An Introduction to Machine Learning with Scikit-learn

Speaker:

Dr. Venkateswara Raoo,

NIT Warangal

Note: These slides were assembled by Dr. K. V. Rao, with grateful acknowledgement of the many others who made their course materials available online.

Outline

- Introduction to Machine Learning
  - Learning
  - Types of Machine Learning
- Different Problems and Solutions
- Implementation of ML algorithms using Scikit-learn

Learning ?

- We say, we are learning something when the performance is improving with our experience.
- Learning = Improving with experience at some task.
- Human’s Learn from experience.

Machine Learning

- Machine Learning?
  - Improve over task *T*.
  - With respect to performance measure *P*.
  - Based on experience *E*.

- What are *T*, *P*, *E* ? How do we formulate a machine learning problem?
- Handwriting recognition
  - *T* – classifying handwritten words within images.
  - *P* – percent of words correctly classified.
  - *E* – database of handwritten words with given classifications.
- Robot Driving
  - T – Driving on public four-lane highways using vision sensors.
  - P – Average distance traveled before an error (human supervisor).
  - E – sequences of images and steering commands recorded observing a human driver.

Machine Learning

- Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

- **The primary aim is to allow the computers learn automatically** without human intervention or assistance and adjust actions accordingly.

Source: https://towardsdatascience.com/introduction-to-machine-learning-f41aabc55264

Classic Approaches vs Machine Learning

- Let's say we want to predict the price of a house based on the size of the house, the size of its garden, and the number of rooms it has.

Size of house

Size of garden

Number of rooms

</> CLASSICAL APPROACH

$ = 1.2 × + 0.7 × + 3.1 ×

House pricing formula is known and explicitly coded

ML APPROACH

$ = A × + B × + C ×

Model with unknown A, B and C to be defined

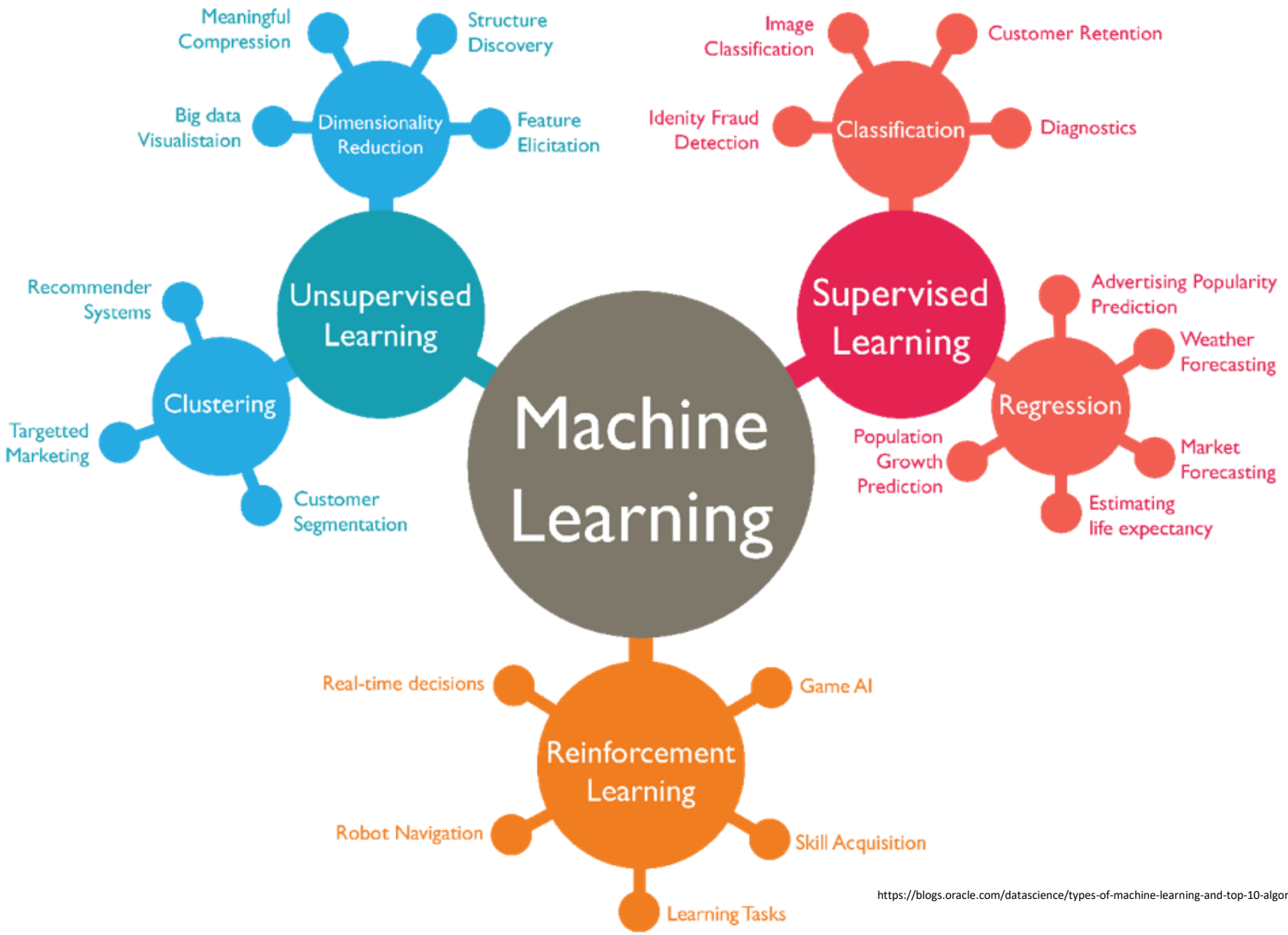
Available data to determine A, B and C
(to fit the model)

Why is Machine Learning Important?

- Explosive growth of data (click here: *https://www.thinkful.com/blog/what-is-data-science)*
- Data is the lifeblood of all business.
- Data-driven decisions increasingly make the difference between keeping up with competition or falling further behind.
- Machine learning can be the key to unlocking the value of corporate and customer data and enacting decisions that keep a company ahead of the competition.

- **Machine Learning Use Cases**
- **Manufacturing.** Predictive maintenance and condition monitoring
- **Retail.** Upselling and cross-channel marketing
- **Healthcare and life sciences.** Disease identification and risk satisfaction
- **Travel and hospitality.** Dynamic pricing
- **Financial services.** Risk analytics and regulation
- **Energy.** Energy demand and supply optimization
- And many more.

Types of ML algorithms



https://blogs.oracle.com/datascience/types-of-machine-learning-and-top-10-algorithms-everyone-should-know-v2

Scikit Learn

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

- https://scikit-learn.org/stable/

- Installation Instructions
  - https://scikit-learn.org/stable/install.html#installation-instructions

Source: https://towardsdatascience.com/introduction-to-machine-learning-f41aabc55264

Supervised Learning

Supervised Learning

- Applications in which the data comprises examples of the input variables along with their corresponding target values/vectors are known as supervised learning problems.

- Supervised learning algorithms try to model relationships between the target prediction output (y) and the input features (**x**) such that we can predict the output values for new data based on those relationships.

- **Problem:** Given a the training set of pairs $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, where $\mathrm{x_i} \in R^d$ and $\mathrm{y_i}$ is a target variable, the task is to predict for $x_{n+j}, \mathrm{j} \geq 1$.

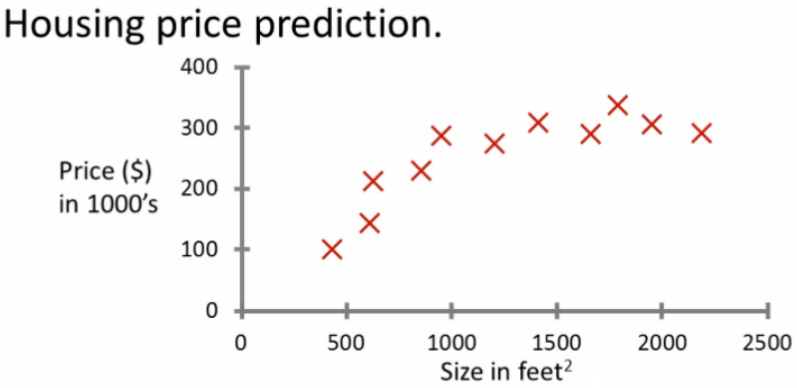
- The main types of supervised learning problems include **regression** and **classification** problems

- List of Common Algorithms include Nearest Neighbor, Naive Bayes, Decision Trees, Linear Regression, Support Vector Machines (SVM), etc.

Regression Problem

- Given a the training set of pairs $\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \ldots, \left(x^{(m)}, y^{(m)}\right)$, where $x^{(i)} \in R^d$ and $y^{(i)}$ is a **continuous** target variable, the task is to predict for $x^{(m+j)}, j \geq 1$.
- How do we predict housing prices
  - Collect data regarding housing prices and how they relate to size in feet.

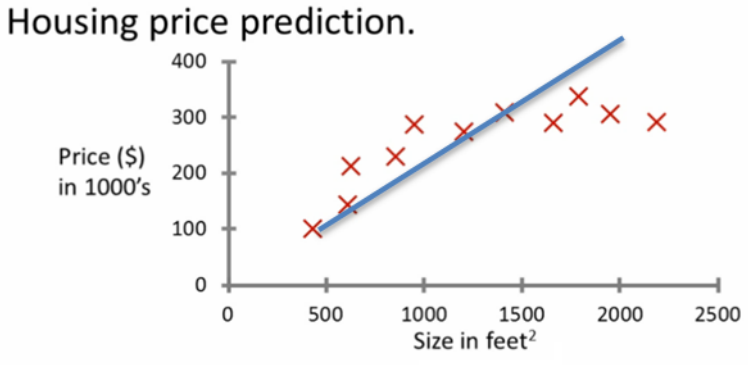
- "Given this data, a friend has a house 750 square feet - how much can they be expected to get?“



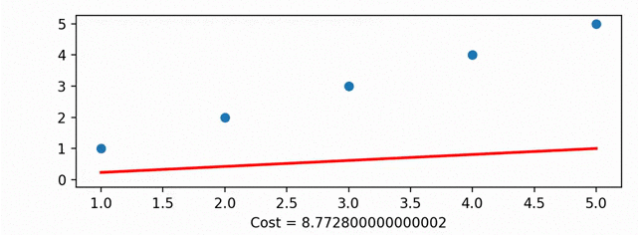
Linear Regression

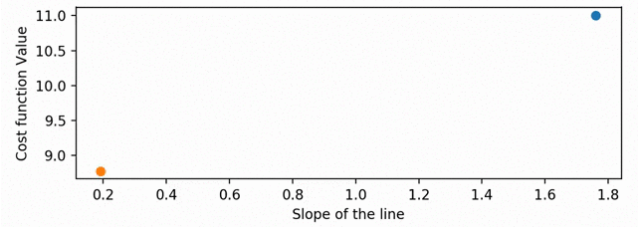
- Output is modelled as a linear combination of input variables.

- "Given this data, a friend has a house 750 square feet - how much can they be expected to get?“
  - Maybe $150 000



Linear Regression

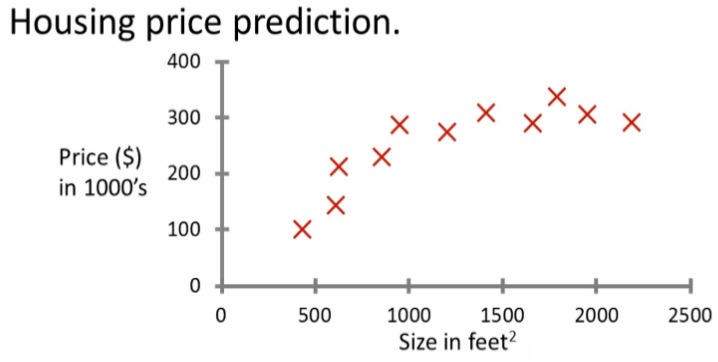




Source: https://towardsdatascience.com/machine-learning-algorithms-in-laymans-terms-part-1-d0368d769a7b

- Many lines are possible !!

- Which is the best line ?
  - The line that minimizes the difference between the actual and estimated prices.
- A cost function lets us figure out how to fit the best straight line to our data.
- What makes a line different ?
  - Parameters θ_0, θ_1
- What is our objective ?
  - Choose these parameters θ_0, θ_1 so that prediction $h_\theta(x)$ is close to ground truth y for our training examples, i.e. minimize the difference between h(x) and y for each/any/ every example.



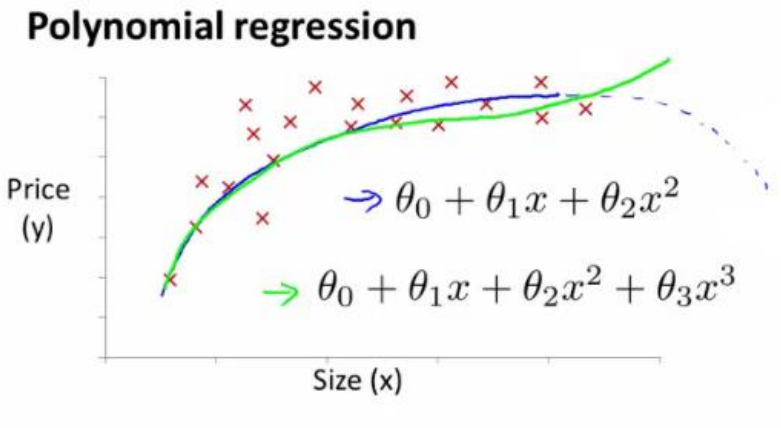
Linear Regression with Sklearn

- http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- http://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html
- http://scikit-learn.org/stable/modules/linear_model.html

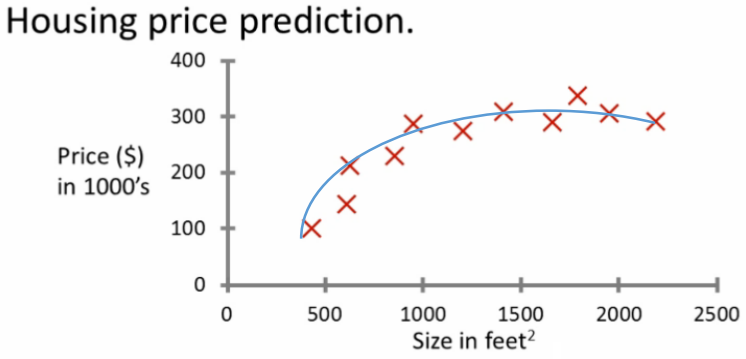
Polynomial Regression

- Output is modelled as a nonlinear function of input variables.

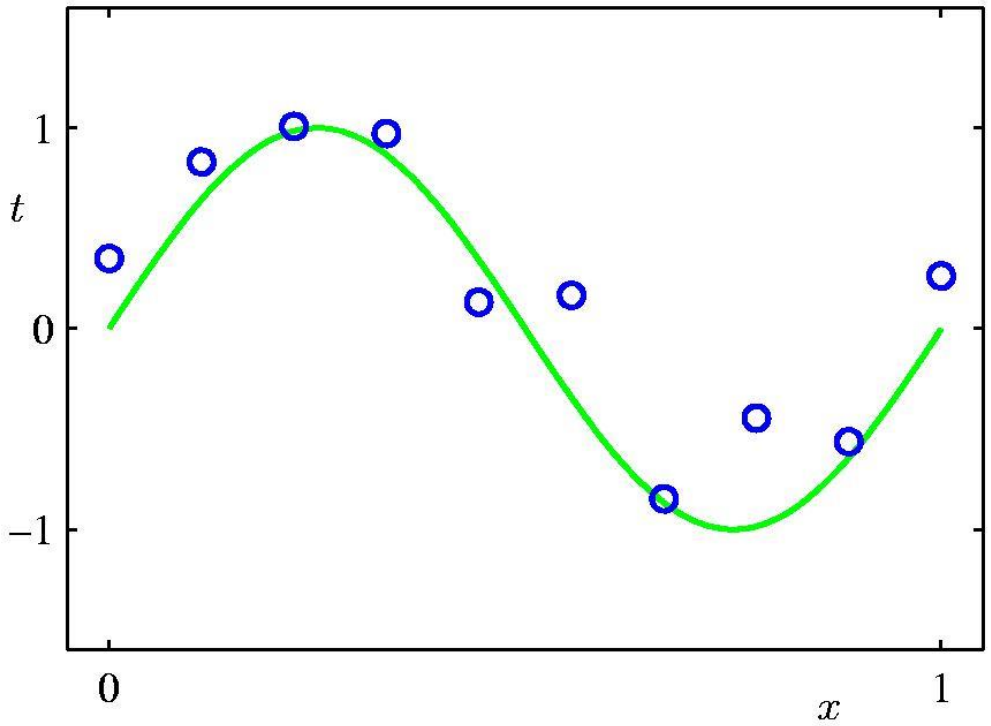
$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_M x^M$$



- "Given this data, a friend has a house 750 square feet, how much can they be expected to get?“
  - Second order polynomial
  - Maybe $200 000



Polynomial Curve Fitting



Plot of a training data set of m = 10 points, shown as blue circles, each comprising an observation of the input variable x along with the corresponding target variable t. The green curve shows the function sin(2πx) used to generate the data.

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_M x^M$$

What **M** should we choose?→**Model Selection**

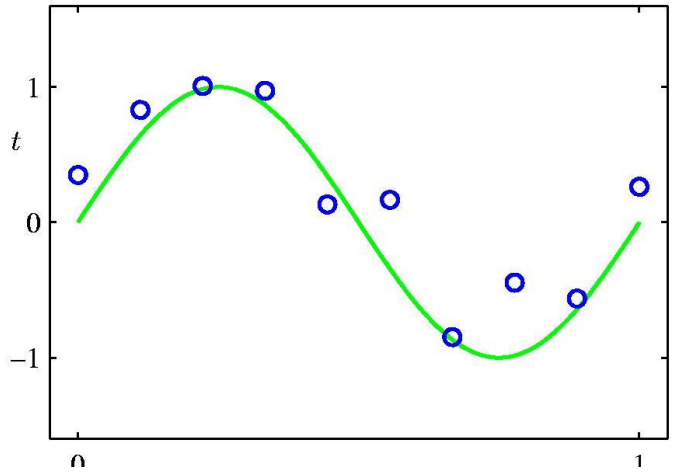
Given **M**, what $\theta's$ should we choose? →**Parameter Selection**

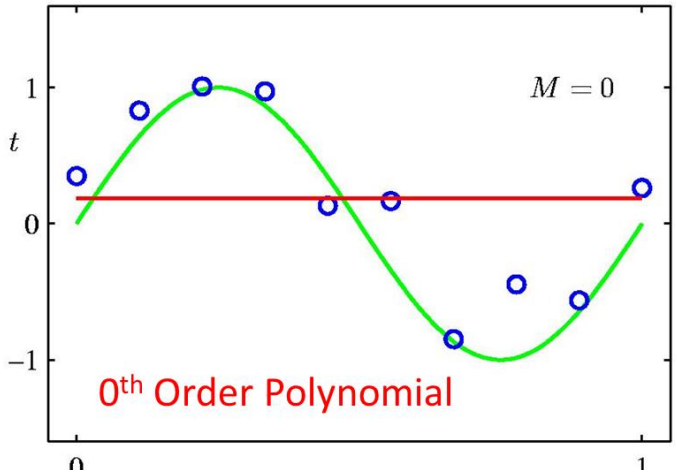
Regularization

- Penalize large coefficient values

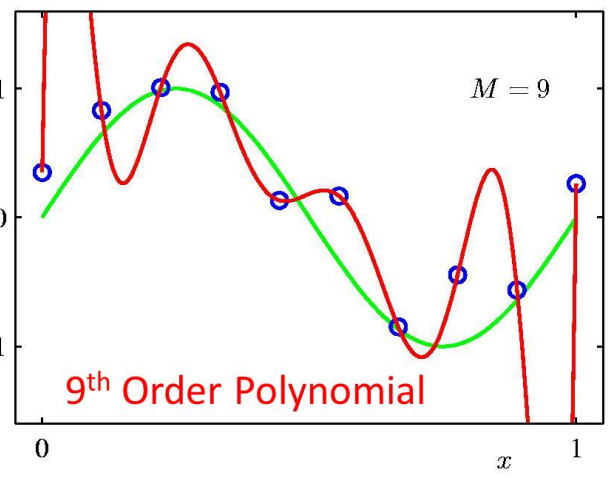
Loss function: $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^i\right) - y^i\right)^2 + \lambda \|\theta\|_l$

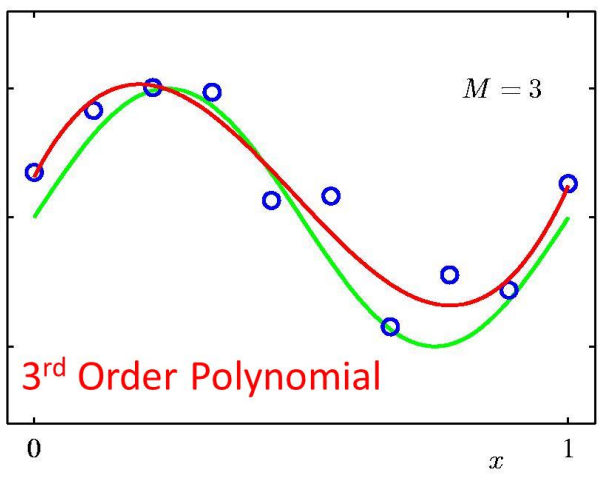
Idea: penalize high weights that contribute to high variance and sensitivity to outliers.

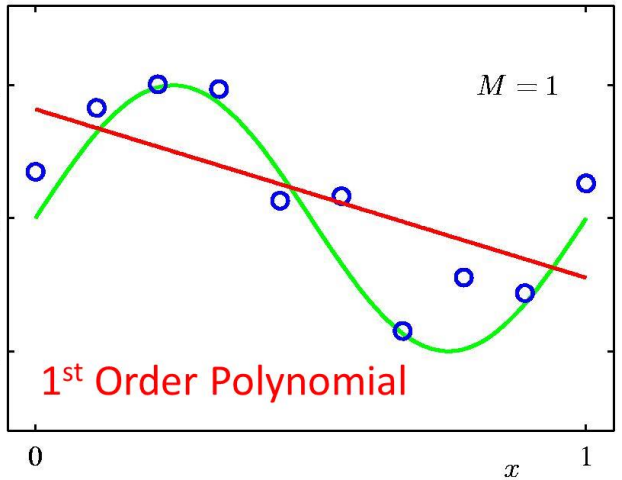




	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43



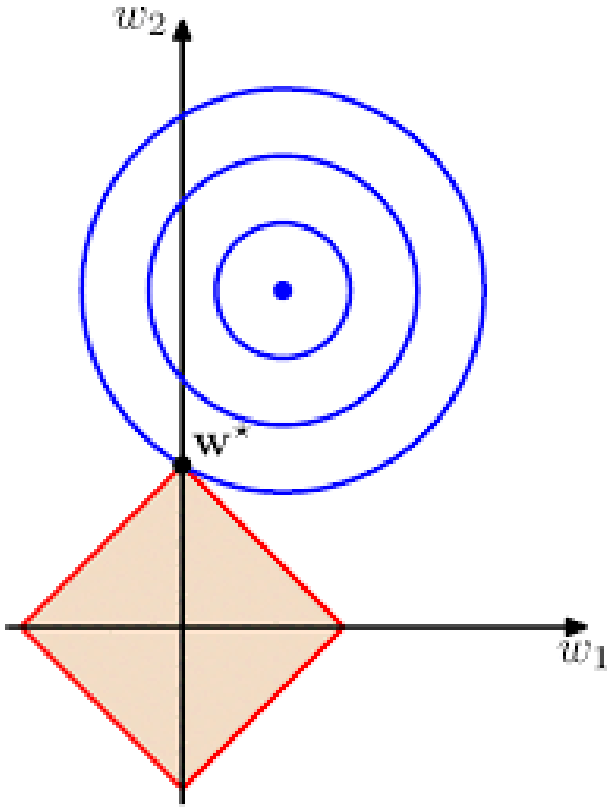




Lasso regression

- $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^i\right) - y^i\right)^2 + \lambda \|\theta\|_1$
- L1 - norm
- $\boldsymbol{w = \theta}$

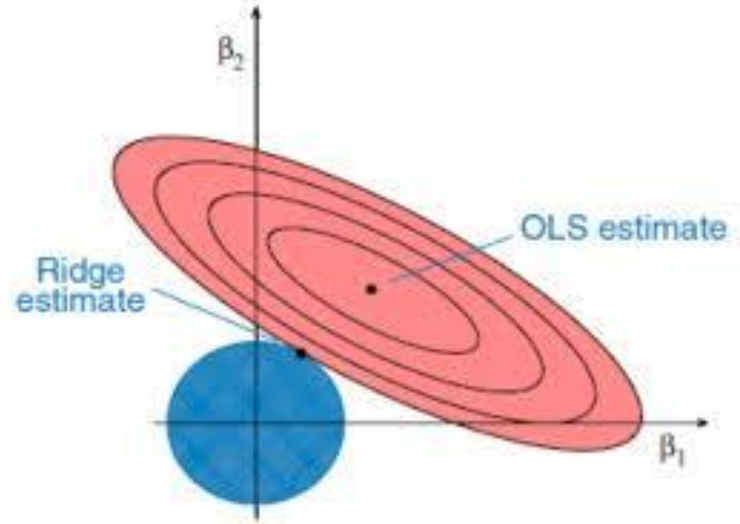
$$||x||_1 = \sum_i |x_i| = |x_1| + |x_2| + \ldots + |x_i|$$



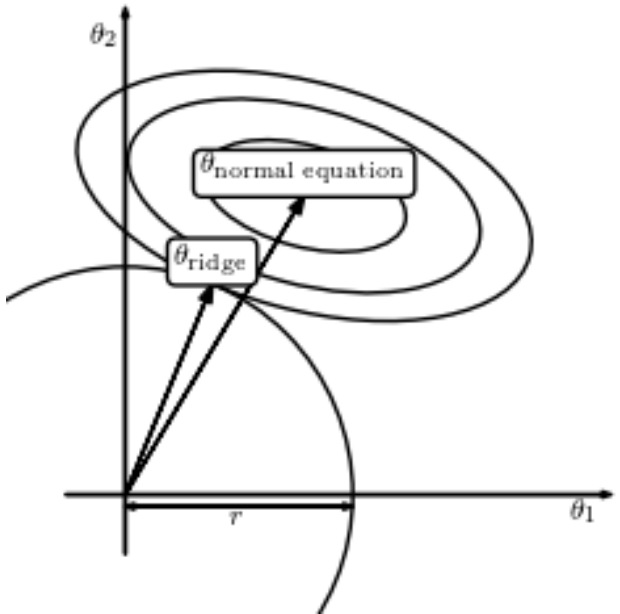
Ridge Regression

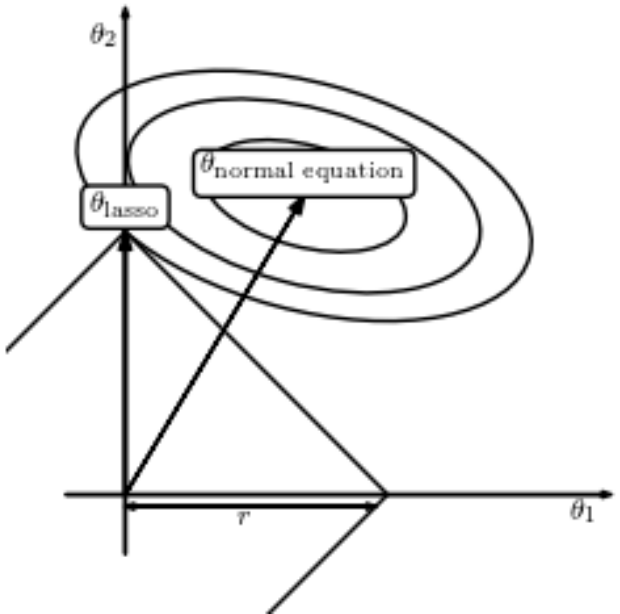
- $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i\right)^2 + \frac{\lambda}{2} \|\theta\|_2$
- L2-norm

$$\|x\|_2 = \sqrt{\left(\sum_i x_i^2\right)} = \sqrt{x_1^2 + x_2^2 + \ldots + x_i^2}$$



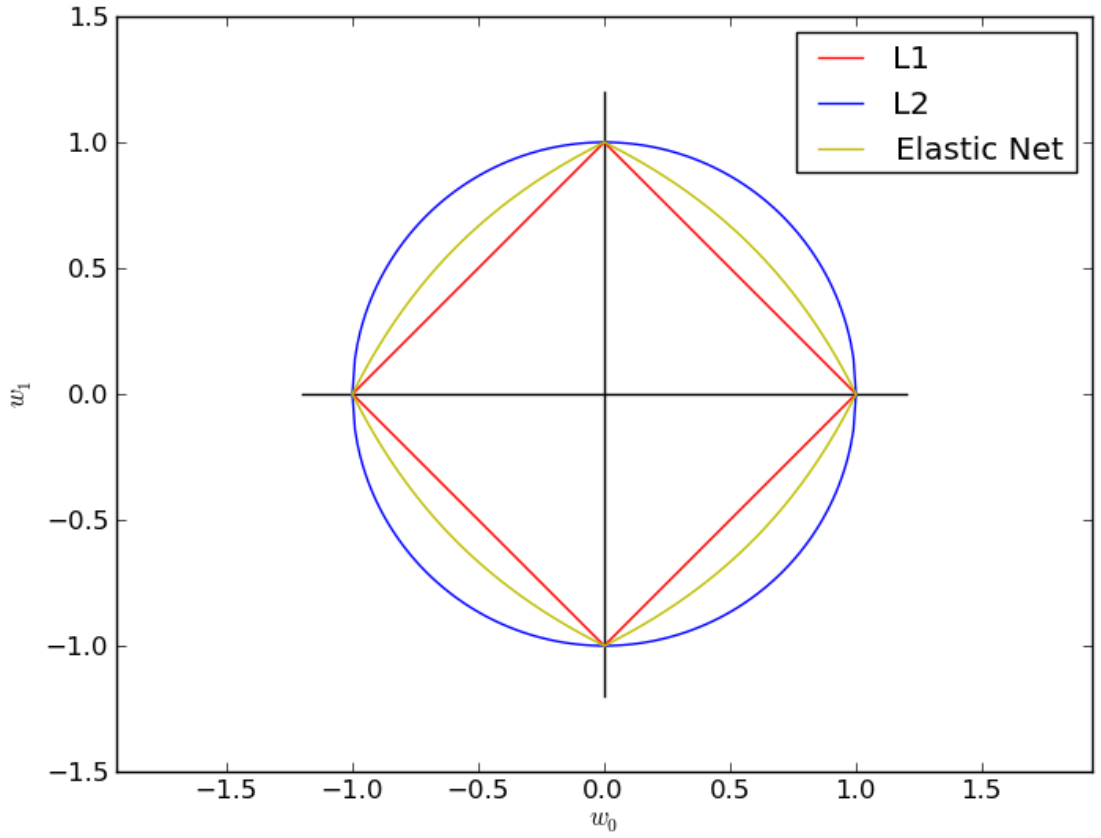
- $\boldsymbol{w - A - B}$





ElasticNet Regression

- $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^i\right) - y^i\right)^2 + \lambda_1 \|\theta\|_1 + \lambda_2 \|\theta\|_2$

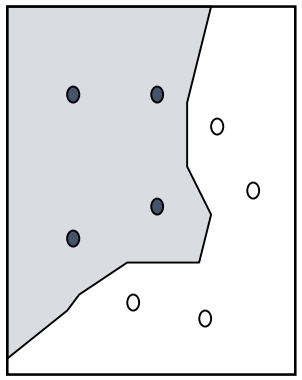


Polynomial Regression with Sklearn

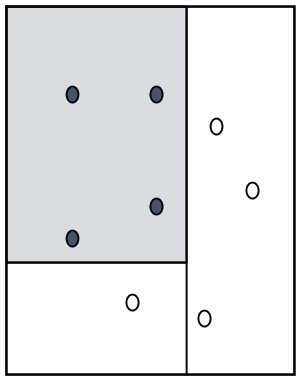
- http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html
- http://scikit-learn.org/stable/auto_examples/linear_model/plot_polynomial_interpolation.html

Classification

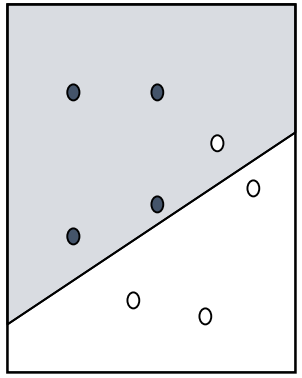
- Given a the training set of pairs $\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \ldots, \left(x^{(m)}, y^{(m)}\right)$, where $x^{(i)} \in R^d$ and $y^{(i)}$ is a **discrete** target variable, the task is to predict for $x^{(m+j)}, j \geq 1$.



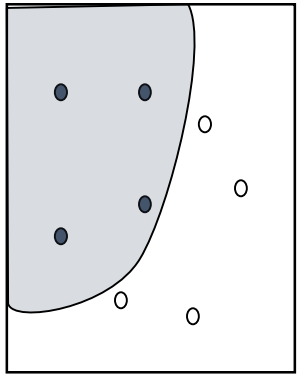
Nearest Neighbor



Decision Tree



Linear Functions



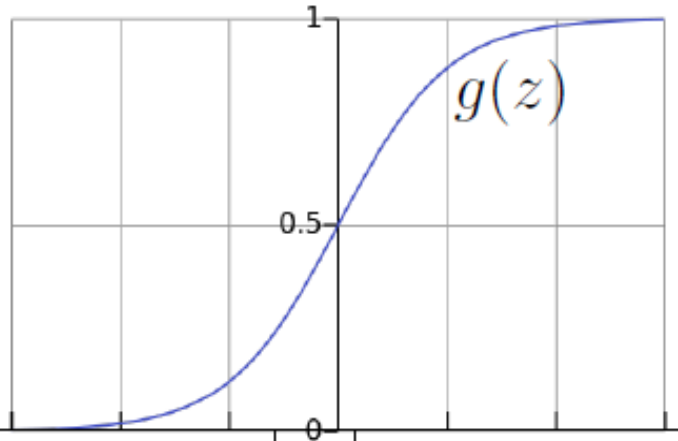
Nonlinear Functions

Logistic Regression for Classification

- A probabilistic approach to learn a classifier.
  - $h_\theta(x)$ should give $P(y = 1|x; \theta)$
  - $0 \leq h_\theta(x) \leq 1$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}\right)$$

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



$\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}$ should be large negative values for negative instances

$\boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{x}$ should be large positive values for positive instances

- Assume a threshold and
  - Predict y=1 if $h_\theta(x) \geq 0.5$
  - Predict y=1 if $h_\theta(x) < 0.5$

Logistic Regression

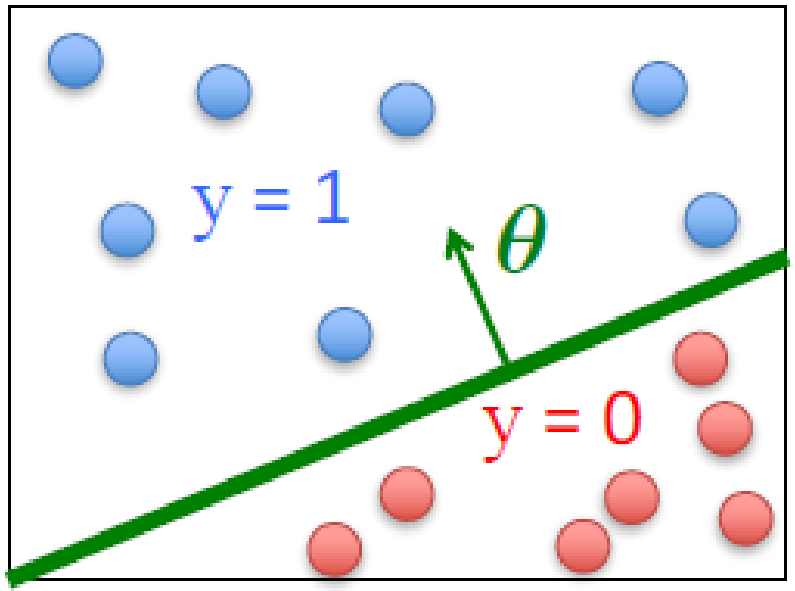
- Model:

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

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

$$\theta^T = [\theta_0 \ \theta_1 \ \ldots \theta_d]$$

$$\mathrm{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$$



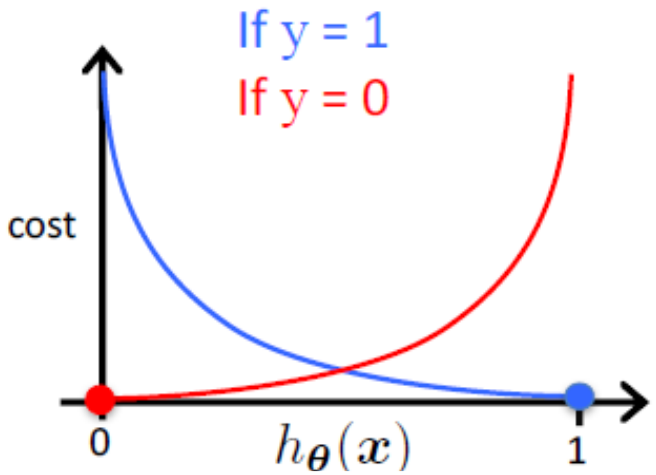
Logistic regression objective

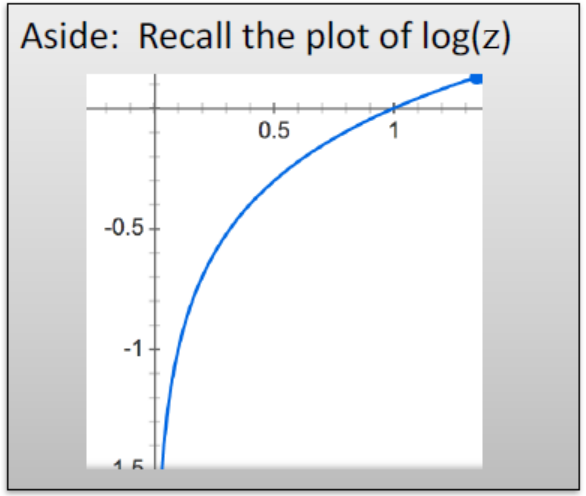
- Logistic regression objective

$$\min_{\theta} J(\theta)$$

$$J(\theta) = -\sum_{i=1}^{m} \left[y^i \log\, h_\theta(x^i) + (1 - y^i) \log\left(1 - h_\theta(x^i)\right) \right]$$

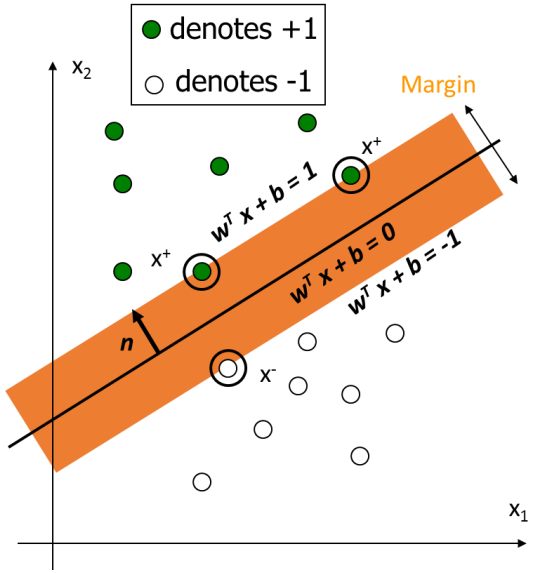
- $cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & if\ y = 1 \\ -\log(1 - h_\theta(x)) & if\ y = 0 \end{cases}$



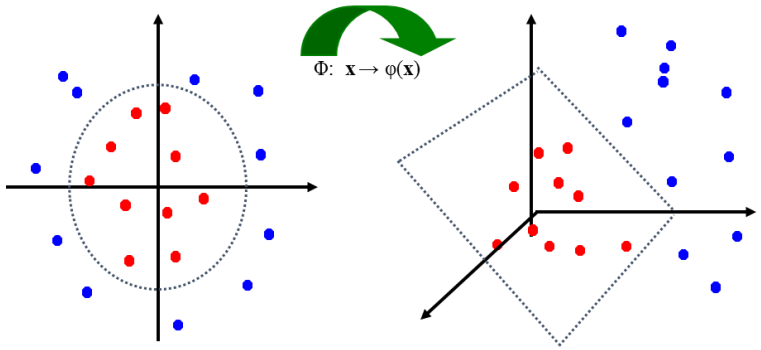


Support Vector Machines (SVMs)

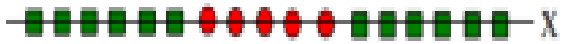
- Linear SVM

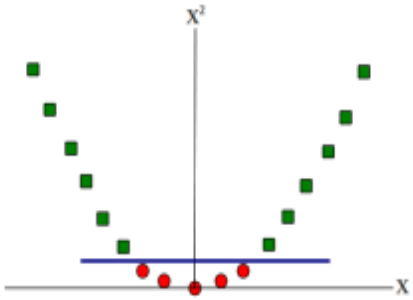


- Kernel SVM



$x \rightarrow \{x, x^2\}$



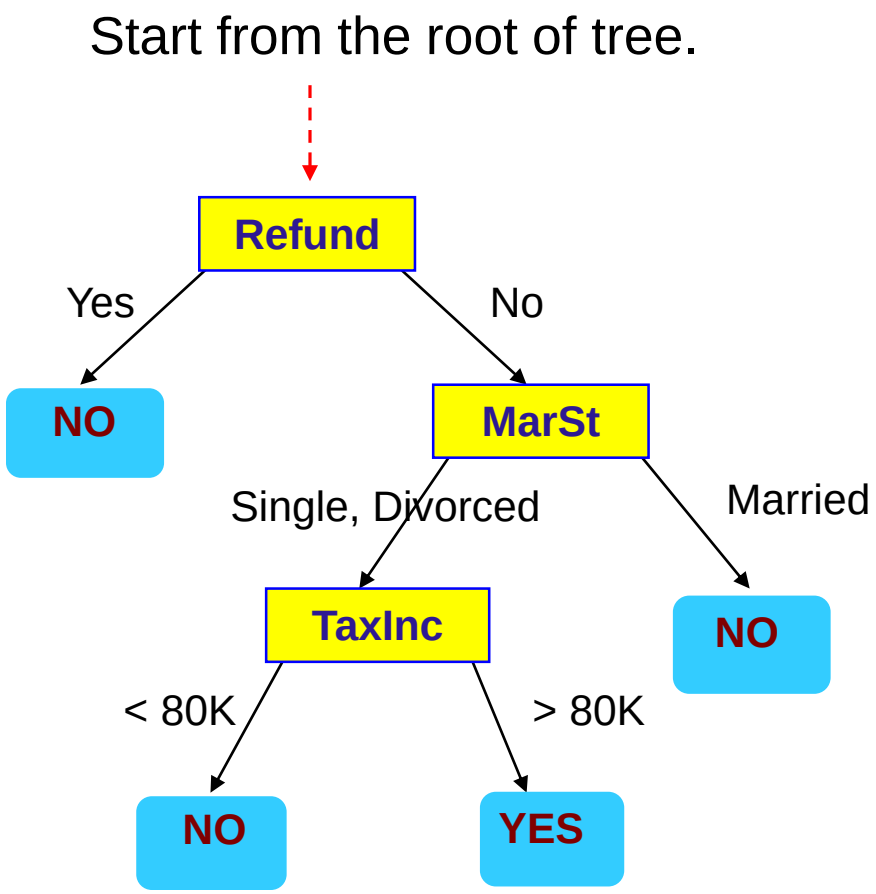


Linear in the new representation ≡ nonlinear in the old representation

Multiclass Classification

- SVM is essentially for 2-class classification
- Many approaches to handle multiple classes
  - Combining binary classifiers
    - One-against-all (OAA-SVM)
    - One-against-one (OAO-SVM)
    - Decision Tree Structure (DAG-SVM)
    - Binary Tree structure (BT-SVM)
    - Error correcting codes (ECOC)
  - Training a single classifier
    - Multi-class optimization
    - Constraint classification

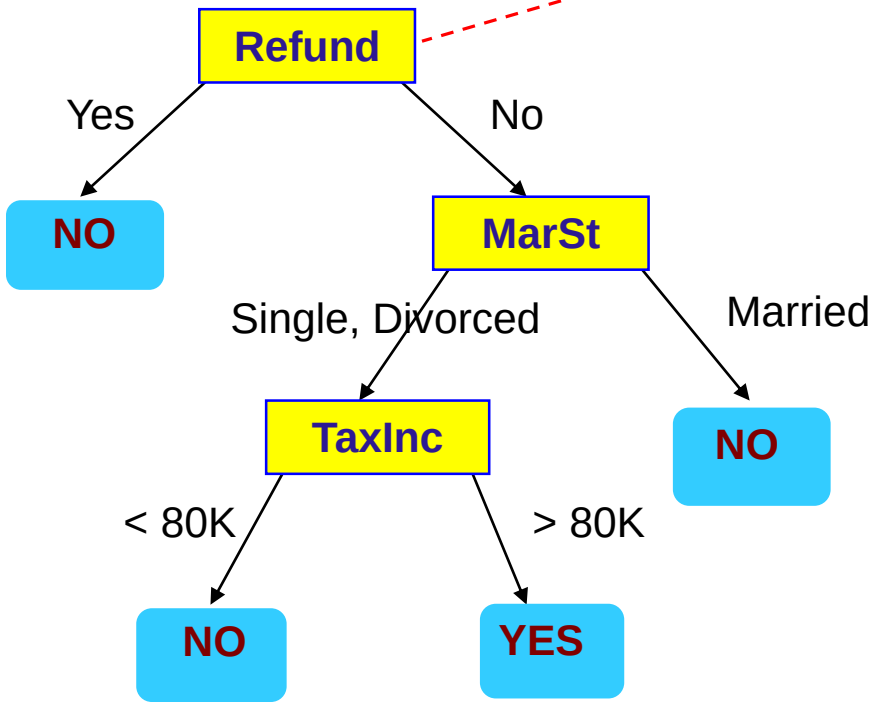
Decision Tree



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

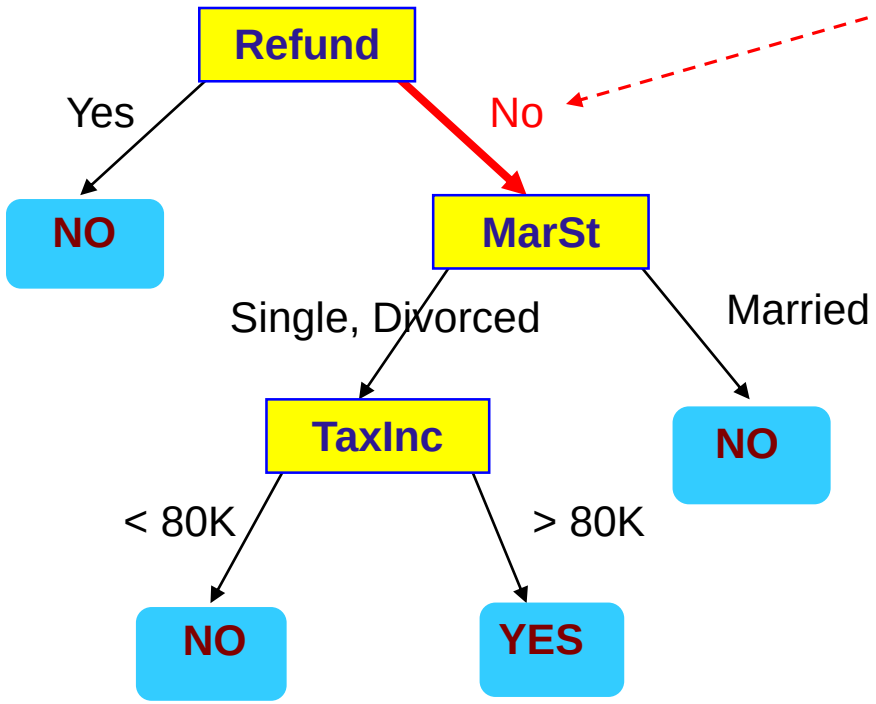
Decision Tree



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

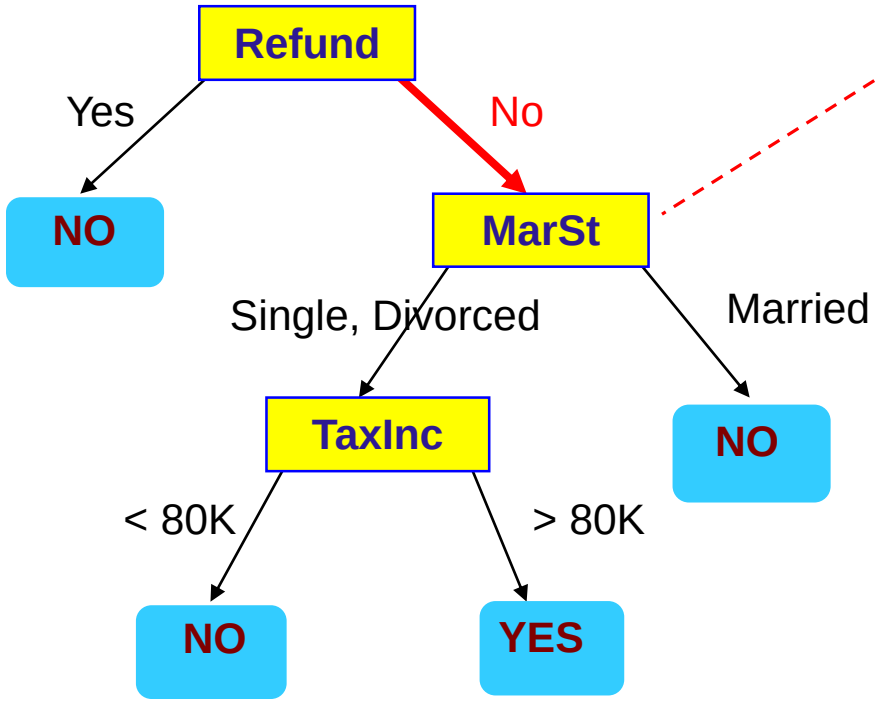
Decision Tree



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

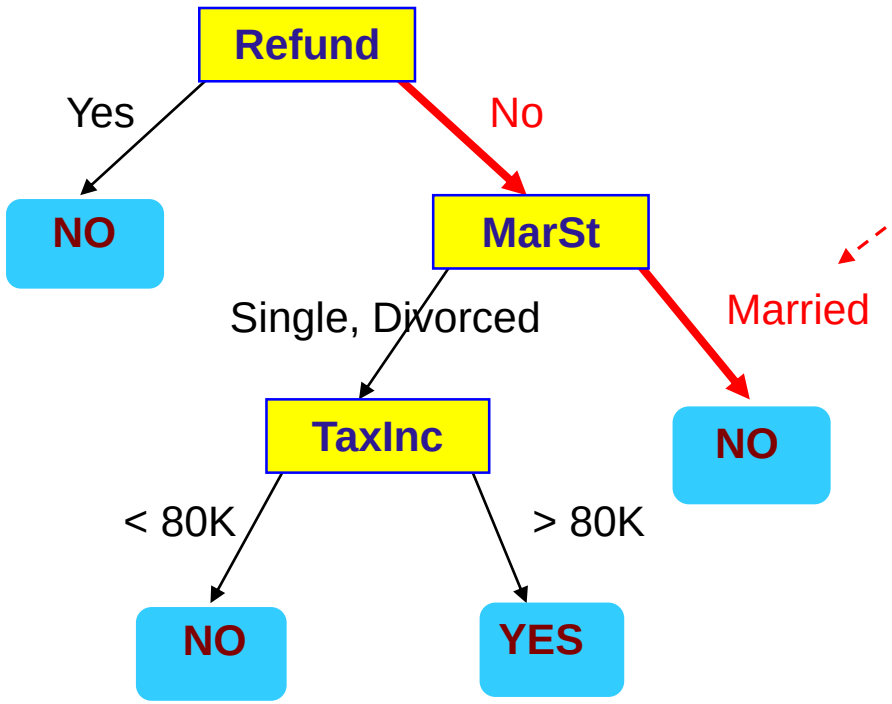
Decision Tree



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

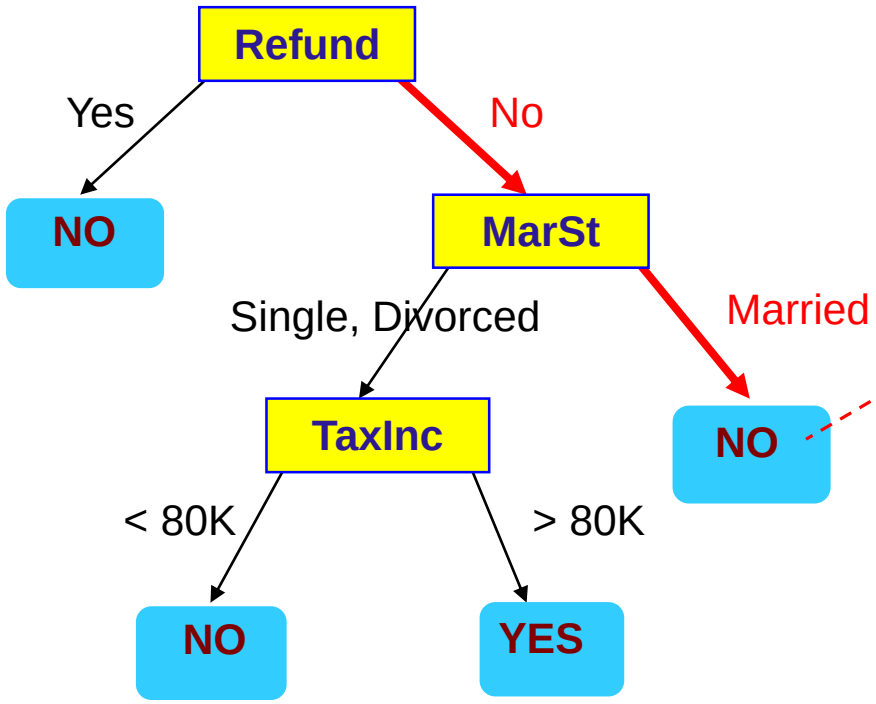
Decision Tree



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Decision Tree

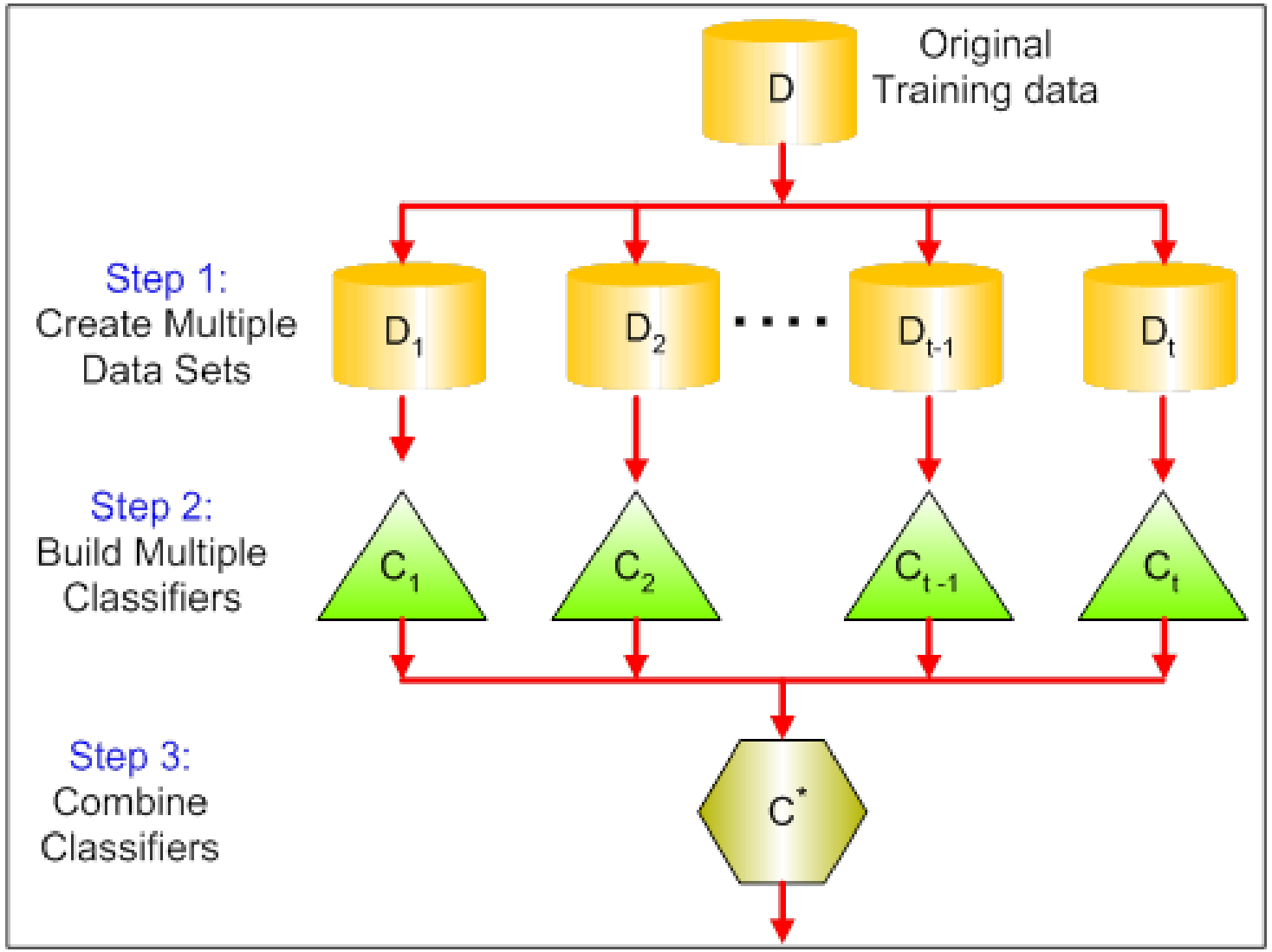


Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Assign Cheat to “No”

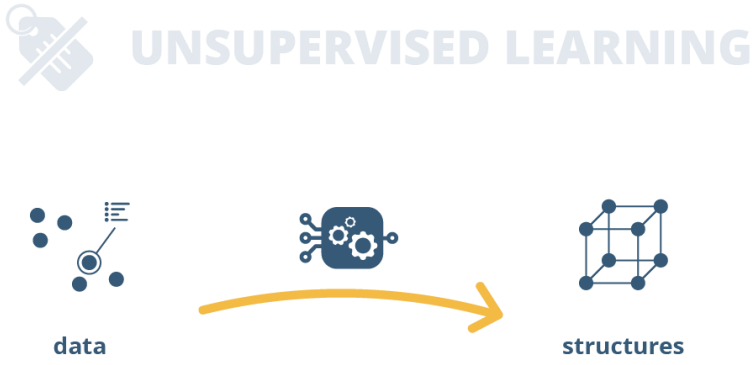
Ensemble Method: Bagging



Source: https://medium.com/@uttam94/bagging-and-boosting

Source: https://towardsdatascience.com/introduction-to-machine-learning-f41aabc55264

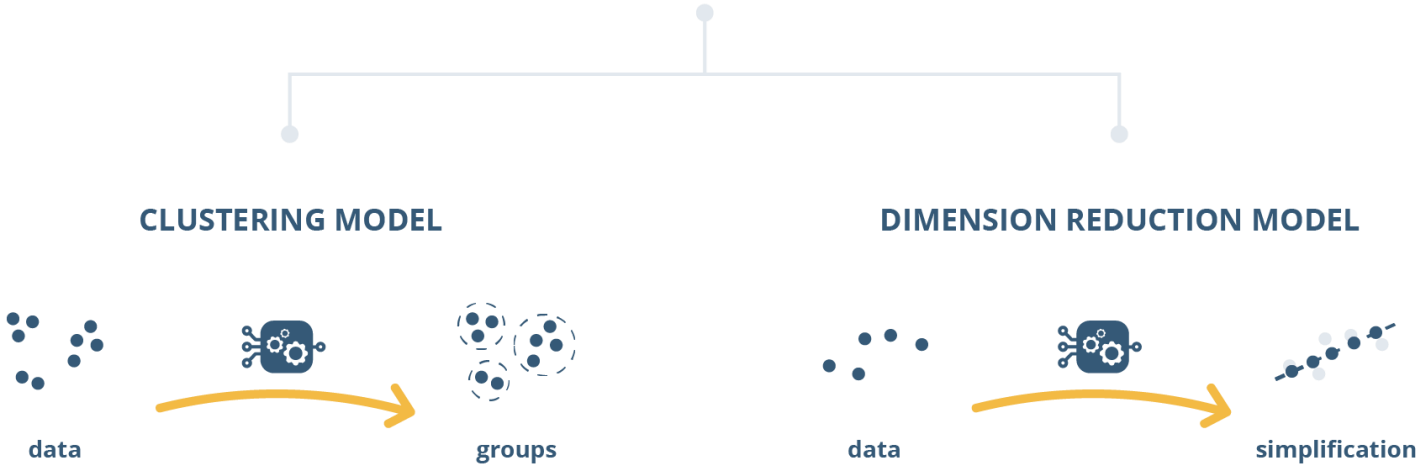
Unsupervised Learning



Unsupervised learning models find structures among all the data points described by their features.

In unsupervised learning, data have no labels and models try to answer questions like:

- are there groups among the data ?
- is it possible to simplify the data ?



Clustering models try to find groups among the data.

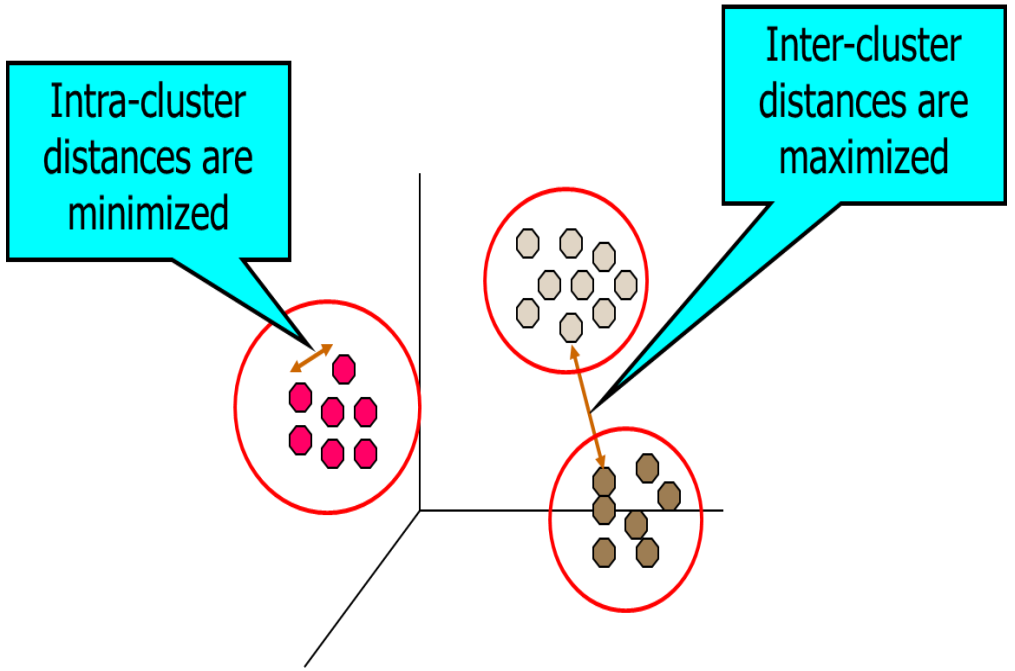
Dimension reduction models try to express data with a fewer number of dimensions.

Unsupervised Learning

- The computer is trained with unlabeled data.
- Useful in cases where the human expert doesn't know what to look for in the data.
- Family of machine learning algorithms which are mainly used in pattern detection and descriptive modeling.
- No output categories or labels here based on which the algorithm can try to model relationships.
- These algorithms try to use techniques on the input data to mine for rules, detect patterns, and summarize and group the data points which helps in deriving meaningful insights and describe the data better to the users.
- The main types of unsupervised learning algorithms include Clustering algorithms, Association rule learning algorithms, Dimensionality Reduction Techniques.

Clustering

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Clustering is a technique for finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups
- Clustering is often called an unsupervised learning task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.



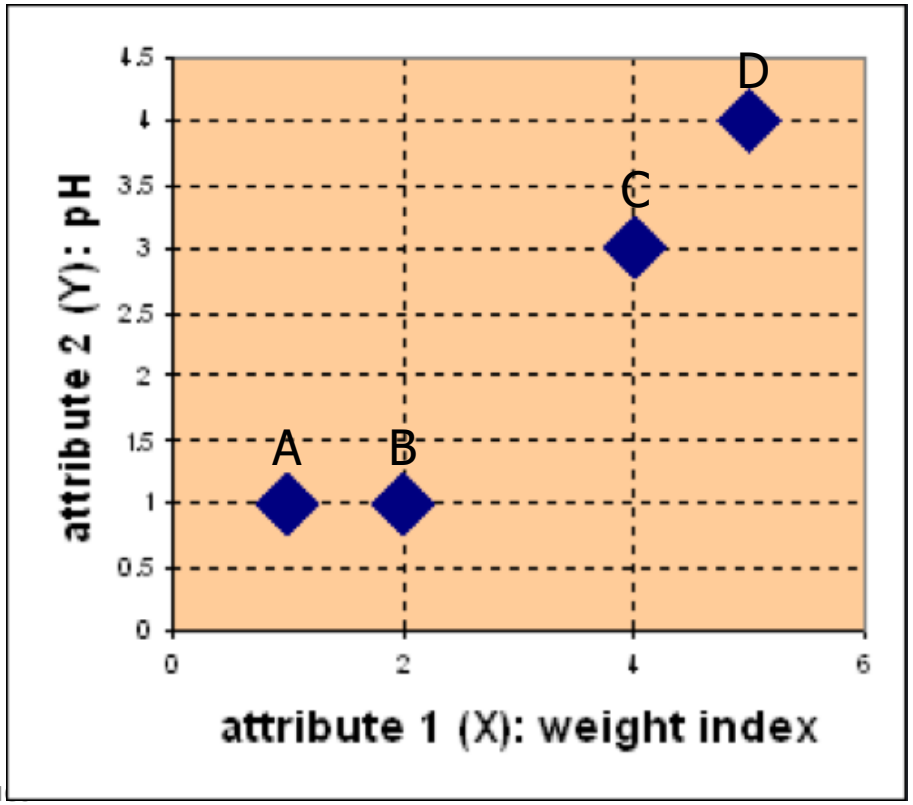
K-means algorithm

- Given the cluster number *K*, the *K-means* algorithm is carried out in three steps after initialization:
  - Initialization: set seed points (randomly)
  1. Assign each object to the cluster of the nearest seed point measured with a specific distance metric
  2. Compute new seed points as the centroids of the clusters of the current partition (the centroid is the centre, i.e., *mean point*, of the cluster)
  3. Go back to Step 1), stop when no more new assignment (i.e., membership in each cluster no longer changes)

K-means - Example

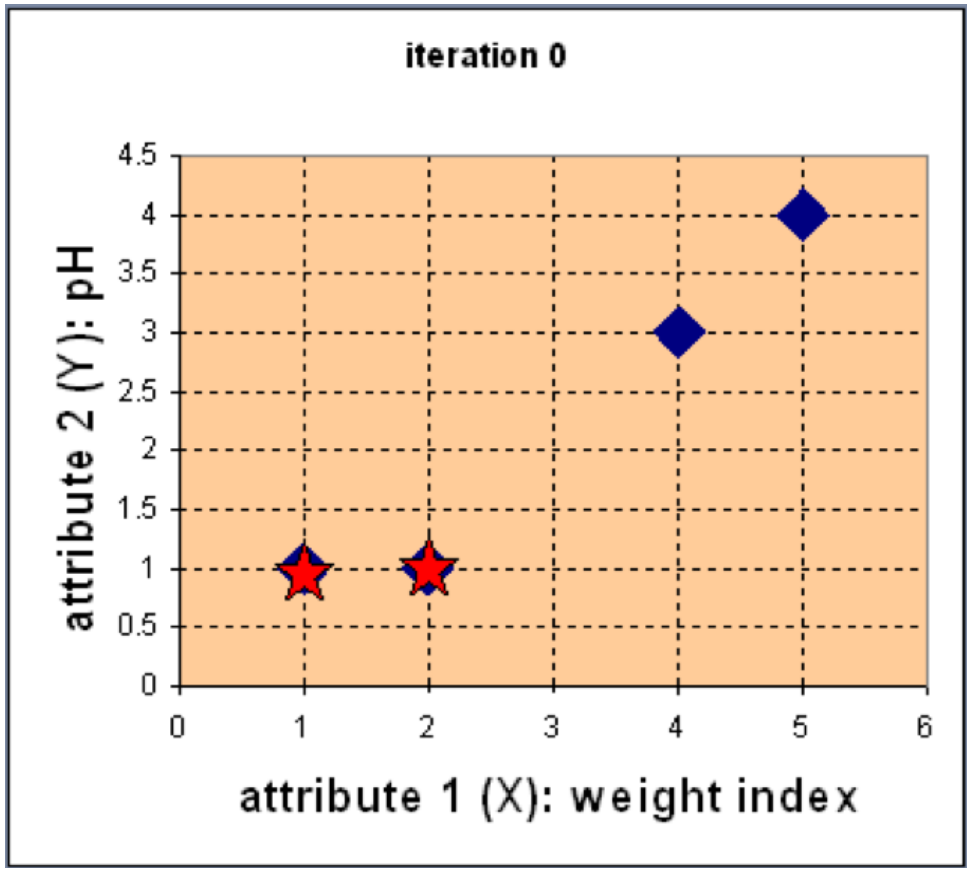
- Problem:
  - Suppose we have 4 types of medicines and each has two attributes (pH and weight index). Our goal is to group these objects into *K=2* group of medicine.

Medicine	Weight	pH-Index
A	1	1
B	2	1
C	4	3
D	5	4



K-means - Example

- Step 1: Use initial seed points for partitioning



$$c_1 = A, c_2 = B$$

$$D^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (2,1) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

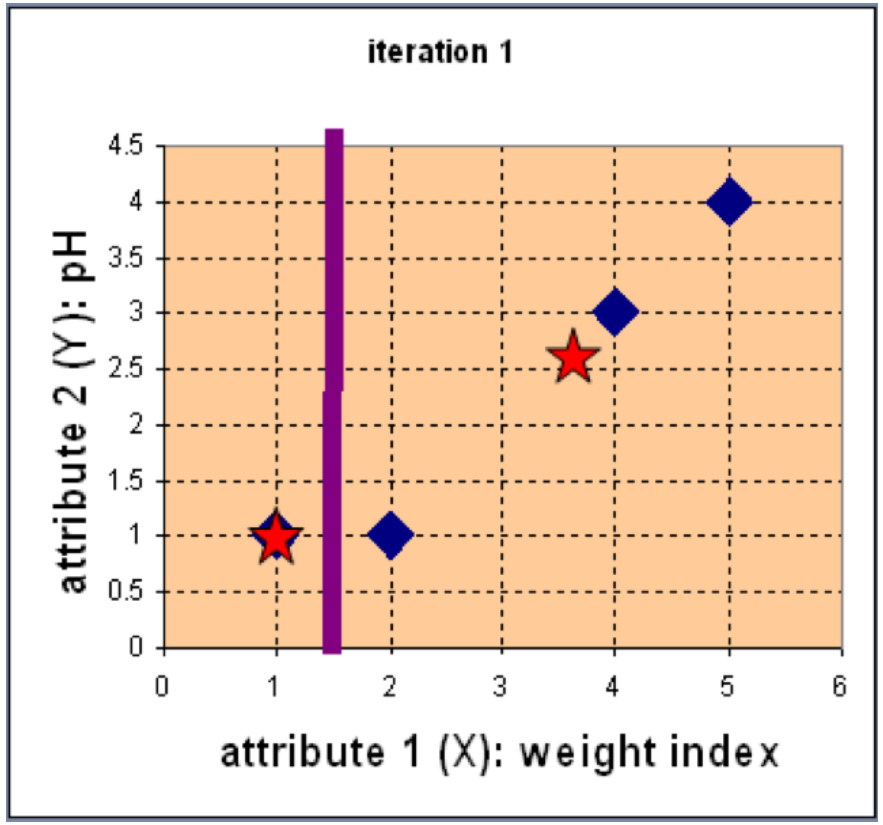
$$d(D,c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

$$d(D,c_2) = \sqrt{(5-2)^2 + (4-1)^2} = 4.24$$

Assign each object to the cluster with the nearest seed point

K-means - Example

- Step 2: Compute new centroids of the current partition



Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

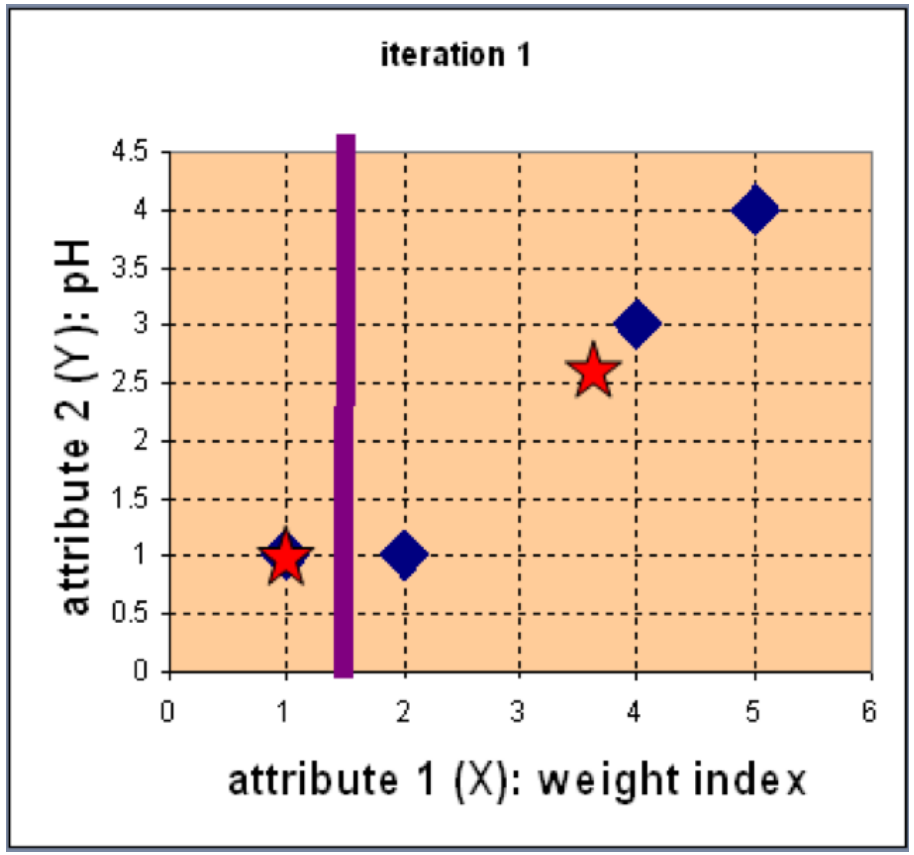
$$c_1 = (1,\ 1)$$

$$c_2 = \left(\frac{2+4+5}{3},\ \frac{1+3+4}{3}\right)$$

$$= \left(\frac{11}{3},\ \frac{8}{3}\right)$$

K-means - Example

- Step 2: Renew membership based on new centroids



Compute the distance of all objects to the new centroids

$$D^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} c_1 = (1,1) & group-1 \\ c_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

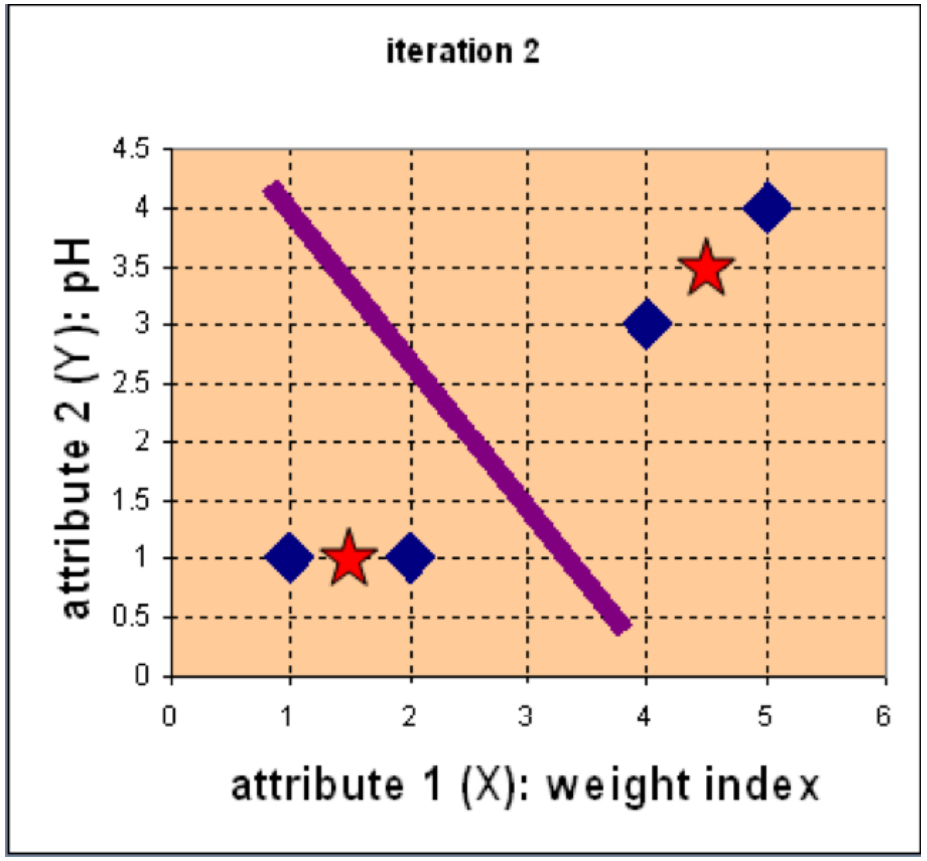
$$\begin{matrix} A & B & C & D \end{matrix}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{matrix} X \\ Y \end{matrix}$$

Assign the membership to objects

K-means - Example

- Step 3: Repeat the first two steps until its convergence



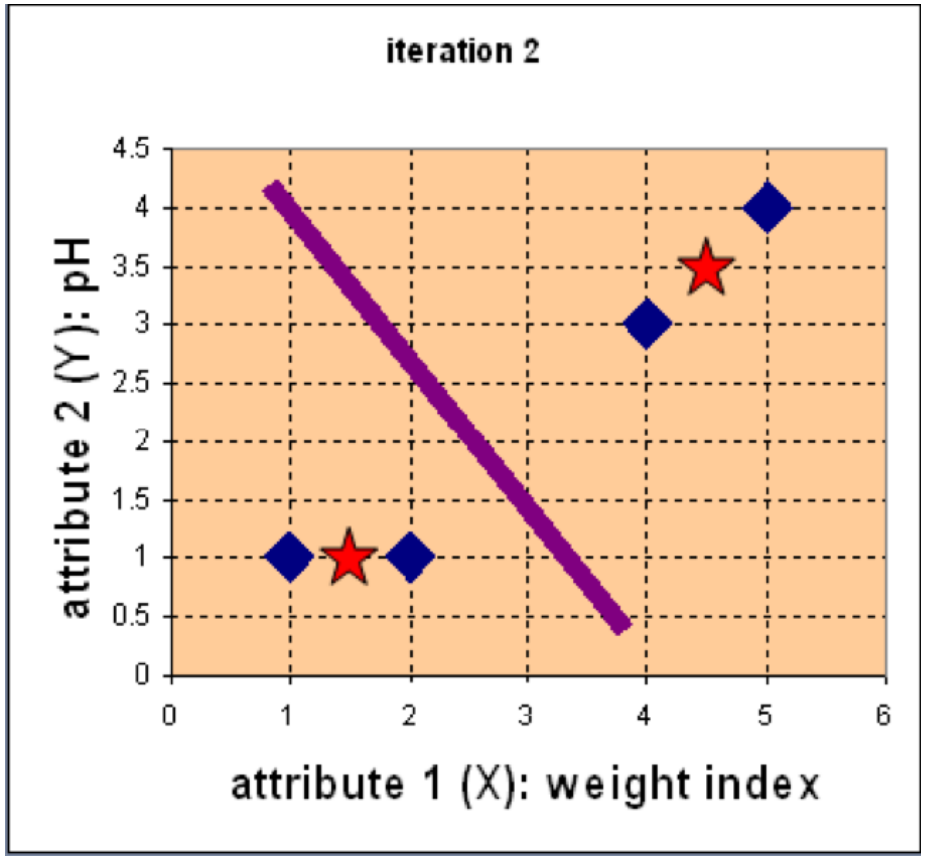
Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

$$c_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1\frac{1}{2}, 1)$$

$$c_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4\frac{1}{2}, 3\frac{1}{2})$$

K-means - Example

- Step 3: Repeat the first two steps until its convergence



Compute the distance of all objects to the new centroids

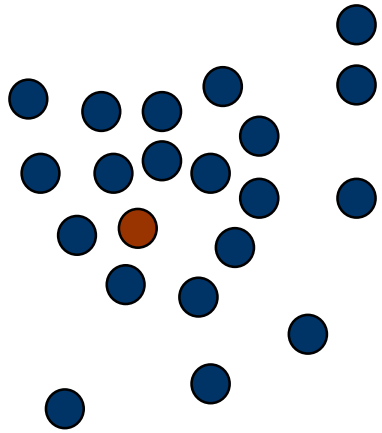
$$D^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{matrix} c_1 = (1\frac{1}{2}, 1) & group-1 \\ c_2 = (4\frac{1}{2}, 3\frac{1}{2}) & group-2 \end{matrix}$$

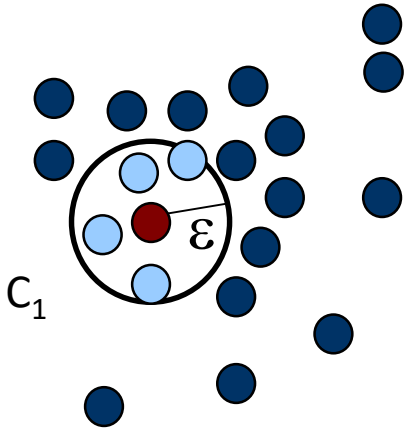
$$\begin{matrix} A & B & C & D \end{matrix}$$

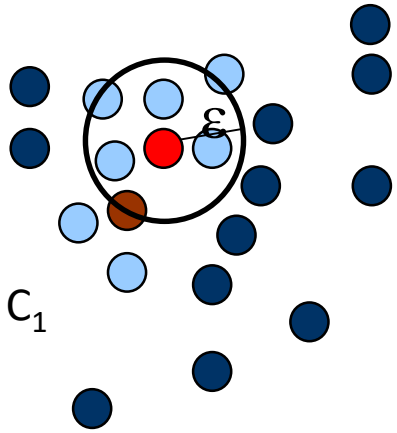
$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \quad \begin{matrix} X \\ Y \end{matrix}$$

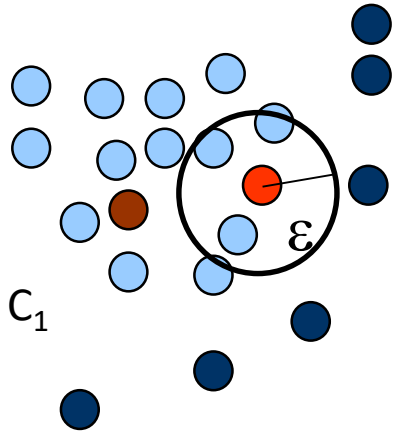
Stop due to no new assignment
Membership in each cluster no longer change

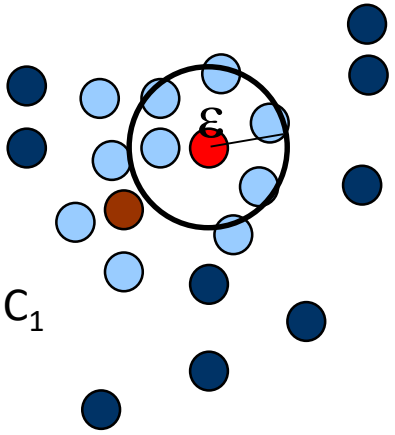
DBSCAN



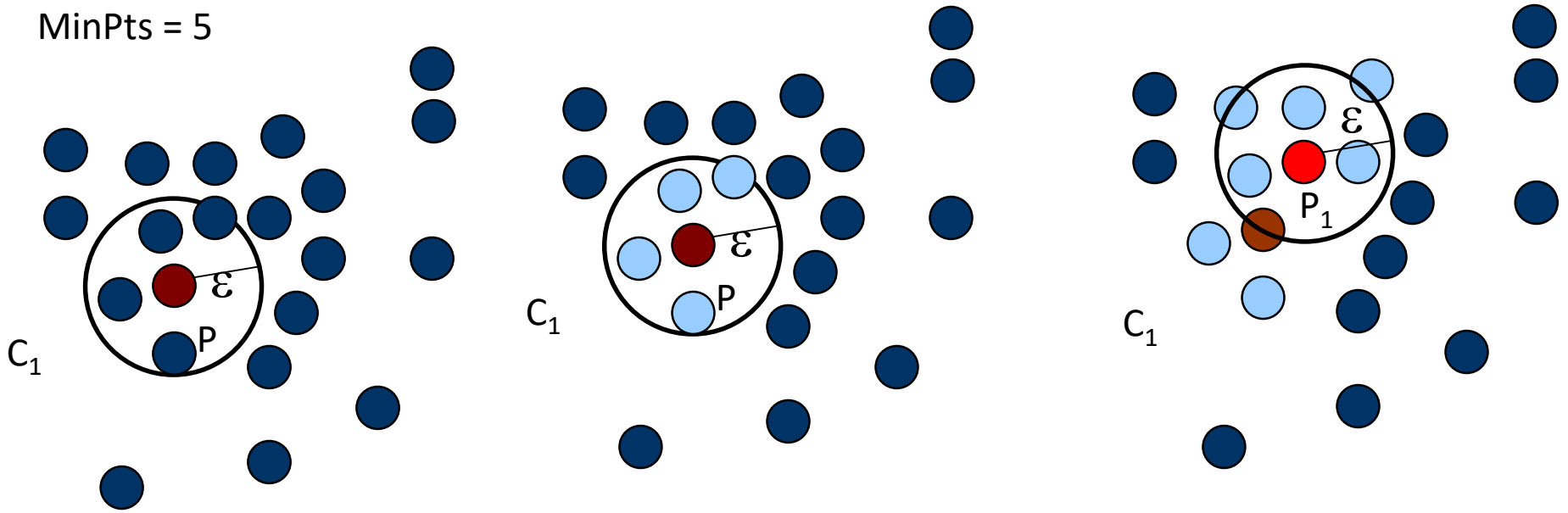






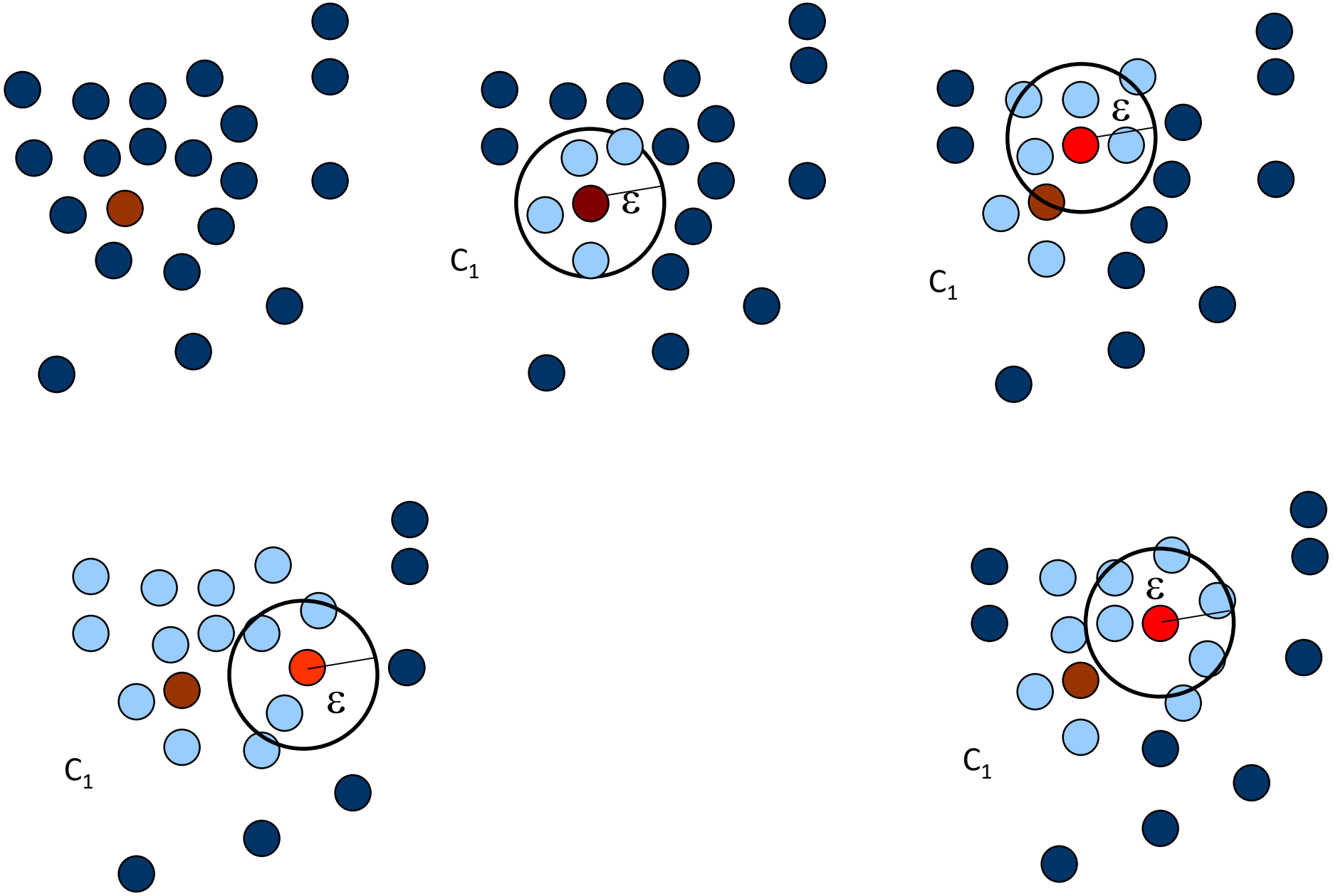


Clustering Techniques 46

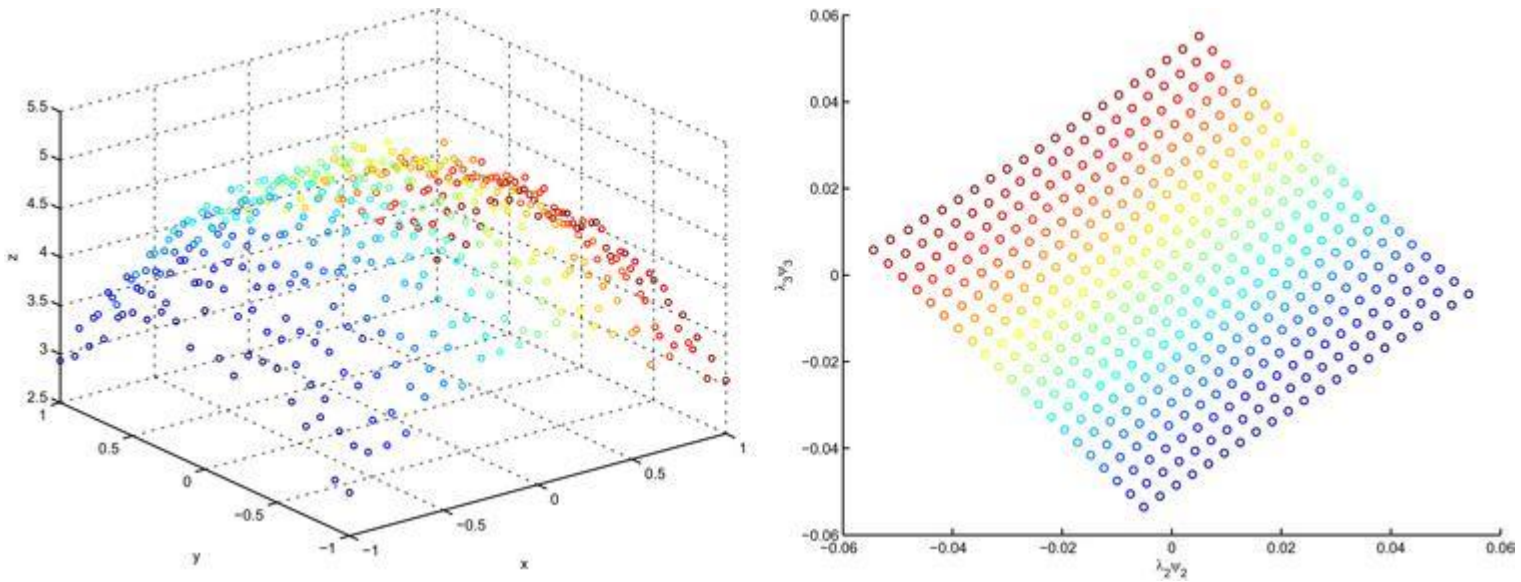


1. Check the ε-neighborhood of p;
2. If p has less than MinPts neighbors then mark p as outlier and continue with the next object
3. Otherwise mark p as processed and put all the neighbors in cluster C

1. Check the unprocessed objects in C
2. If no core object, return C
3. Otherwise, randomly pick up one core object p_1, mark p_1 as processed, and put all unprocessed neighbors of p_1 in cluster C

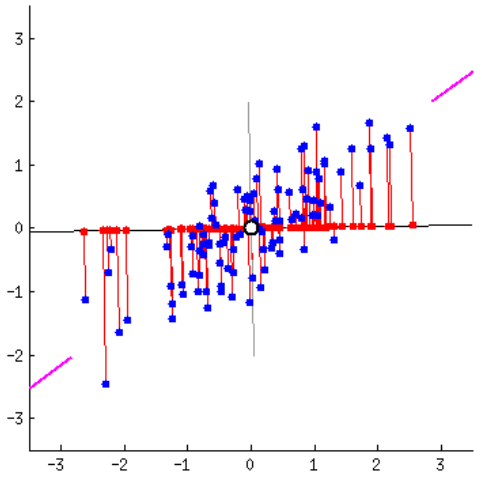

ε
C1
ε
C1
ε
C1
ε
C1

Dimensionality Reduction



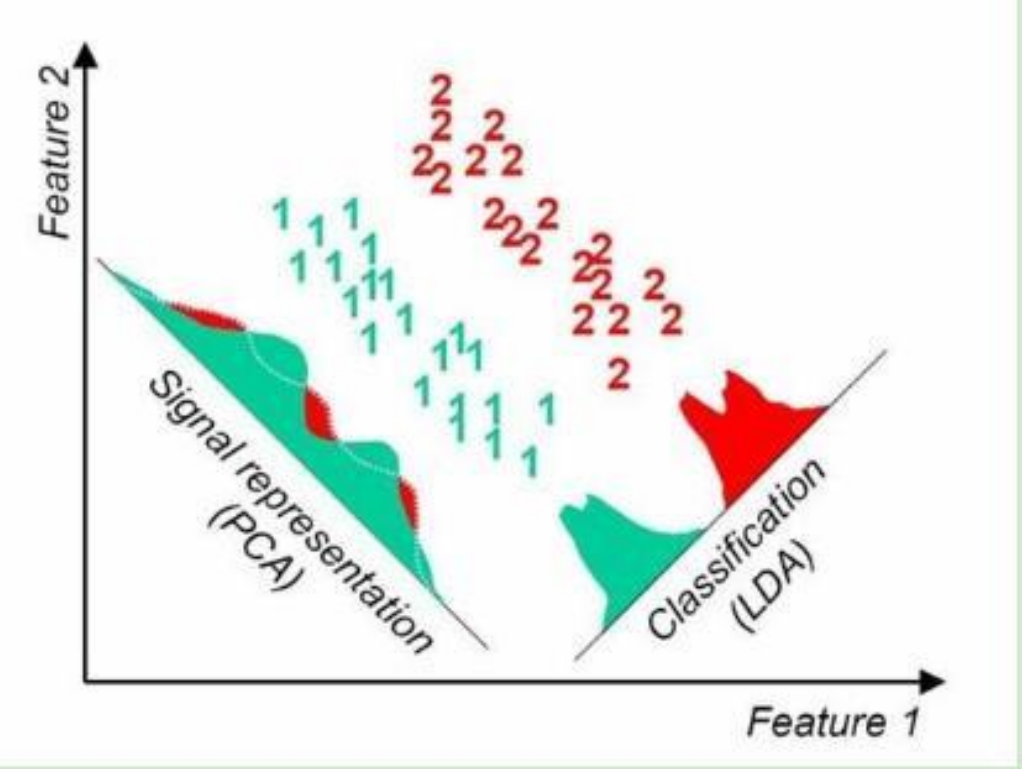
Principle Component Analysis

- Project data onto a space variance is maximized and Error is minimized.



Source: https://builtin.com/data-science/step-step-explanation-principal-component-analysis

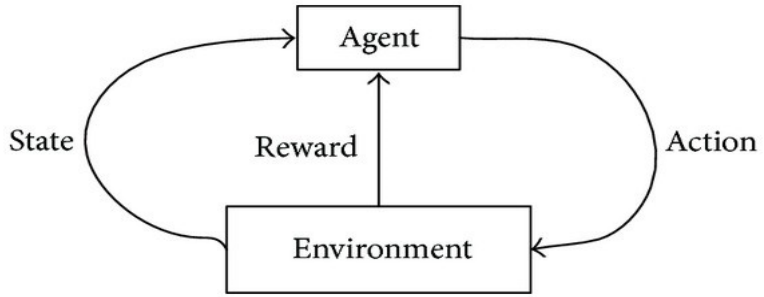
Linear Discriminant Analysis



Source: https://www.programmersought.com/article/33174132390/

Reinforcement Learning

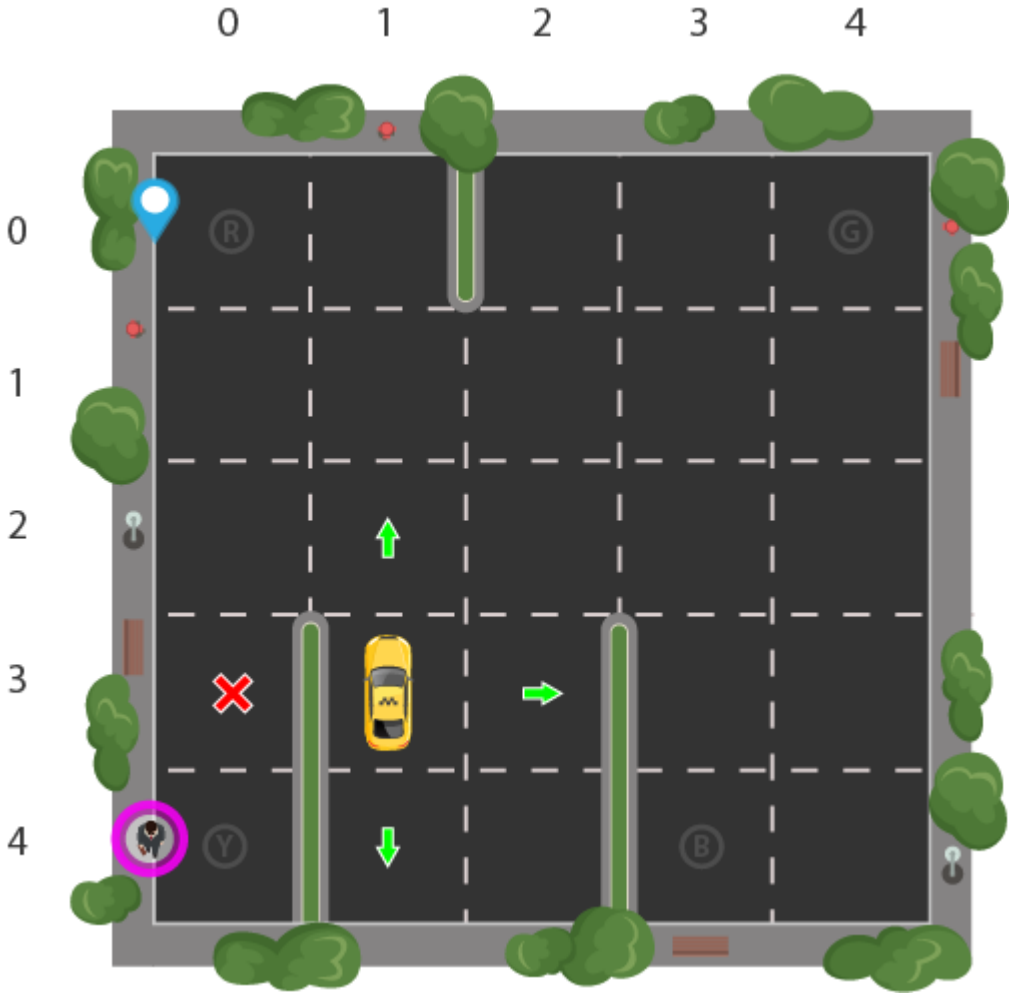
- Aims at using observations gathered from the interaction with the environment to take actions that would maximize the reward or minimize the risk.
- Reinforcement learning algorithm (called the agent) continuously learns from the environment in an iterative fashion.
- It allows machines and software agents to automatically determine the ideal behavior within a specific context, in order to maximize its performance.
- Simple reward feedback is required for the agent to learn its behavior; this is known as the reinforcement signal.
- In the problem, an agent is supposed to decide the best action to select based on his current state.



- Input state is observed by the agent.
- Decision making function is used to make the agent perform an action.
- After the action is performed, the agent receives reward or reinforcement from the environment.
- The state-action pair information about the reward is stored.
- List of Common Algorithms: Q-Learning, Temporal Difference (TD), Deep Adversarial Networks.
- Some applications of the reinforcement learning algorithms are computer played board games (Chess, Go), robotic hands, and self-driving cars.

Self Driving Cab

- six possible actions: south, north, east, west, pickup, dropoff.
- The **blue letter** represents the current passenger pick-up location, and the **purple letter** is the current destination.
- Reward:
  - -1 for each step
  - -10 for wrong pickup/drop off
  - 20 for drop off at the right location
- Possible number of states in the environment 500.



Useful links (Reinforcement Learning)

- https://www.learndatasci.com/tutorials/reinforcement-q-learning-scratch-python-openai-gym/

- https://medium.com/intro-to-artificial-intelligence/q-learning-a-value-based-reinforcement-learning-algorithm-272706d835cf

Reference Books