Machine Learning

Basics - 1

Thang Vu

6.11.2025

Questionnaire

- Please go to Ilias
- Open the questionnaire ’Probabilities & Optimization’
- Note that the questionnaire is anonymous, meaning we only receive the final statistics and responses, not the identities of the individuals who submitted them.

What is Machine Learning?

- Methods to enable computers to:
  - Learn from data
  - Improve themselves
- Without being explicitly programmed
- Similar to human thinking and learning

An Example: Sentiment Analysis

```
pos_dict = [‘good’, ‘great’, ‘positive’]
neg_dict = [‘bad’, ‘weak’, ‘terrible’]
tokens = tokenize(input)
pos_count = 0
neg_count = 0
for w in tokens:
        if w in pos_dict: pos_count++
        if w in neg_dict: neg_count++
if pos_count > neg_count:
        return positive
else return negative
```

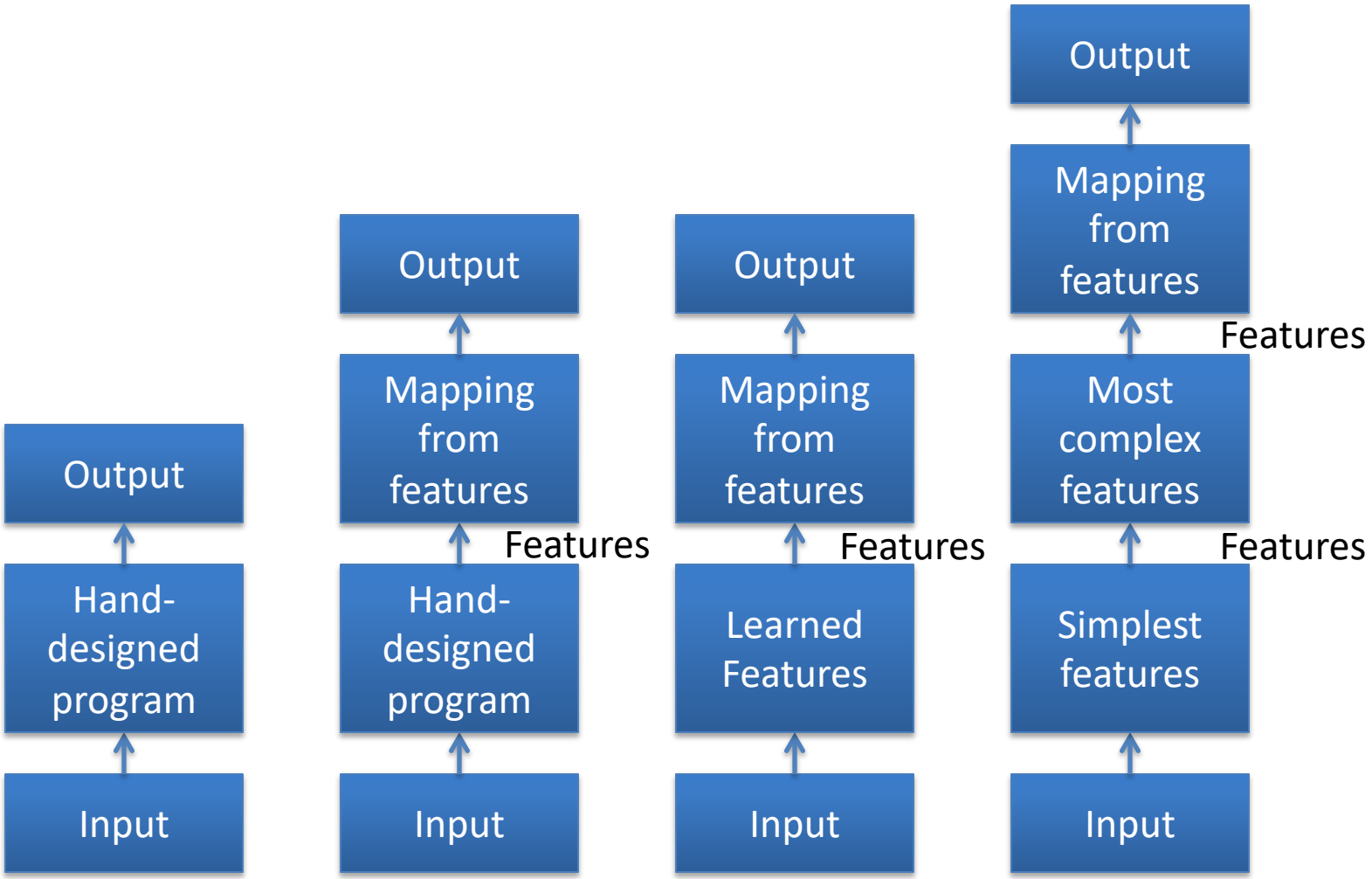
An Example: Sentiment Analysis

- This is an excellent movie: *positive*
- This movie is great: *positive*
- Wow, what a movie: *positive*
- I’m very positive about this movie.: *positive*
- Don’t waste time on it, it is horrible: *negative*
- Such a bad movie: *negative*
- I wasted two hours of my life: *negative*
- …..

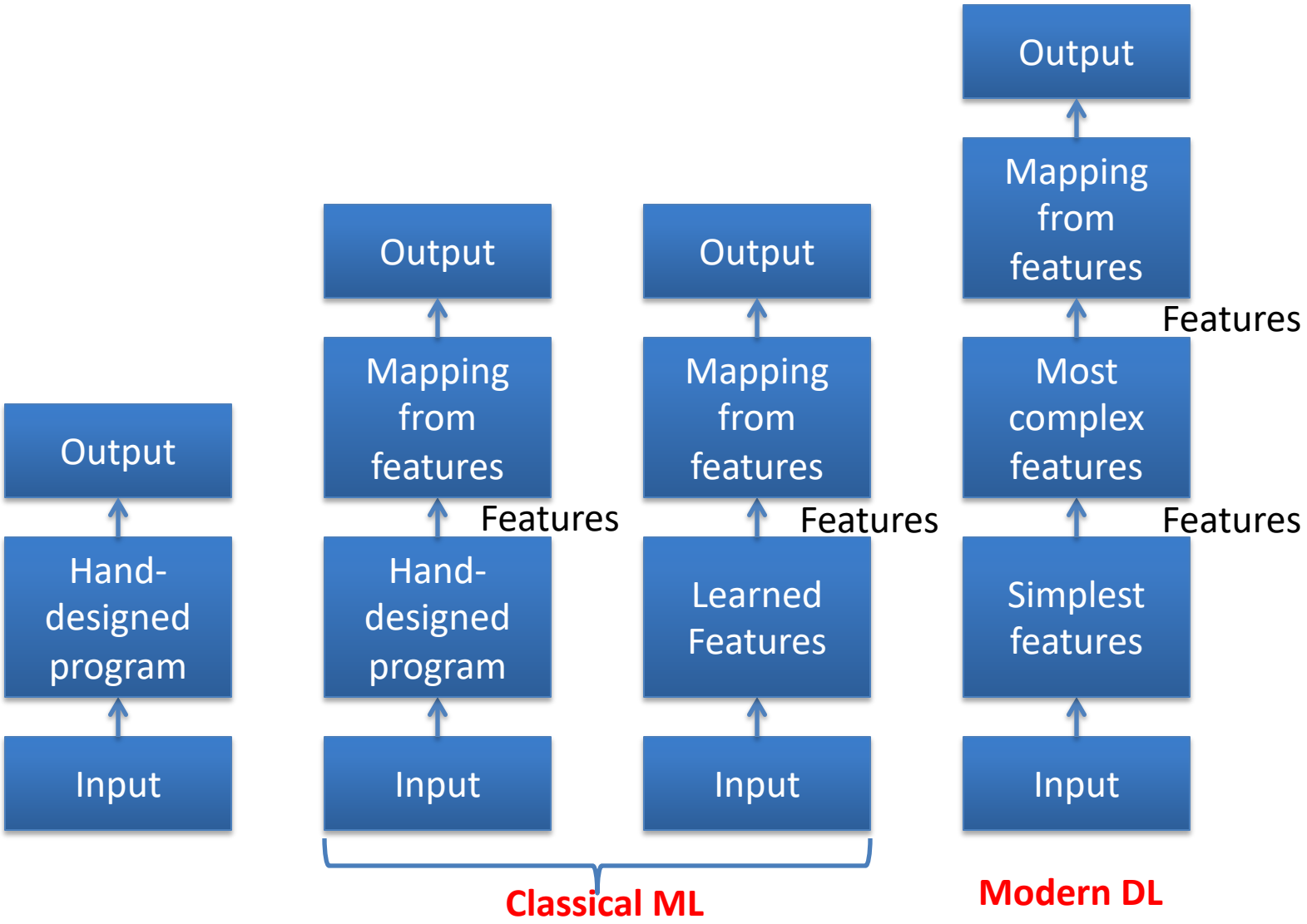
Data is the Key!

- Data is the crucial ingredient of machine learning
  - Collected from anywhere
  - Stored in many forms and formats
    - Structured data
    - Unstructured data
    - Usually messy
- In ML, data is a *list of examples* in which the machine learning methods can learn something from there
- **Understanding your data is an important step**

Machine Learning Pipeline



Machine Learning Pipeline



Features

- Transform a data point in a vector of numbers

Preprocessing
Feature extraction

(2, -3, 127, 0 …… , 2.4, 5) ‘car’

(1, -24, 12, 4, ….., 2.3, -5) ‘cyclo’

(10, -5, 23, -4.5, …., 2, -6) ‘motorbike’

Features for Speech and Language

- The same procedure can be applied to other tasks
- Speech recognition:

Features for Speech and Language

- The same procedure can be applied to other tasks
- Speech recognition:

- Natural language processing:

‘happy’ → (0, 0, 0, 0, 1, 0, 0, 0) 1-hot vector

(0.4, 0.3, 1, -0,2, …, 0,5) word embeddings

ML Models

(Non-)Parametric Models

- Machine learning models
  - Parametric (e.g. linear regression model) vs.
  - Non-parametric (e.g. k-nearest neighbors)

(Non-)Parametric Models

- Machine learning models
  - Parametric (e.g. linear regression model) vs.
  - Non-parametric (e.g. k-nearest neighbors)
- Parametric models have a fixed number of parameters which can be trained
  - Faster to use but they make strong assumptions about the nature of the data distributions

Example: Parametric Models

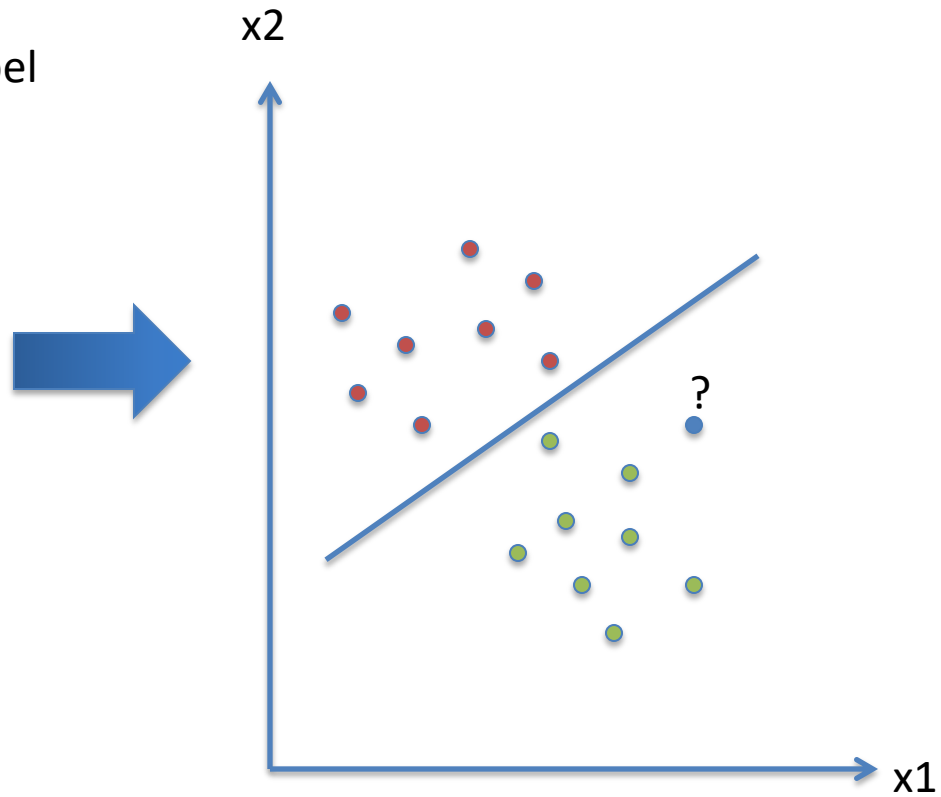
Target/class/label

y

Nr.	x1	x2	label
1	-0.2	0.8	1
2	0.4	-0.3	0
3	0.9	0.4	1
4	1	0	1

Data input

x



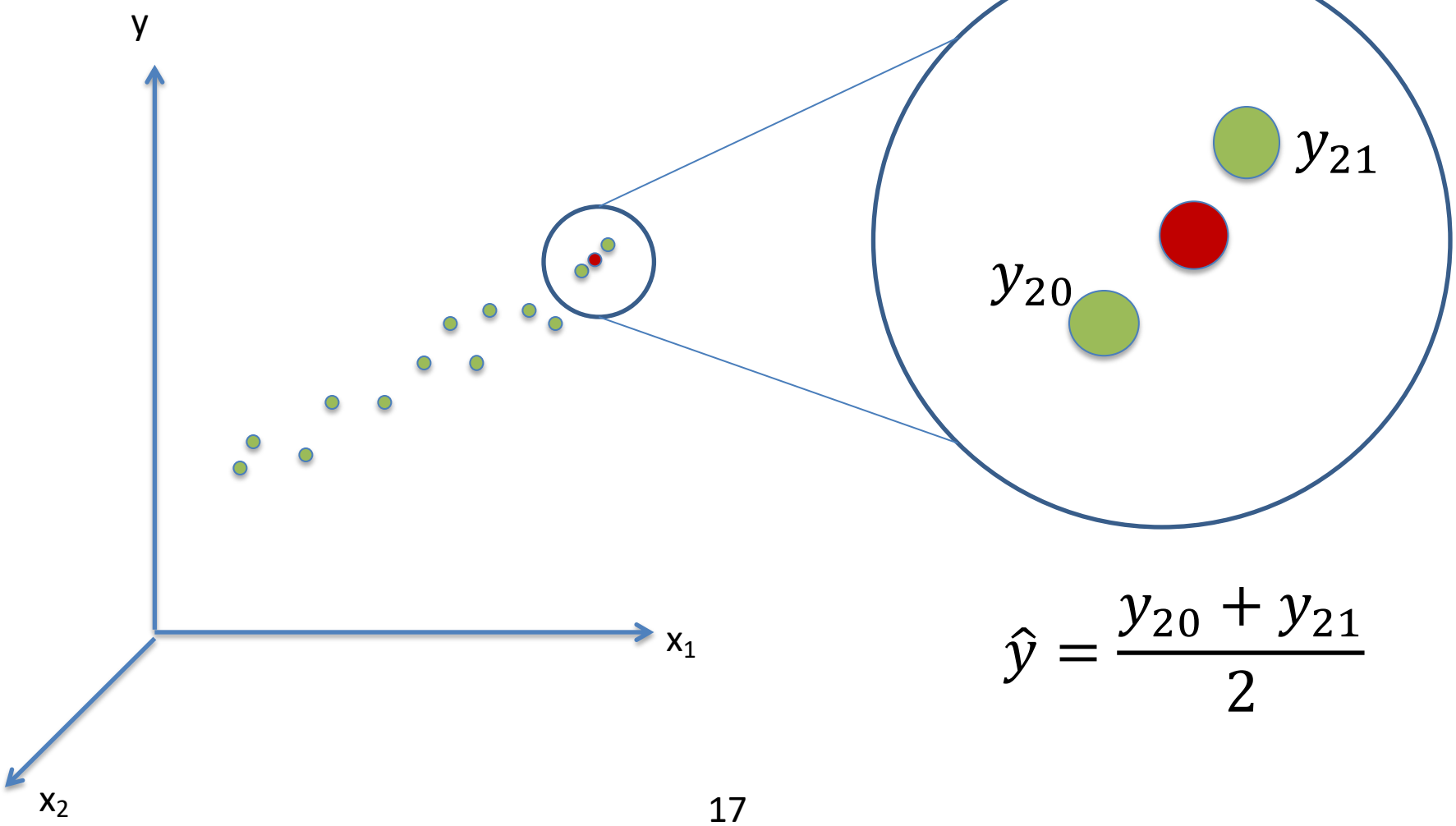
$$y = w_1 x_1 + w_2 x_2 + b$$

(Non-)Parametric Models

- Machine learning models
  - Parametric (e.g. linear regression model) vs.
  - Non-parametric (e.g. k-nearest neighbors)
- Parametric models have a fixed number of parameters which can be trained
  - Faster to use but they make strong assumptions about the nature of the data distributions
- Non-parametric models do not have parameters
  - More flexible but often computationally intractable for large datasets

Example: Non-parametric Models

- k = 2



$$\hat{y} = \frac{y_{20} + y_{21}}{2}$$

Hyperparameters

- Hyperparameters = settings that you can use to control the behavior of your machine learning models
  - Model hyperparameters, e.g.,
    - k in k-NNR
    - how large is your network?
  - Training hyperparameters, e.g.,
    - how fast do you want to update the parameters?
- Hyperparameters are *NOT the trainable parameters* that are estimated during training
- How to pick the best hyperparameters?
  - You have to search for them, but how?

Live Voting

Datasets

- At least three datasets:
  - Training set
    - The machine learning models are optimized on the training samples
  - Validation / Development set
    - To verify the generalizability of the model during training
    - To tune hyperparameters
  - Test / Evaluation set:
    - Unseen dataset
    - Test the generalizability of the model AFTER training
- Sometimes, we only have a training and a test set
  - In this case, a part of the training set can be held out and used as a development set

Generalization

- The central problem in ML is called *generalization*
  - Performance on new unseen inputs
- When training a model, we monitor
  - *Training error*
  - *Validation error*
- When testing a model on new unseen inputs, we report *generalization errors* or *test errors*
- Assumptions:
  - Examples in each dataset are independent of each other
  - The datasets are identically distributed

Generalization

- Transform the data in vector of numbers

Preprocessing
Feature extraction

(2, -3, 127, 0 …… , 2.4, 5) ‘car’

(1, -24, 12, 4, ….., 2.3, -5) ‘cyclo’

(10, -5, 23, -4.5, …., 2, -6) ‘motorbike’

New test data

??

(1, 2, 127, 0, …., -4, -5)

Generalization

- Transform the data in vector of numbers

Preprocessing
Feature extraction

(2, -3, 127, 0 …… , 2.4, 5) ‘car’

(1, -24, 12, 4, ….., 2.3, -5) ‘cyclo’

(10, -5, 23, -4.5, …., 2, -6) ‘motorbike’

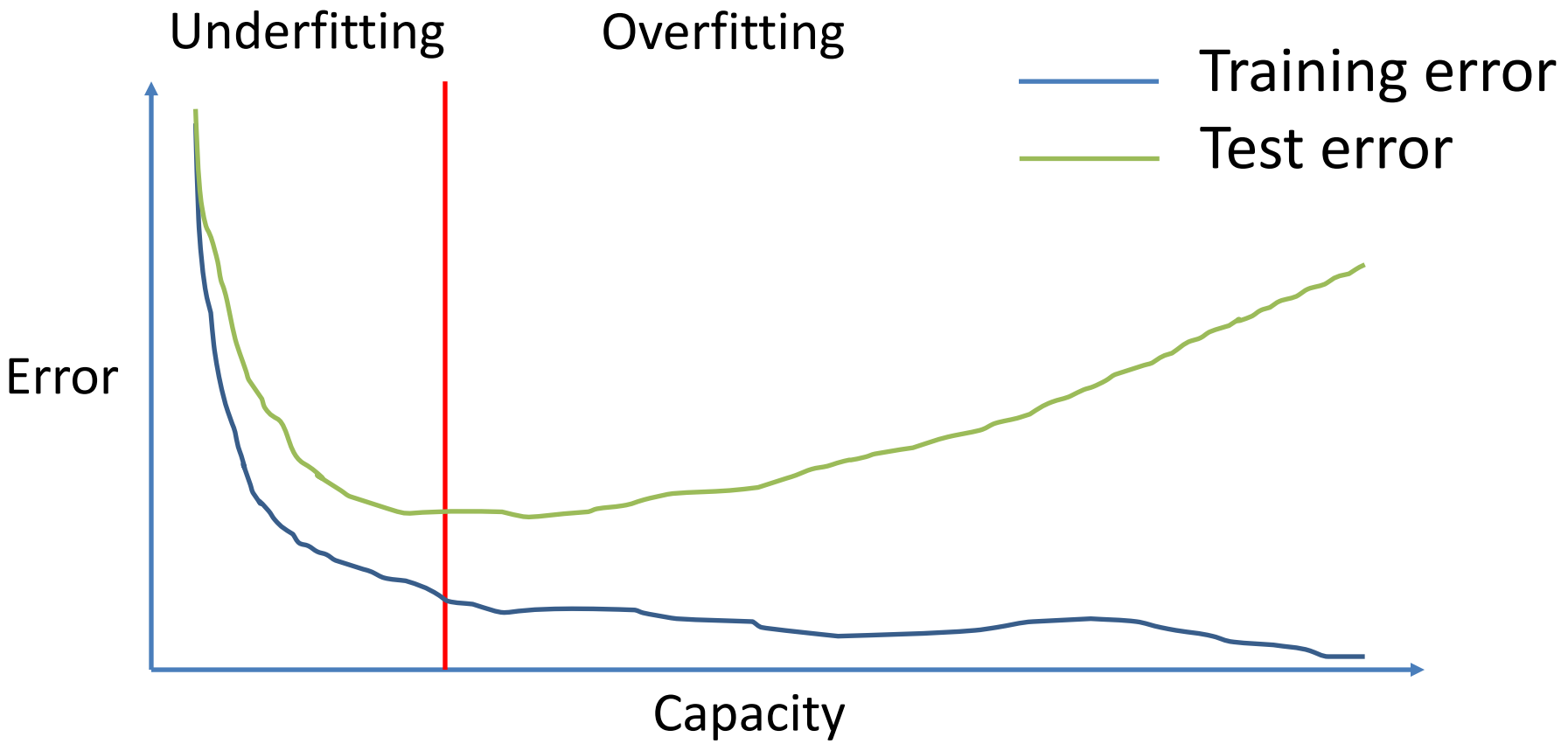
New test data

 ??

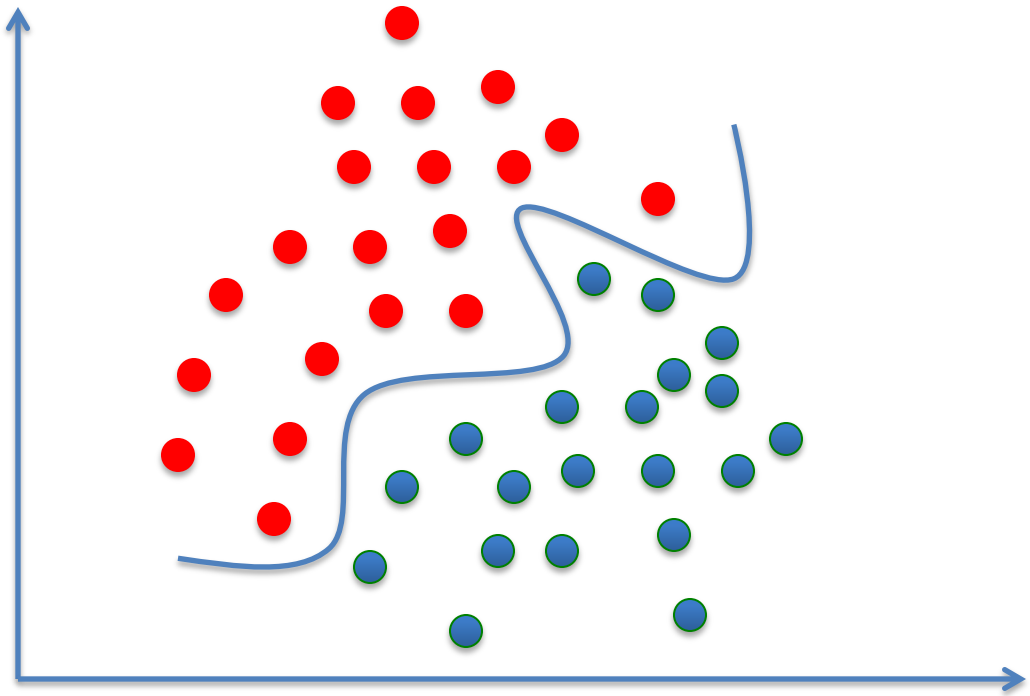
(1, 2, 127, 0, …., -4, -5)  ‘motorbike’

Capacity - Overfitting - Underfitting

- Capacity, overfitting and underfitting

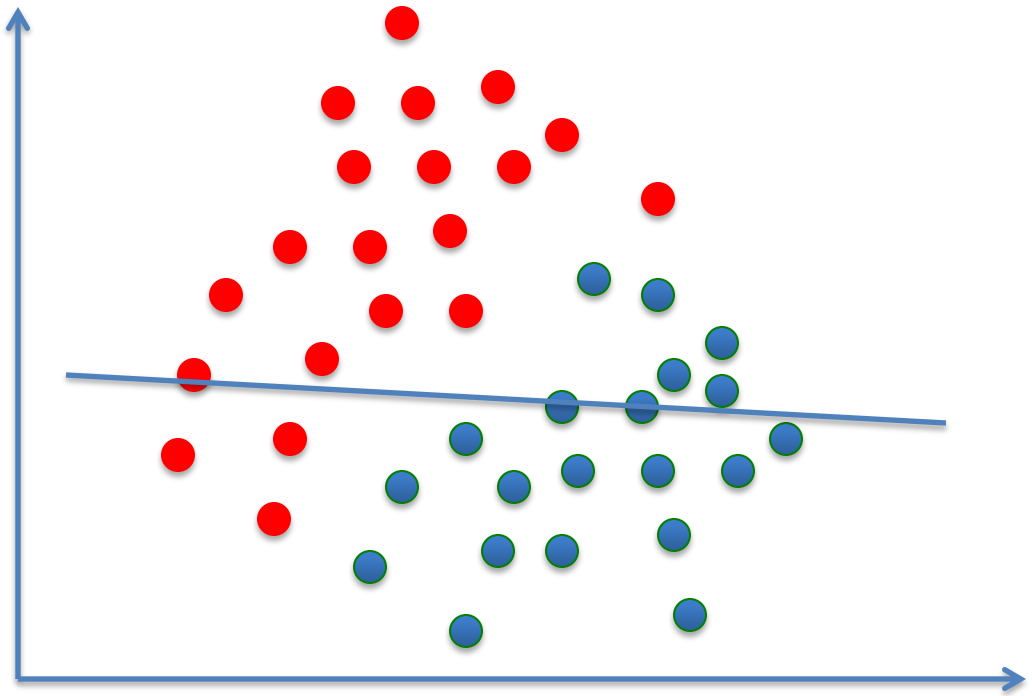


Overfitting



- Works perfectly on the training data
- But NOT on unseen testing data, i.e. the gap is too large

Underfitting



- Does not work well even on the training data

Live Voting

Machine Learning Paradigms

- Supervised learning: predict target y from input x
  - Classification: y represents a category or class
  - Regression: y is a real-value number
  - Usually, the training data x is given with its corresponding label y
- Unsupervised learning: no explicit prediction target y
  - Density estimation: model the probability distribution of x
  - Clustering: discover the underlying structure of data
- Other types, e.g., reinforcement learning

Unsupervised learning models

Overview

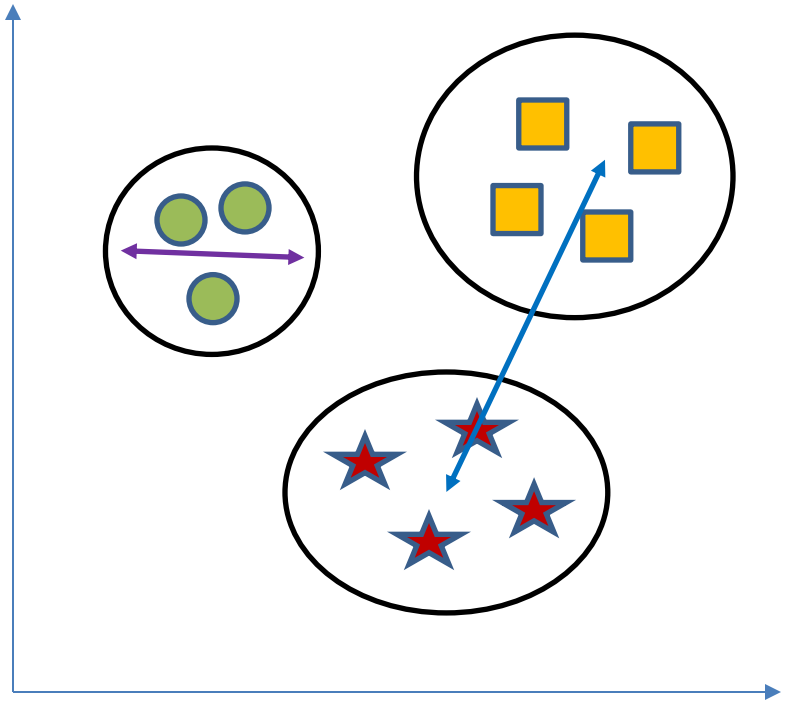
- Questionnaire
- Unsupervised Methods
- Supervised Methods

Unsupervised Learning

- No explicit prediction target y
- Goal: Discover underlying structure in data
  - Clustering

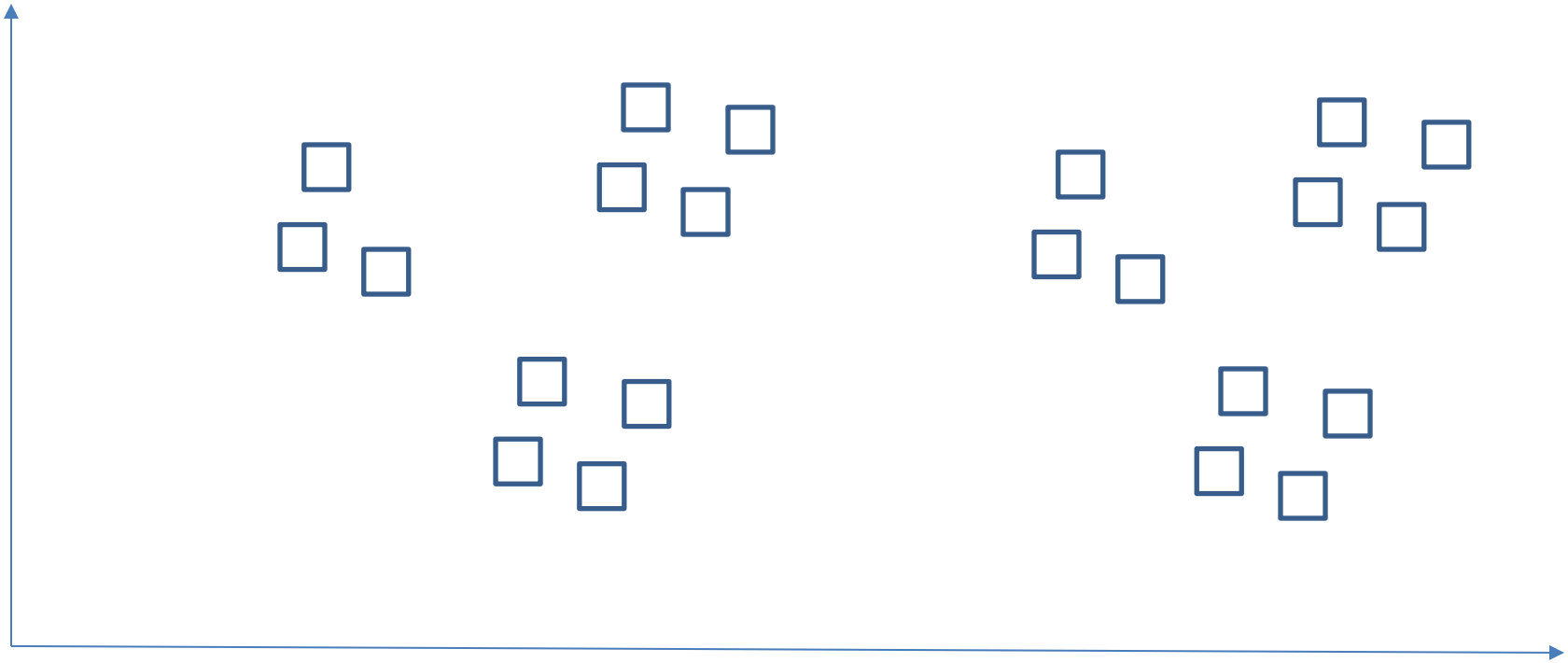
Clustering

- Aggegration of objects into homogeneous groups (clusters or classes)
- The objectives are
  - High homogenity within a cluster/class
  - High heterogenity between clusters/classes



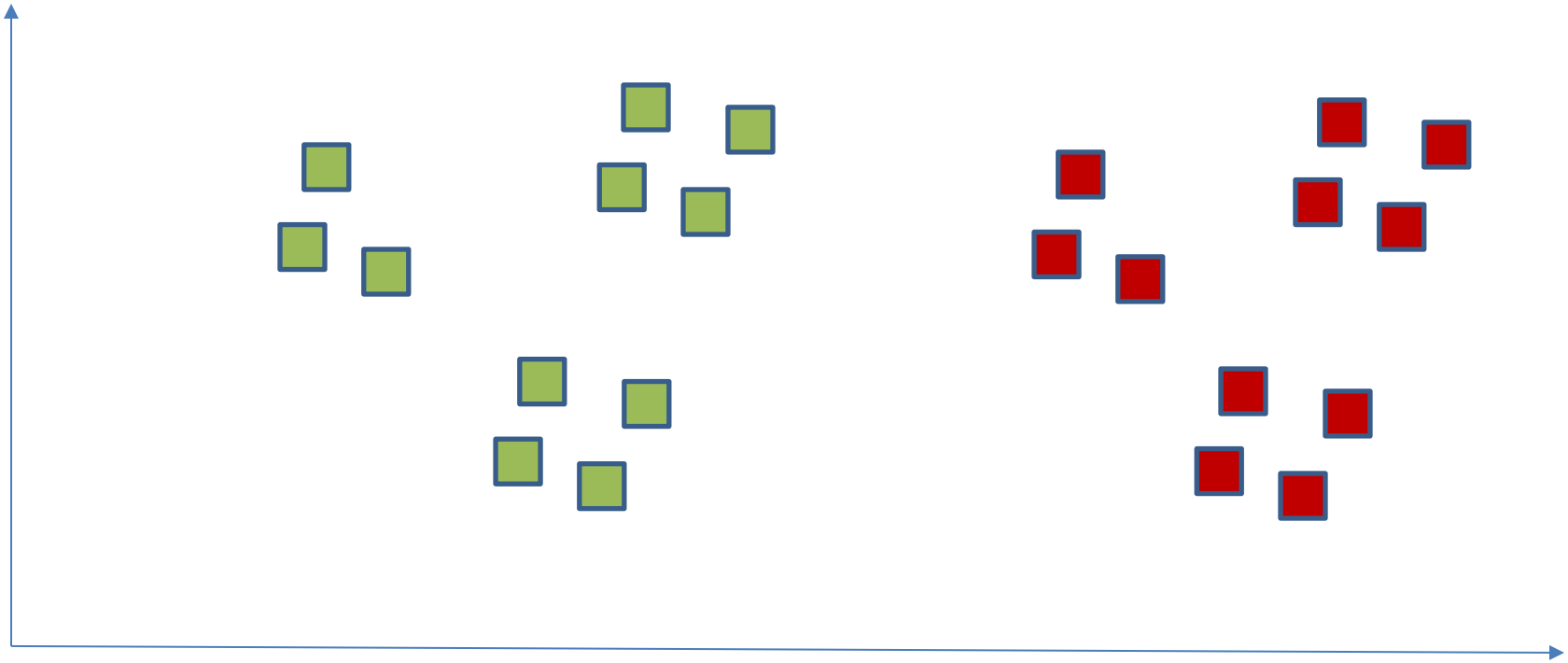
Ambiguity in Clustering

- How many clusters?



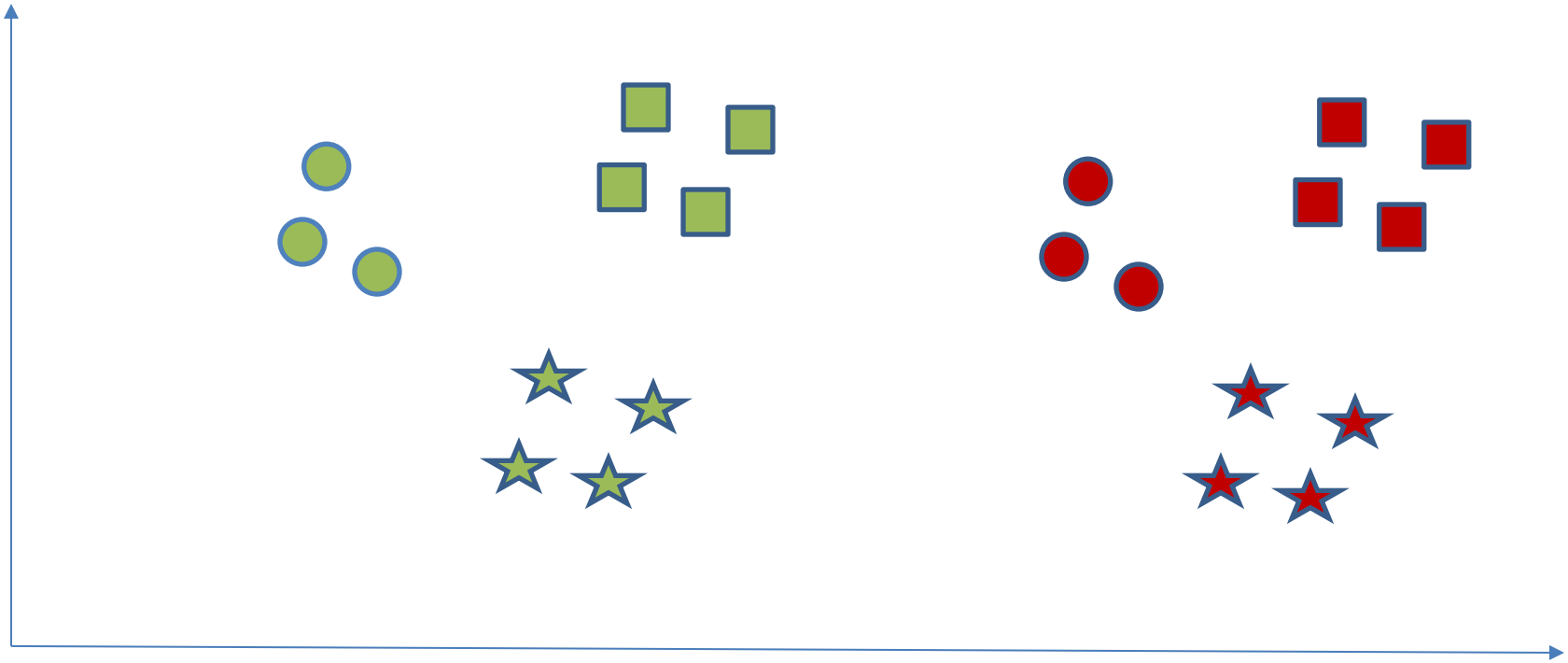
Ambiguity in Clustering

- How many clusters?



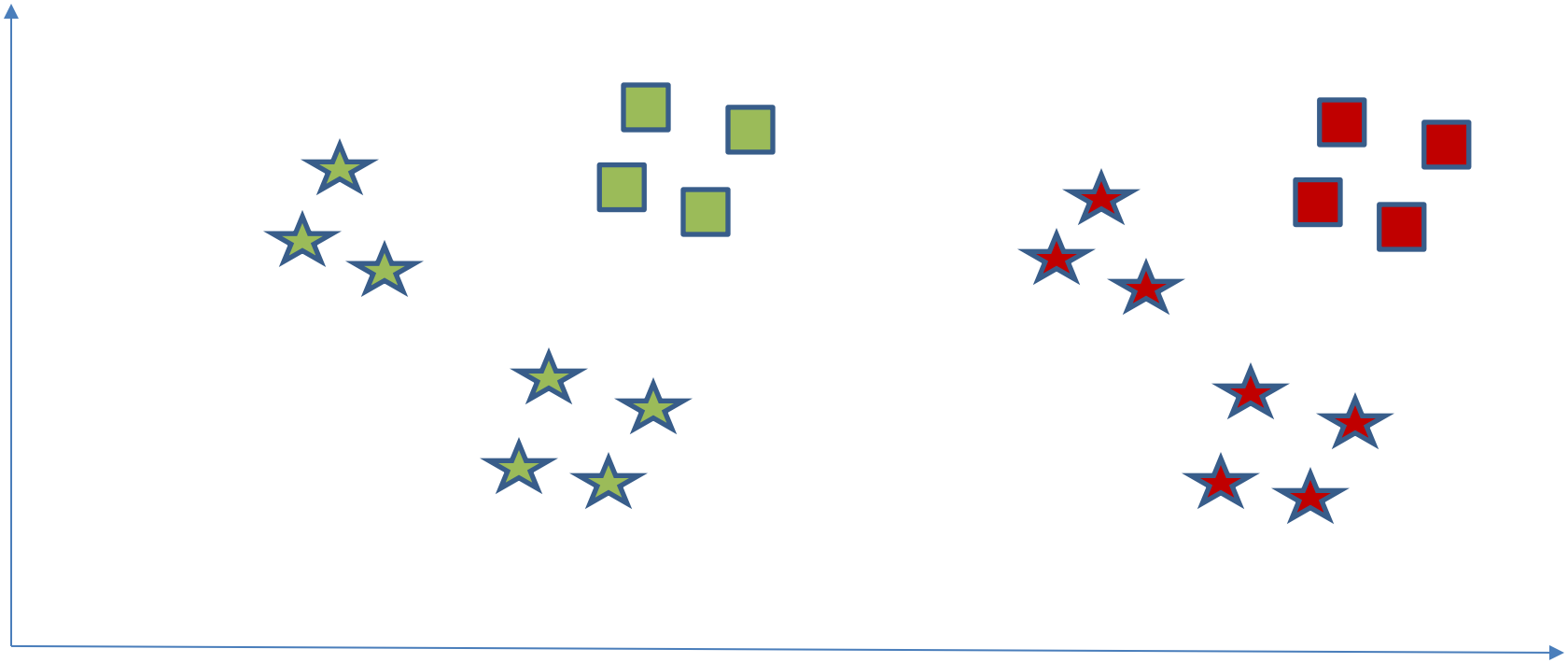
Ambiguity in Clustering

- How many clusters?



Ambiguity in Clustering

- How many clusters?



Similarity and Distance

- Similarity describes the degree to which two objects are alike
  - e.g. cosine similarity

$$\cos(x, y) = \frac{x \cdot y}{||x|| \, ||y||}$$

- Distance is an inverse of similarity ('dissimilarity')
  - e.g. Euclidean distance

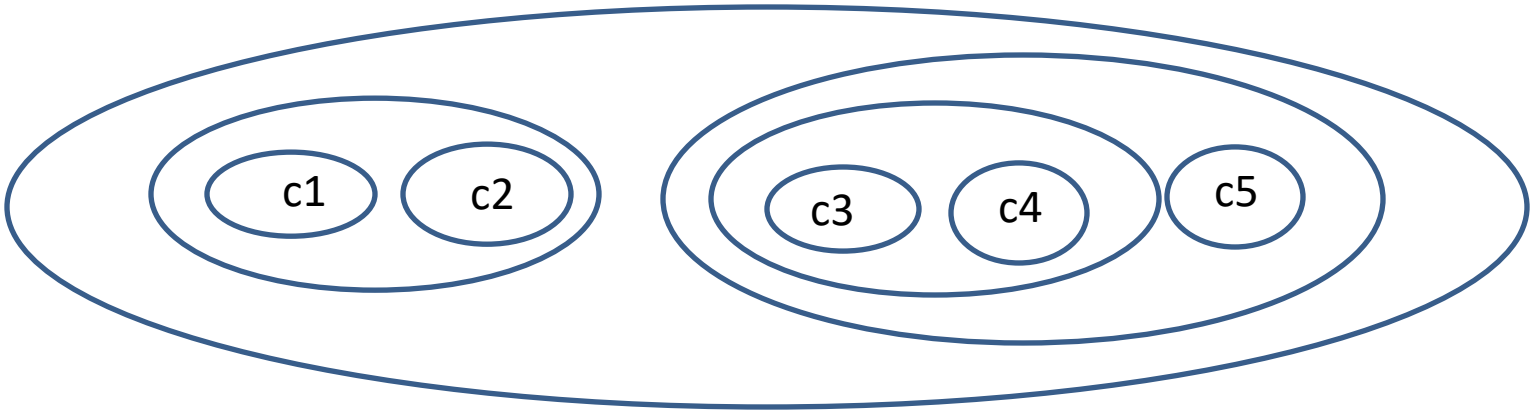
$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Types of Clustering

- Partitional Clustering
  - Partitioning of data objects into non-overlapping subregions
  - Each object is exactly assigned to one subregion

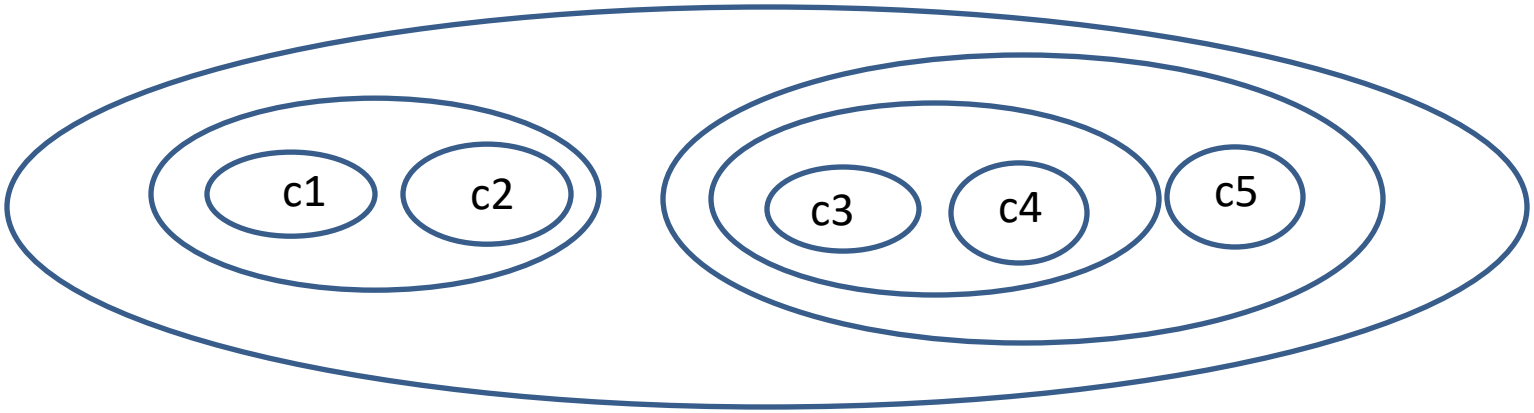
Types of Clustering

- Partitional Clustering
  - Partitioning of data objects into non-overlapping subregions
  - Each object is exactly assigned to one subregion
- Hierarchical Clustering
  - Nested clusters which form a tree



Types of Clustering

- Partitional Clustering
  - Partitioning of data objects into non-overlapping subregions
  - Each object is exactly assigned to one subregion
- Hierarchical Clustering
  - Nested clusters which form a tree



Partitional Clustering: K-means

Step 1	**Initialize:** Given a value of k and sample vectors v_1, ... v_T, initialize any k means (e.g. $\mu_i = v_i$)
Step 2	**Nearest-Neighbor classification**: Assign every vector v_i to its closest centroid $\mu_{f(i)}$
Step 3	**Parameters update**: Replace every μ_i by the mean of all sample vectors that have been assigned to it (centroid of cluster)
Step 4	**Iteration**: If not satisfied yet, go to **Step 2**

- Possible stop-criteria:
  - A fixed number of iterations
  - The average (maximum) distance $|v_i - \mu_{f(i)}|$ is below a fixed value
  - The derivative of the distance is below a fixed value (nothing happens)

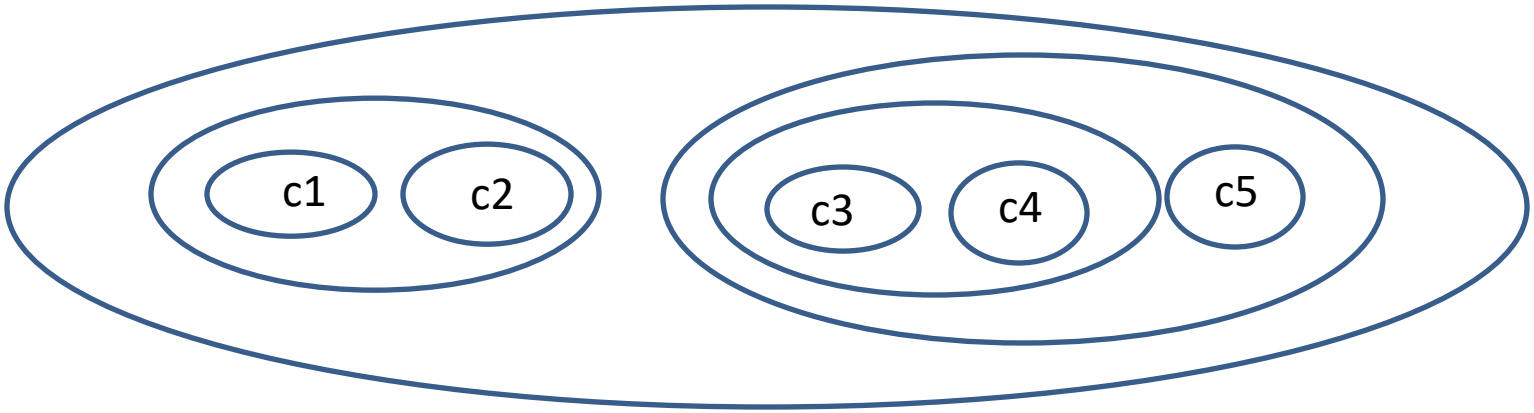
Visualizing K-means

- https://clustering-visualizer.web.app/kmeans

Live Voting

Types of Clustering

- Partitional Clustering
  - Partitioning of data objects into non-overlapping subregions
  - Each object is exactly assigned to one subregion
- Hierarchical Clustering
  - Nested clusters which form a tree



Hierarchical Clustering

- Agglomerative – bottom-up approach
  - Each object starts in its own cluster
  - Pairs of clusters are merged to create the hierachy
- Divisive – top-down approach
  - All objects start in one cluster
  - Splits are performed recursively to grow the tree

Evaluation of Clustering Methods

- Primary objectives in clustering:
  - High intra-cluster similarity
  - Low inter-cluster similarity

⇒ *Internal criteria* for the quality of a clustering

- Note that good scores on internal criteria do not imply effectiveness in applications

⇒ Direct evaluation in downstream applications

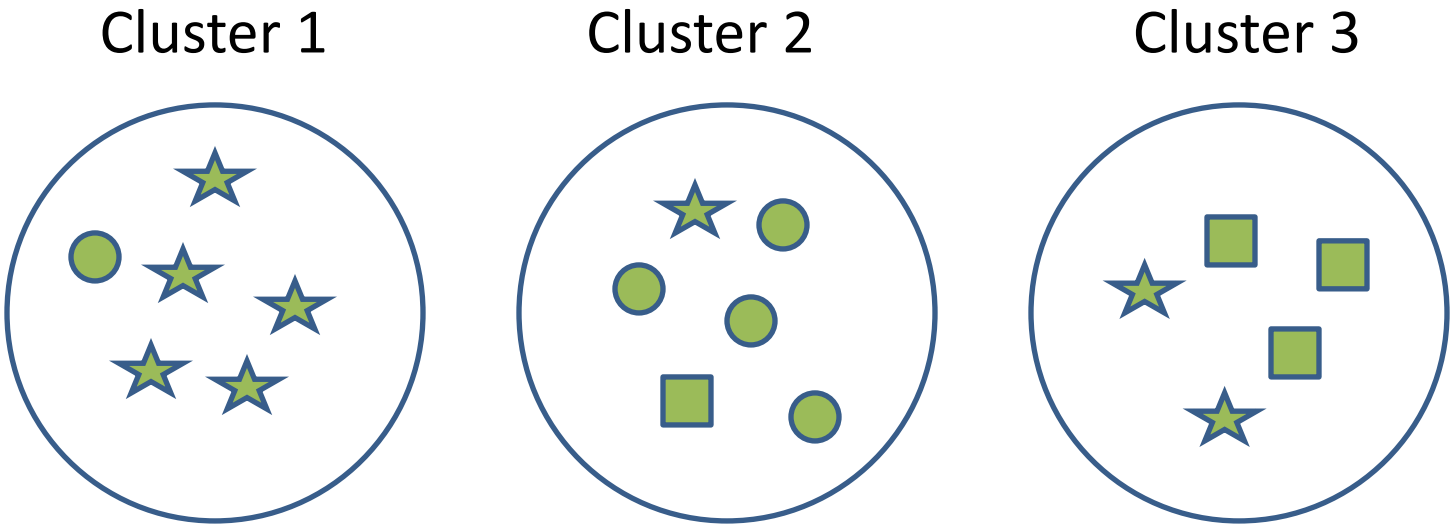
- If a gold standard (i.e., labels y) exists, *external criteria* can be used

An Internal Criteria: Davies-Bouldin Index

$$DB = \frac{1}{n} \sum_{i=1}^{n} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$$

- n is the number of clusters
- c_x is the centroid of cluster x
- σ_x is the average distance of all elements in cluster x to centroid c_x
- $d(c_x; c_y)$ is the distance between centroid c_x and c_y

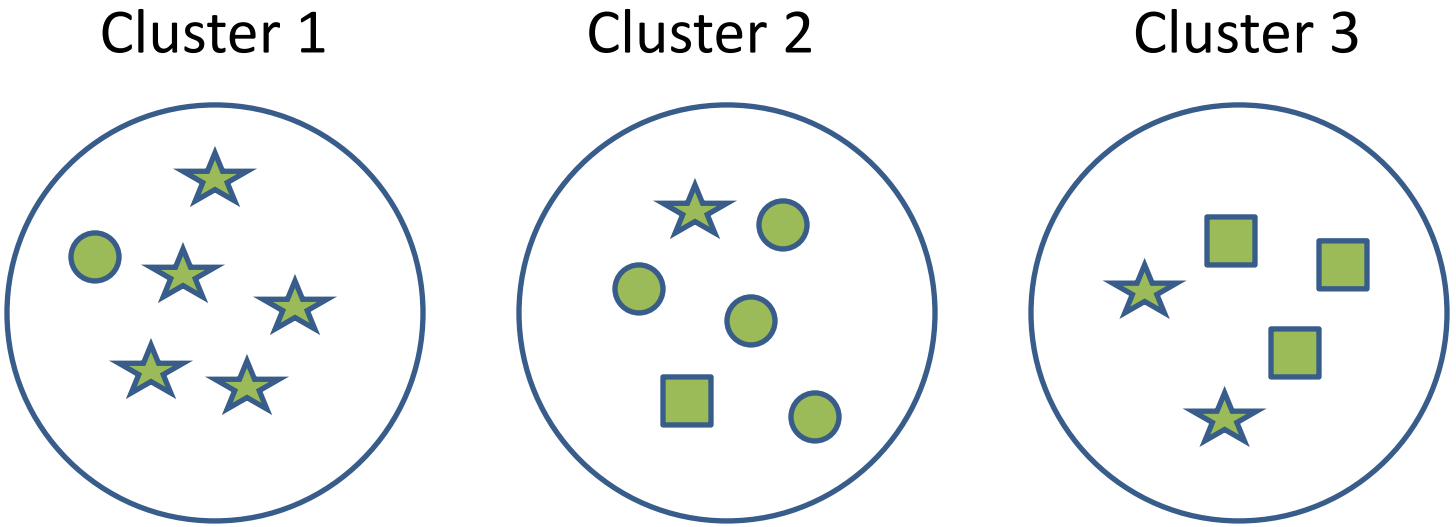
An External Criteria: Purity



- ★ appears 5 times in cluster 1
- ● appears 4 times in cluster 2
- ■ appears 3 times in cluster 3

$$purity = \frac{1}{17}(5 + 4 + 3) \approx 0.71$$

An External Criteria: Purity



$$purity(C, L) = \frac{1}{N}\sum_{k} \max_{j} |c_k \cap l_j|$$

- $C = \{clusters\}$ and $L = \{$ ■ ● ★ $\}$

Supervised learning models

Supervised Learning

- Supervised learning: predict target y from input x
  - *Training data x is given with its corresponding label, y*
  - Regression: y is a real-valued number
  - Classification: y represents a category or class

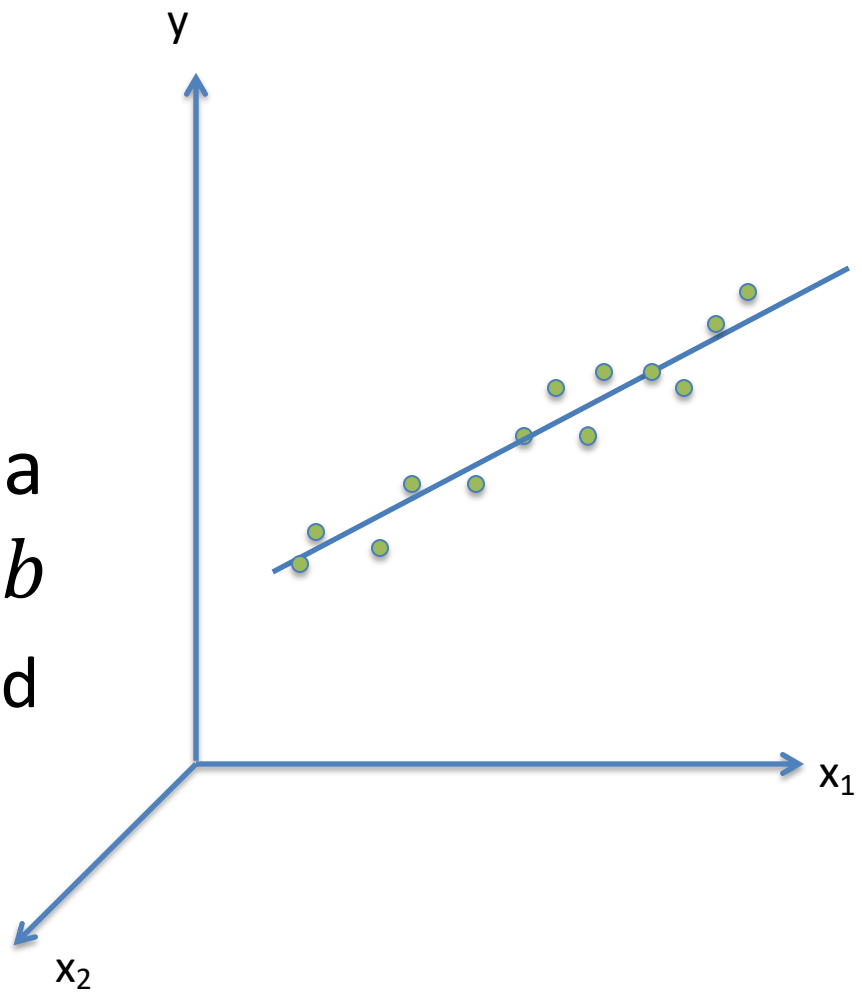
Supervised Learning

- Supervised learning: predict target y from input x
  - *Training data x is given with its corresponding label, y*
  - Regression: y is a real-valued number
  - Classification: y represents a category or class

Linear Regression

- LR solves a regression problem
  - Input: a vector $x \in R^n$
  - Output: a scalar $y \in R$
- LR model is specified with a vector w $\in R^n$ and a bias b
  - The predicted y is computed as follows:

$$\hat{y} = w^T x + b$$



Example: Housing Price Prediction

- In the Boston data set collected in 1978, an object can be characterized by:
  - CRIM - per capita crime rate by town
  - ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
  - INDUS - proportion of non-retail business acres per town
  - CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
  - NOX - nitric oxides concentration (parts per 10 million)
  - RM - average number of rooms per dwelling
  - AGE - proportion of owner-occupied units built prior to 1940
  - DIS - weighted distances to five Boston employment centres
  - RAD - index of accessibility to radial highways
  - TAX - full-value property-tax rate per $10,000
  - PTRATIO - pupil-teacher ratio by town
  - B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
  - LSTAT - % lower status of the population
  - PRICE - Median value of owner-occupied homes in $1000's

Housing Price Prediction: Features

- In the Boston data set collected in 1978, an object can be characterized by:
  - CRIM - per capita crime rate by town
  - ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
  - INDUS - proportion of non-retail business acres per town
  - CHAS - Charles River dummy variable (1 if tract bounds river; 0 otherwise)
  - NOX - nitric oxides concentration (parts per 10 million)
  - RM - average number of rooms per dwelling — x
  - AGE - proportion of owner-occupied units built prior to 1940
  - DIS - weighted distances to five Boston employment centres
  - RAD - index of accessibility to radial highways
  - TAX - full-value property-tax rate per $10,000
  - PTRATIO - pupil-teacher ratio by town
  - B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
  - LSTAT - % lower status of the population
  - PRICE - Median value of owner-occupied homes in $1000's — y

Housing Price Prediction: Features

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

https://www.kaggle.com/code/shreayan98c/boston-house-price-prediction

Linear Regression for House Price Prediction

- Each object is a point or a vector in a vector space with 13 dimensions $x = [x_0; x_1; \ldots; x_{12}]$ and is labeled with a price y
- The predicted price $\hat{y}$ is computed as follows:

$$\hat{y} = w_0 x_0 + w_1 x_1 + \ \ldots + w_{12} x_{12} + \ b$$

- We need to estimate $[w_0; w_1; \ldots; w_{12}]$ and b so that the error $\sum_i (\hat{y} - y)_i^2$ is minimal for a *training set*

Linear Regression for House Price Prediction

- Each object is a point or a vector in a vector space with 13 dimensions $x = [x_0; x_1; \ldots; x_{12}]$ and is labeled with a price y
- The predicted price $\hat{y}$ is computed as follows:

$$\hat{y} = w_0 x_0 + w_1 x_1 + \ \ldots + w_{12} x_{12} + \ b$$

- We need to estimate $[w_0; w_1; \ldots; w_{12}]$ and b so that the error $\sum_i (\hat{y} - y)_i^2$ is minimal for a *training set*

parameters

loss function

Linear Regression for House Price Prediction

	Attribute	Coefficients
0	CRIM	-0.12257
1	ZN	0.0556777
2	INDUS	-0.00883428
3	CHAS	4.69345
4	NOX	-14.4358
5	RM	3.28008
6	AGE	-0.00344778
7	DIS	-1.55214
8	RAD	0.32625
9	TAX	-0.0140666
10	PTRATIO	-0.803275
11	B	0.00935369
12	LSTAT	-0.523478

https://www.kaggle.com/code/shreayan98c/boston-house-price-prediction

Evaluation of Regression

- Mean squared errors:

$$MSE_{test} = \frac{1}{m}\sum_i (\hat{y}^{(test)} - y^{(test)})_i^2$$

- Root mean squared errors

$$RMSE_{test} = \sqrt{MSE_{test}}$$

Supervised Learning

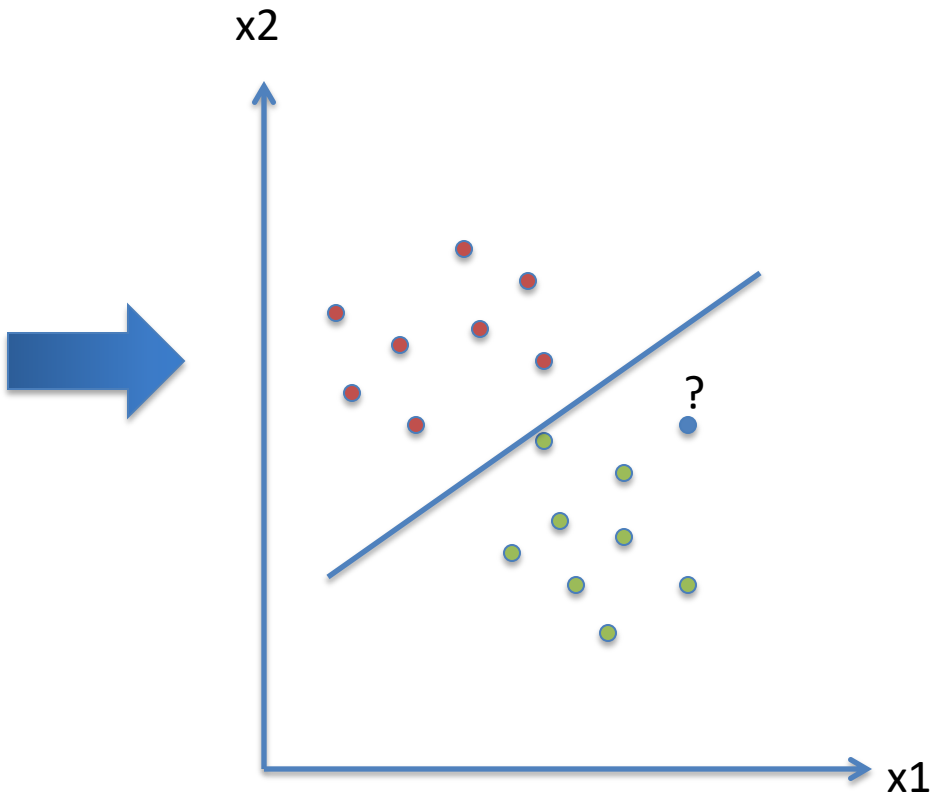
- Supervised learning: predict target y from input x
  - *Training data x is given with its corresponding label, y*
  - Regression: y is a real-valued number
  - Classification: y represents a category or class

Classification

Target/class/label
y

Nr.	x1	x2	label
1	-0.2	0.8	1
2	0.4	-0.3	0
3	0.9	0.4	1
4	1	0	1

Data input
x



Classification Problems

- Binary classification
  - Only two classes
  - Yes or no
- Multi-class classification
  - More than two classes
  - More complicated than binary classfication

yes, motorbike

no

motorbike

bike

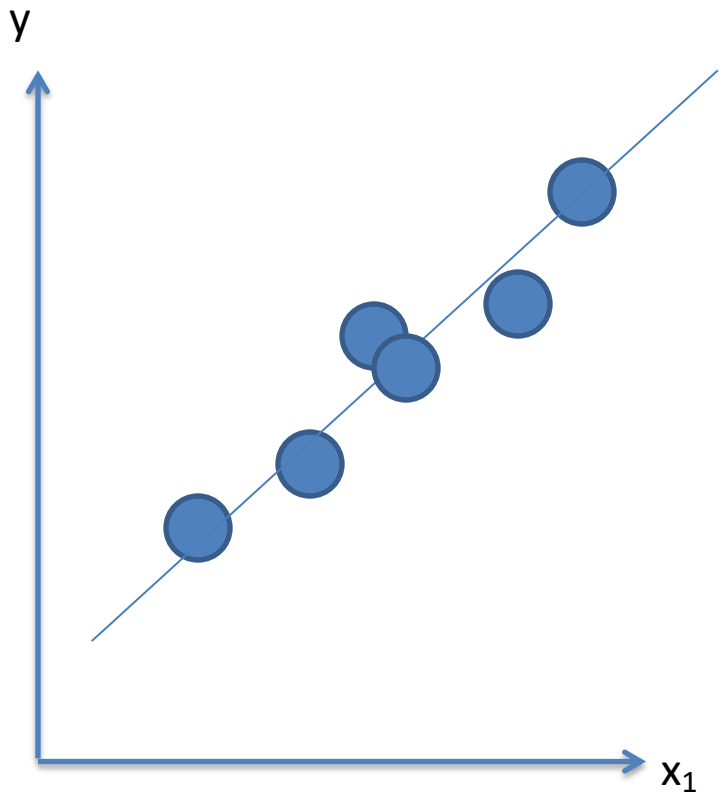
car

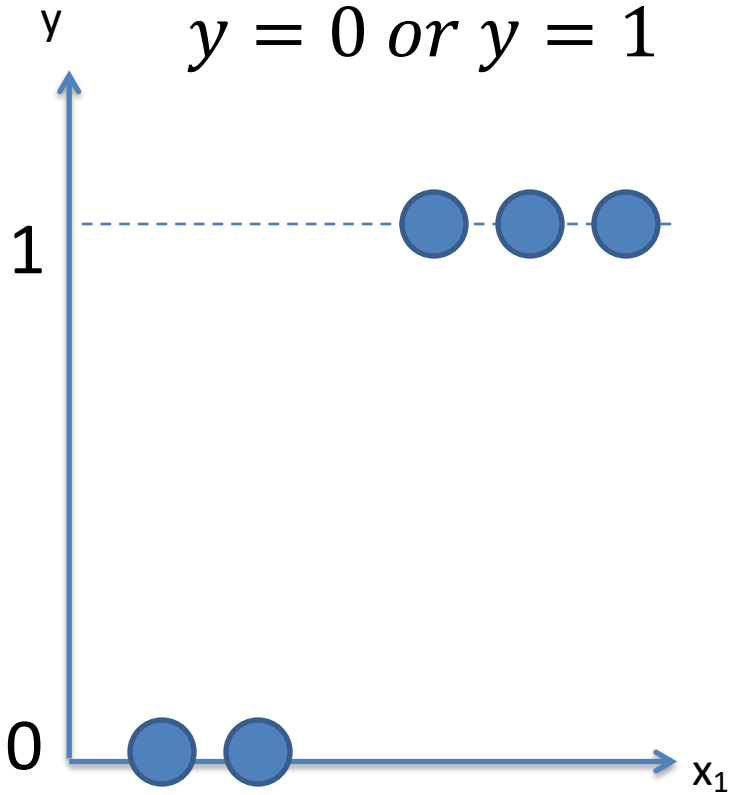
cyclo

something else

Logistic Regression

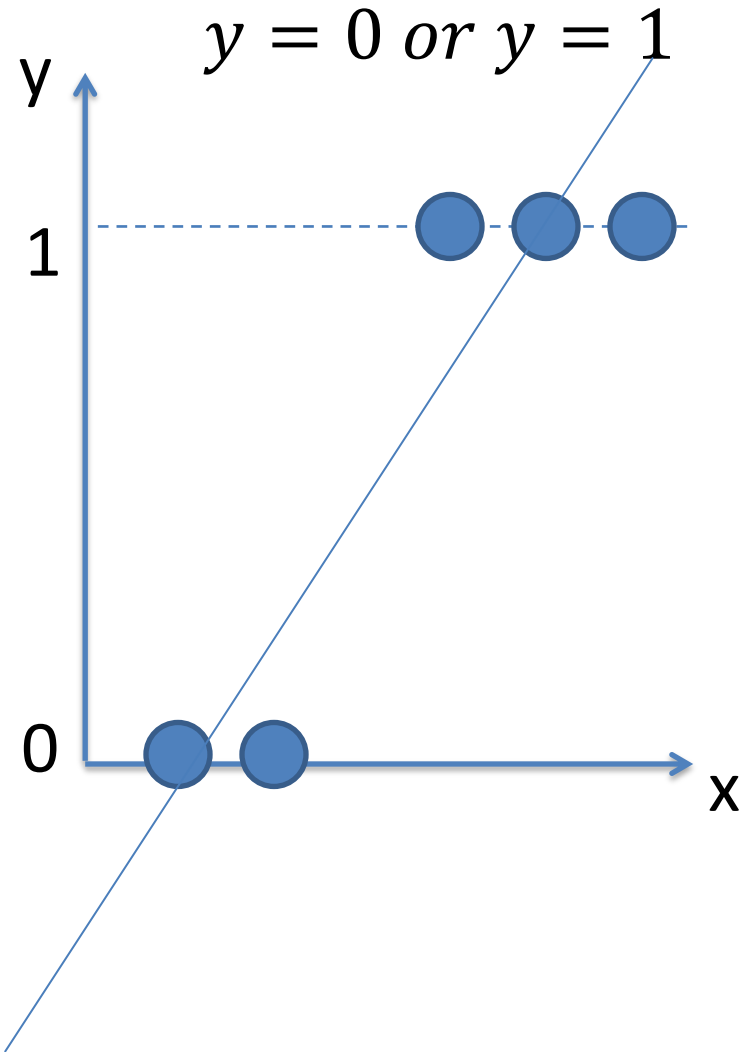
- Solves classification problem, e.g.
  - Email: spam or not spam
  - Sentiment: positive or negative





Logistic Regression

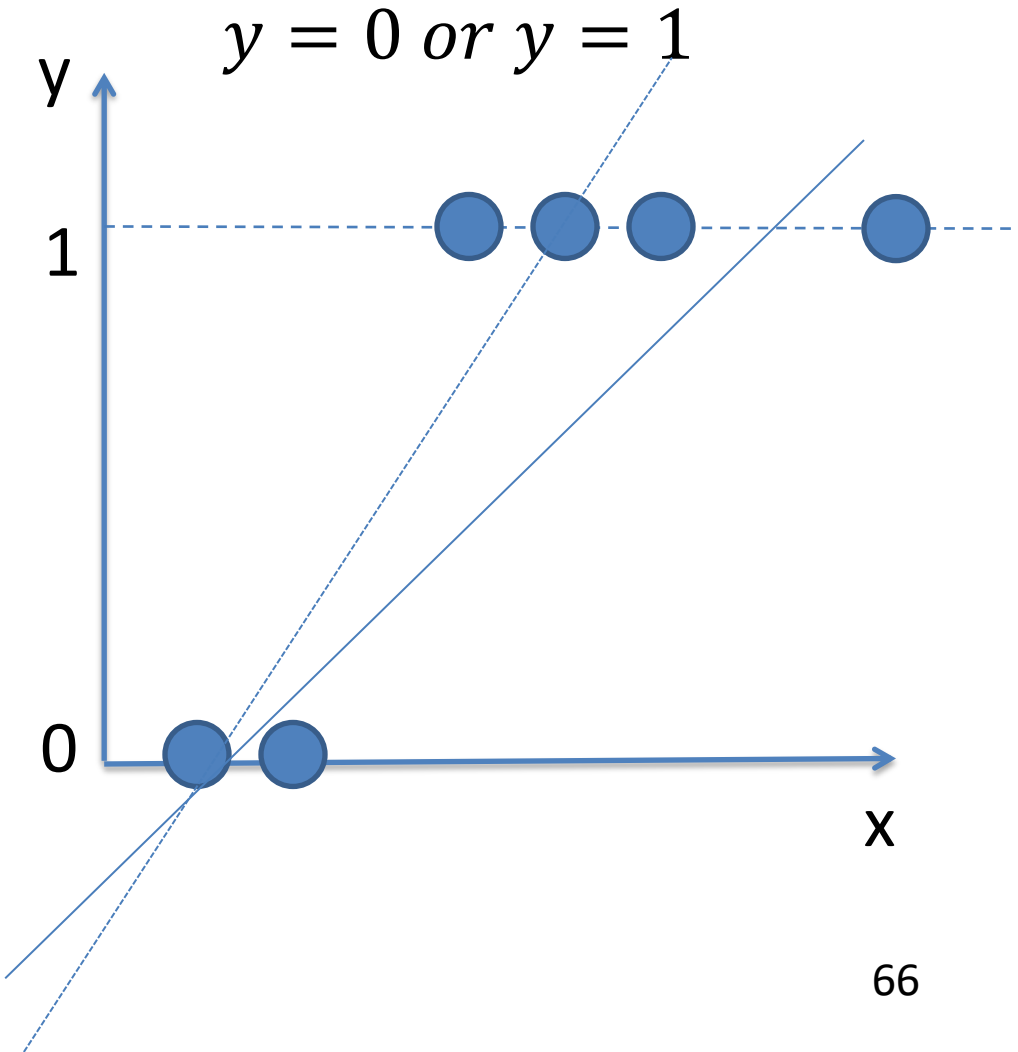
- Idea: Use linear regression



- $h = w_0x_0 + w_1x_1 + \ \ldots + w_nx_n + \ b = \ w^Tx + b$
- $if\ h \geq 0.5\ then\ y = 1\ else\ 0$

Logistic Regression

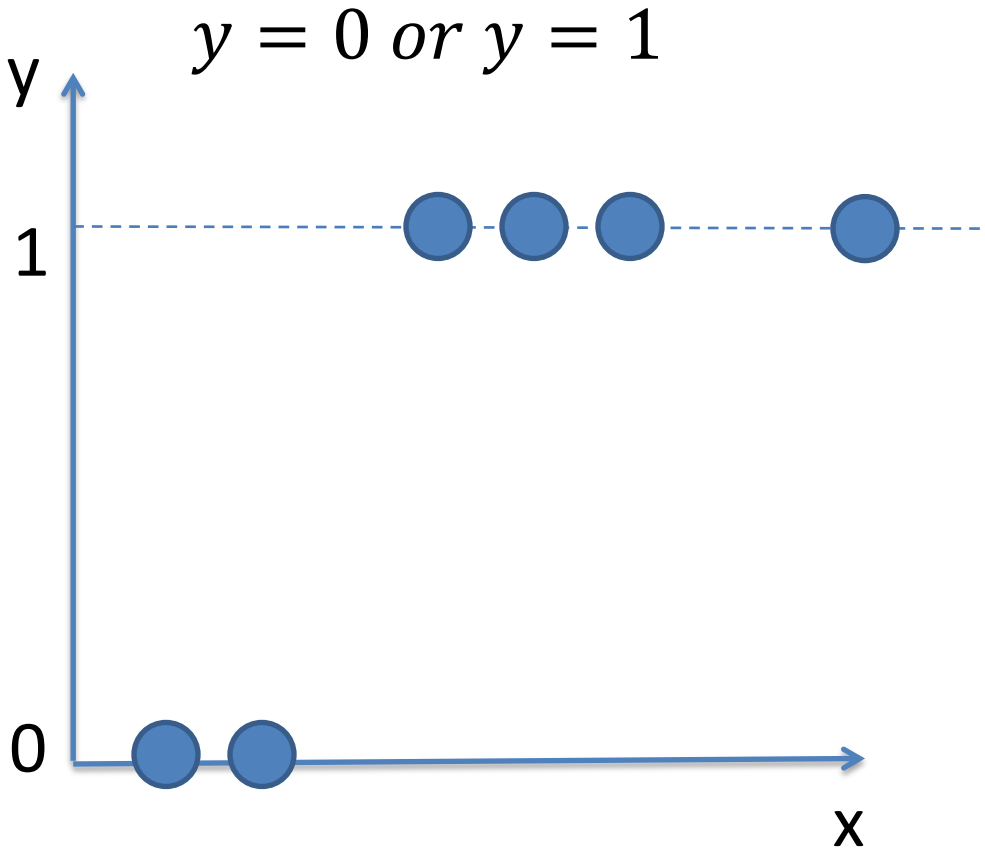
- What happen when we have some very large x?



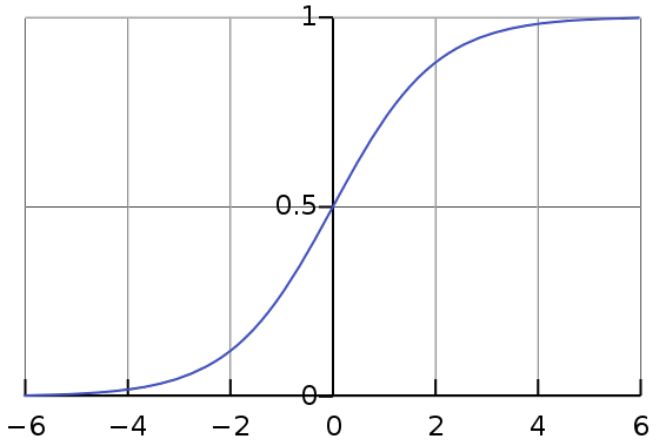
=> Will change our regression line (unnecessarily!)

=> Linear regression is not good for classification

Logistic Regression



- $h(x) = w^T x + b$
- $o(h) = \frac{1}{1+e^{-h}}$

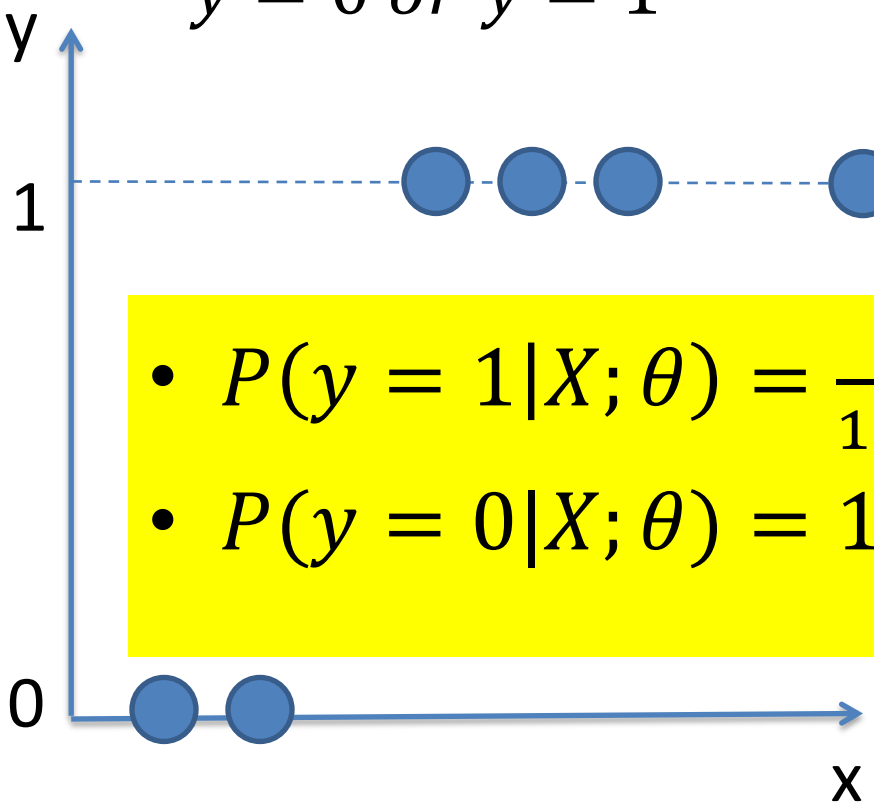


Logistic Regression

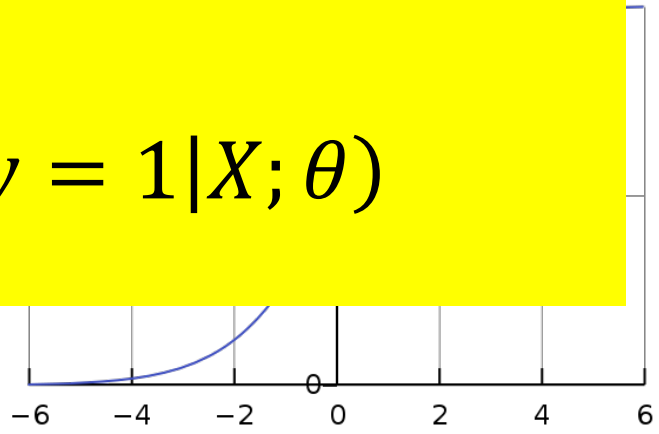
Logistic (sigmoid) function

$y = 0 \; or \; y = 1$

- $h(x) = w^T x + b$
- $o(h) = \frac{1}{1+e^{-h}}$



- $P(y = 1|X;\theta) = \frac{1}{1+e^{-h}}$
- $P(y = 0|X;\theta) = 1 - P(y = 1|X;\theta)$



Thanks for listening!