_3, w_4)$
  - **Deep Learning** is nothing different, it's just a chain of functions applied on the inputs

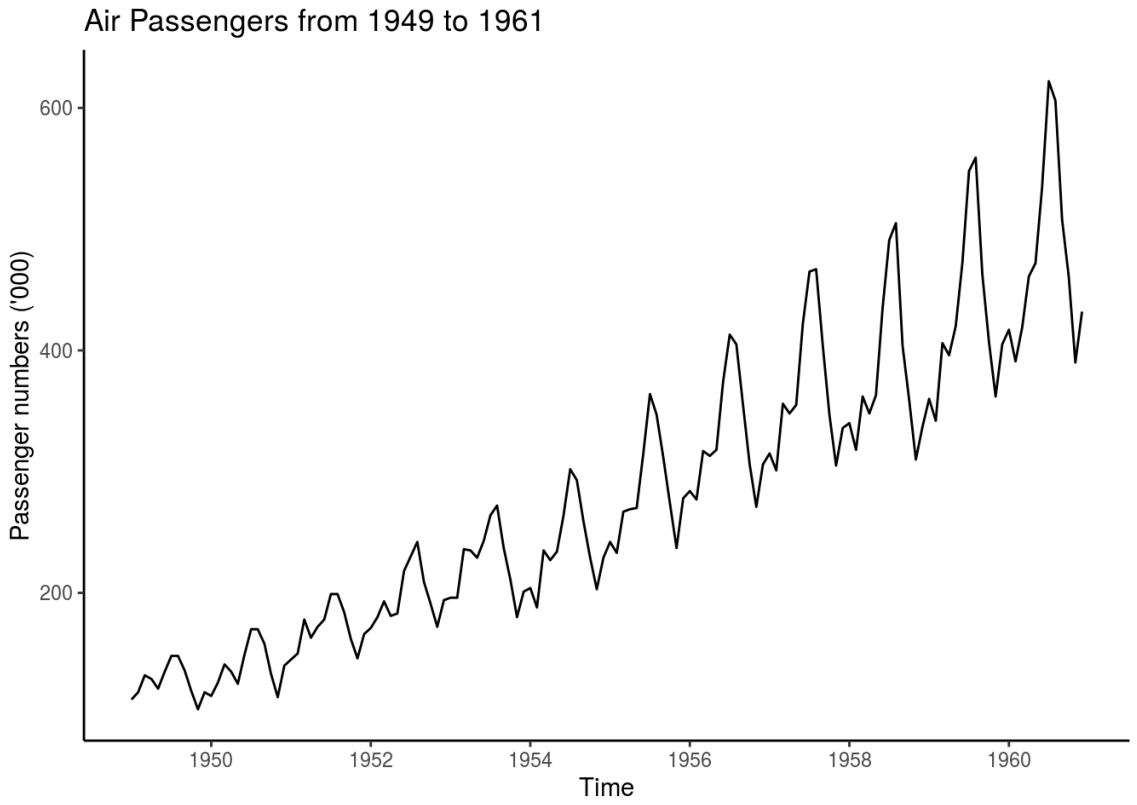
I.I Supervised Machine Learning How does it work?





I.I Supervised ML: Regression

- If F estimates continuous output, then the problem is a regression problem
  - Regular Regression problems:
    - Predict the price of a house / an apartment based on:
      - Location
      - Size
      - built-year
  - Timeseries: forecast future values of a series based on observed set of values
    - Price of flight ticket
    - Stock market



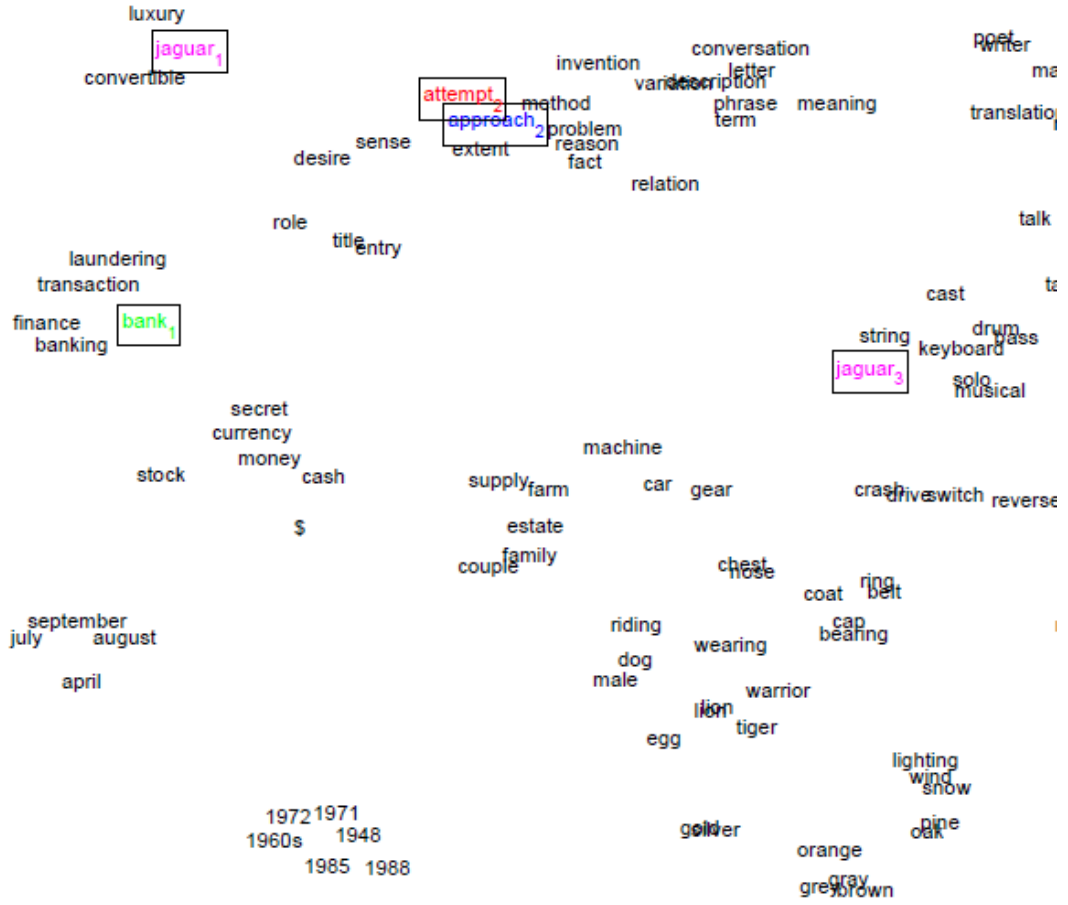
I.I Supervised ML: Classification

- If F's range is discrete and limited
- Examples:
  - Spam filter (binary classification)
  - Text categorization (multiple-class classification)
  - Handwriting recognition
  - Speech recognition
  - …

Example of MNIST Dataset – handwriting digits recognition

I.I Supervised ML: Representation

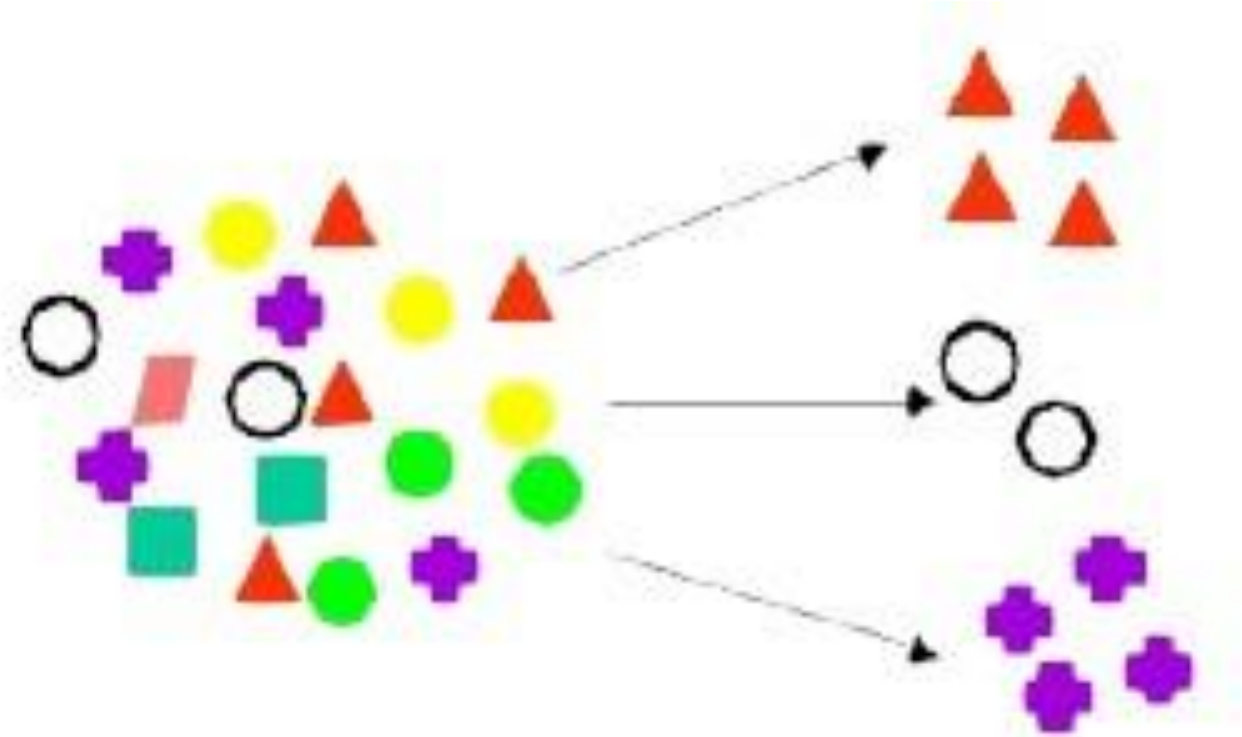
- When F(X) -> X', it's representation problem
  - X' may have less dimension or larger
  - Sometimes it's referred as autoencoder, feature learning
  - Technically, it's unsupervised as we don't need labels; but mechanically it works as supervised algorithm since you still need a lot of (unlabeled) data to fit the function
- Example:
  - Word embedding: words are represented as vectors



Word vectors visualization showed that it can capture semantic level of words

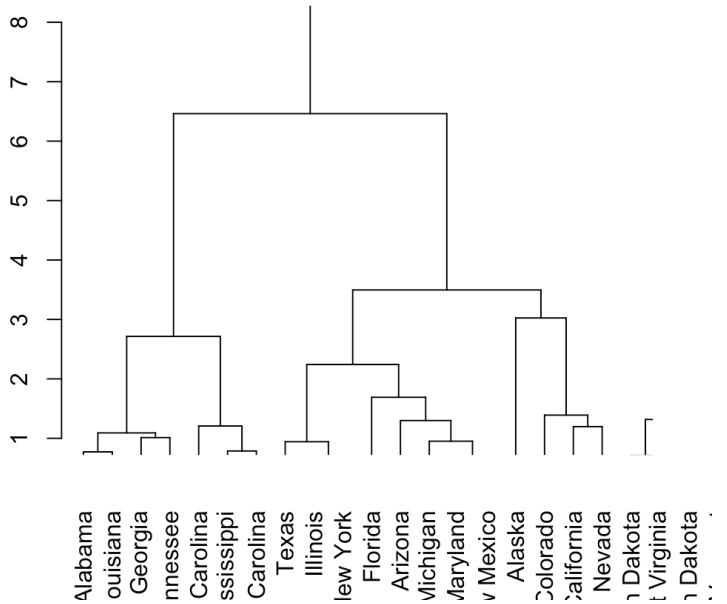
I.II Unsupervised Machine Learning

- No labels are given to the learning algorithm, leaving it on its own to find structure in its input
- Examples:
  - Group objects according to their shape

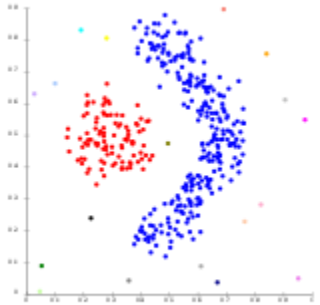


I.II Unsupervised ML **Clustering**

- Discovering clusters
  - Clustering data into groups
- Hierarchical clustering
- Partitional clustering



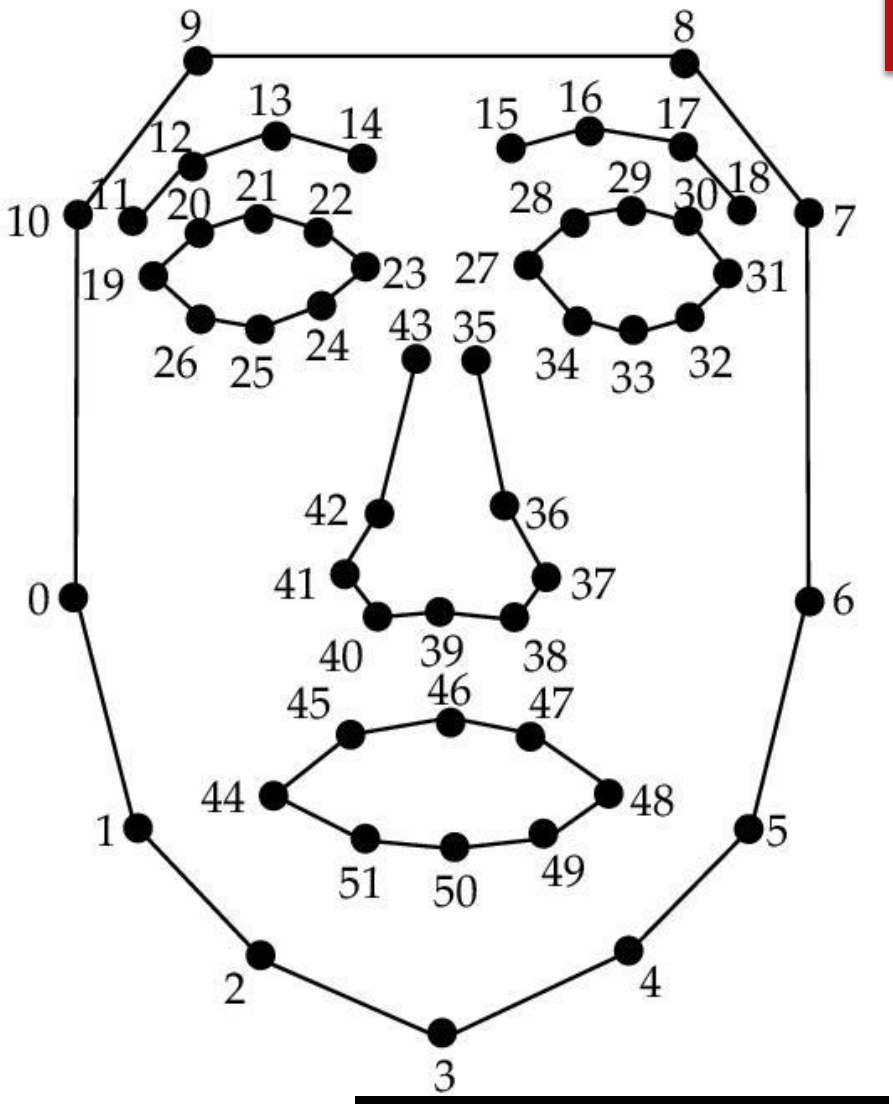
Hierarchical clustering



Partitional Clustering

I.II Unsupervised ML **Discovering Latent Factors**

- Discovering latent factors
  - Dimensionality reduction: reduce high dimensional data to a lower dimensional subspace.
    - E.g: there are many features representing an images, but just few are important: lighting, pose, identity…



I.II Unsupervised ML **Matrix Completion**

- If we have missing data, then the goal of this task is filling the gaps with numbers based on the surrounding values
- Tasks:
  - Image inpainting
  - Collaborative filtering
    - E.g: Given rating matrix of users with watched movies, predict how likely they would rate an unwatched one.

Figure 11: Examples of object removal and image editing using our EdgeConnect model. (Left) Original image. (Center) Unwanted object removed with optional edge information

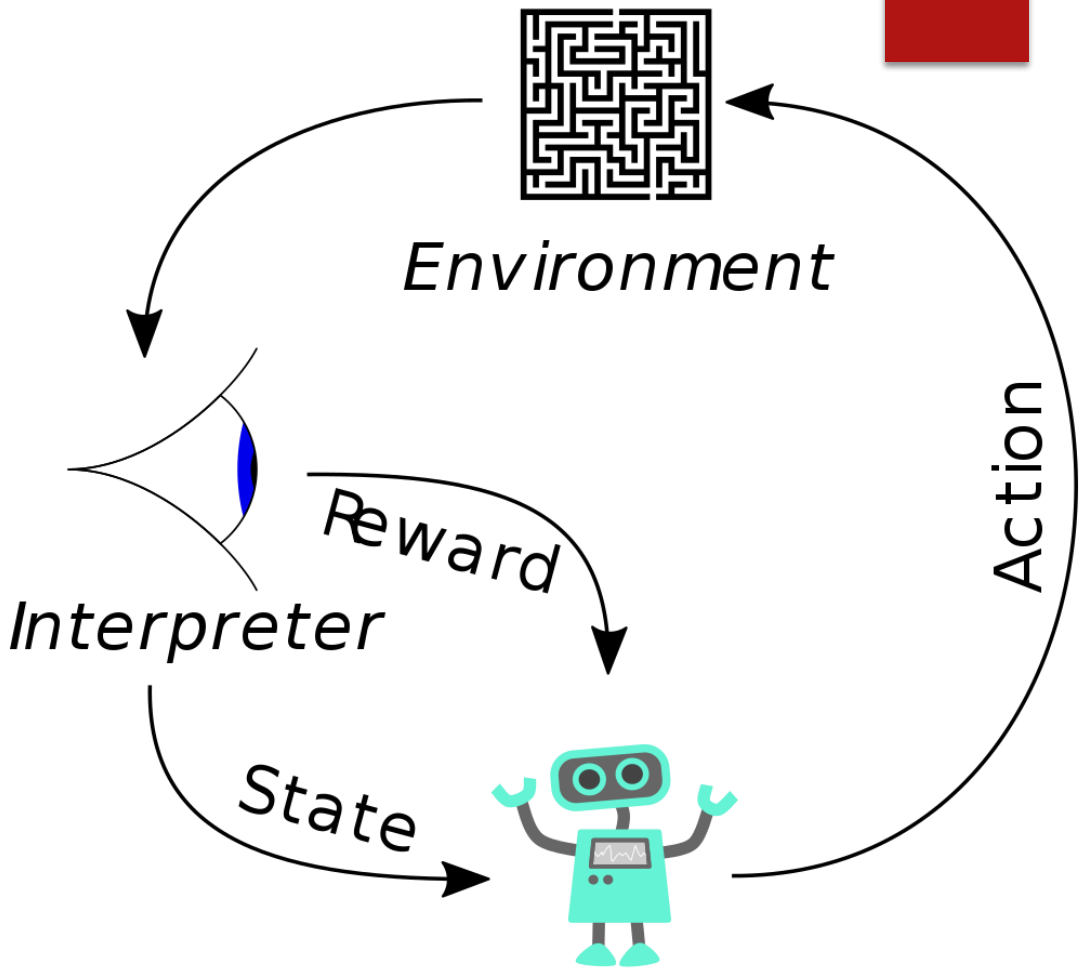
	target	movie_0	movie_1	movie_2	movie_3	movie_4
0	?	1	2	4	4	6
1	?	2	6	3	4	2
2	?	2	3	1	5	3
3	?	1	3	1	2	4
4						

I.III Semi-supervised Learning

- If you have a small set of labelled data, and one to use them as a seed to grow the data bigger:
  - Train your supervised model on that small dataset
  - Run prediction on the bigger data set and get labels
  - Revise the quality
  - Use that data as training data…

I.IV Reinforcement Learning

▶Reinforcement learning (RL) is an area of machine learning concerned with how software agents ought to take actions in an environment in order to maximize the notion of cumulative reward.



Source: wikipedia - https://en.wikipedia.org/wiki/Reinforcement_learning

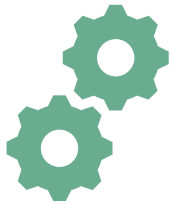
Part II: Data Analytics – What to do?

- Here are questions you have to answer:
  - What problem you are going to solve?
  - Is the data in good quality and in the format that you can load into memory and work with?
  - Is it big enough?
  - What model am I going to use?
  - How do I improve the performance of the model I am using?
  - How do I evaluate the result?
  - Visualize results?

II.I Data Analytics – What to do?
Preprocessing

Filtering, cleaning up data

Data conversion



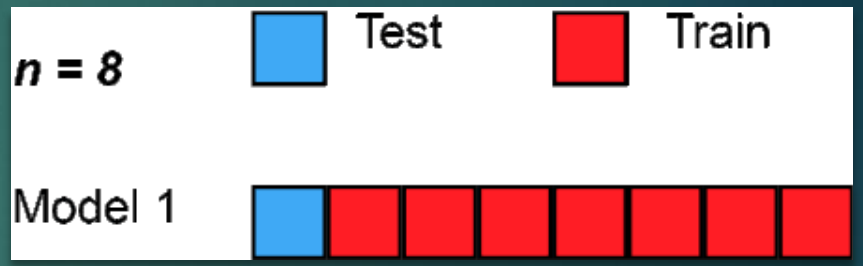
Feature extraction

Feature engineering:

- Extract more features from given data
- Transform the features (feature scaling)

II.I Data Analytics – What to do? **Model Selection**

- Split data into training set, test set, validation set (optional) if your problem is supervised
- Choose some models with different hyper-parameters and train them on the training set.
  - To tune the hyper-parameters, run it on validation set or n-fold validation on training set
  - Never do this:
    - Train and test on the same data set -> over fitting your model
    - Train on training set and test on test set but repeatedly until archive the best set of hyper-parameters -> overfitting



II.I Data Analytics – What to do? **Evaluation**

- Regression:
  - mean square error, root mean square error
    - $mse = \frac{\left(\sum_i (y_i - y_i^*)^2\right)}{n}$
  - Mean absolute error
    - $mae = \sum_i \frac{|y_i - y_i^*|}{n}$
  - R square:
    - $R^2 = 1 - \frac{\sum_i (y_i - y_i^*)^2}{\sum_i (y_i - \bar{y})^2}$

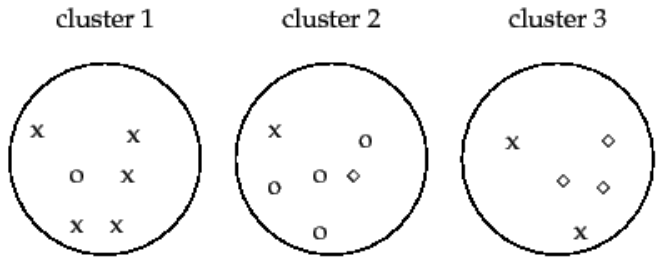
II.I Data Analytics – What to do? **Evaluation**

- Classification: Precision, Recall, F-1
  - True Positive: accurately classified as positive label (TP)
    - E.g: AIDS positive true, corona positive true
  - True negative: accurately classified as negative label (TN)
    - E.g: AIDS negative true, corona negative true
  - False Positive: wrongly classified as positive label (FP)
    - E.g: don't have AIDS, corona but got positive test
  - False Negative: wrongly classified as negative label (FN)
    - E.g: have AIDS, corona but got negative result

- ✓ $Precision = \frac{TP}{TP+FP}$,
- ✓ $Recall = \frac{TP}{TP+FN}$,
- ✓ $F1 = \frac{2(P \times \mathrm{R})}{P+R}$,
- ✓ $\mathrm{accuracy} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$

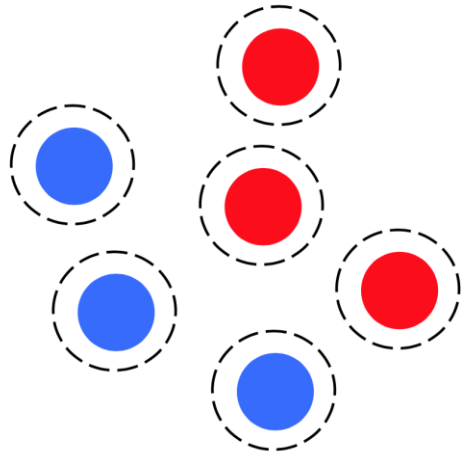
II.I Data Analytics – What to do? **Evaluation**

- Clustering:
  - $Purity(\Omega, C) = \frac{1}{N}\sum_k \max_j |\omega_k \cap c_j|$
    - N is number of members
    - Ω is a set of result-clusters
    - C is a set of gold standard clusters
  - B-cubed (B^3)
    - $Precision = \frac{1}{N}\sum_{p \in Dataset} Precision(p)$
    - $Recall = \frac{1}{N}\sum_{p \in \text{Dataset}} Recall(p)$
    - $F - Score = \frac{1}{N}\sum_{p \in \text{Dataset}} F(p)$
    - Where n = #clusters

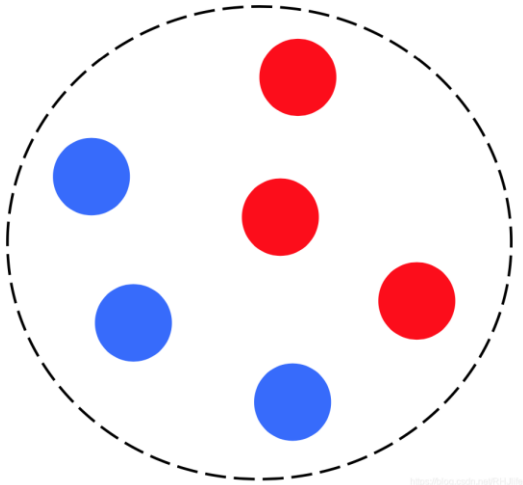


► **Figure 16.1** Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and ◇, 3 (cluster 3). Purity is $(1/17) \times (5+4+3) \approx 0.71$.

100% Precision, 33% Recall



50% Precision, 100% Recall,



II.I Data Analytics – What to do? **Visualization**

- Features visualization
  - Sub-space selection – our intuition is limited to 3-d space
- Result visualization
  - Selection of positive results
  - Selection of negative results (no system is perfect, even human)
  - Be creative! People love to see charts than table of numbers

Part II: Big Data

- If you have a big data (how big should be considered as big, well say 5TB data with the RAM and Hard Drive capacity at the moment – 2020), then you have to concern the technical issue to work with it as you cannot load them all to memory to analyze
- Solution: "divide and conquer", take advantage of parallel computing
- There are many techniques out there, but the most famous in the recent years is: Hadoop (Map – Reduce)

Materials and Classes

- Books:
  - Machine Learning – Tom Mitchell
  - Machine Learning A probabilistic Perspective – Kevin P. Murphy (Advanced)
  - Timeseries analysis – Hamilton
  - Applied Multivariate Statistical Analysis – Richard A. Johnson and Dean W. Wichern
- Online course:
  - Machine Learning – Andrew Ng (Recommended)
    - https://www.coursera.org/learn/machine-learning
  - DeepLearning.ai – Andrew Ng (Recommended)
    - https://www.coursera.org/specializations/deep-learning
  - Tip:
    - Select audit mode to attend those courses without certificate (and so they're free)

Part III Practice

PLAYING WITH REGRESSION

PLAYING WITH CLASSIFICATION

Part III
Practice
General TIPS

Import libraries

from library **import** classes, functions, defintions…

Import library

Numpy is library for matrix computation

Pandas is for loading structured data into DataFrame (similar concept as excel sheet with rows and columns), Series

Matplotlib is data visualization library

Sklearn is library for basic machine learning algorithms

Part III Practice
General TIPS

- Dataset:
  - https://scikit-learn.org/stable/datasets/index.html
- Loading dataset:
  - Regression:
    - from sklearn.datasets import load_boston
    - X, y = load_boston(return_X_y=True)
  - Classification:
    - sklearn.datasets.load_breast_cancer

Part III Practice
General TIPS

- Feature Engineering:
  - Scaling and transformation:
    - `from sklearn import preprocessing`
  - More here:
    - https://scikit-learn.org/stable/modules/preprocessing.html

Part III Practice
General TIPS

Linear Regression:	• https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
Logistic Regression (it's actually a classifier)	• https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
Support Vector Machine	• Regressor: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html • Classifier: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Part III Practice
General TIPS

- Train and predict
  - Choose your model: model = YourModel(hyper_parameters)
  - Fitting model with training set: model.fit(x,y)
  - Predict:
    - y_hat = model.predict(x_test)
    - Remember to scale the x_test before predict, if you scaled your training features
- Evaluation:
  - https://scikit-learn.org/stable/modules/model_evaluation.html