Knowledge Discovery and Data Mining

Supervised Learning

Xuan Song
songx@sustech.edu.cn

Overview of Machine Learning

What is Machine Learning

What is Machine Learning

Machine learning ≈ look for function

- Speech Recognition

$f($ $)=$ "Nice to meet you!"

- Image Recognition

$f($ $)=$ "0"

- Dialogue System

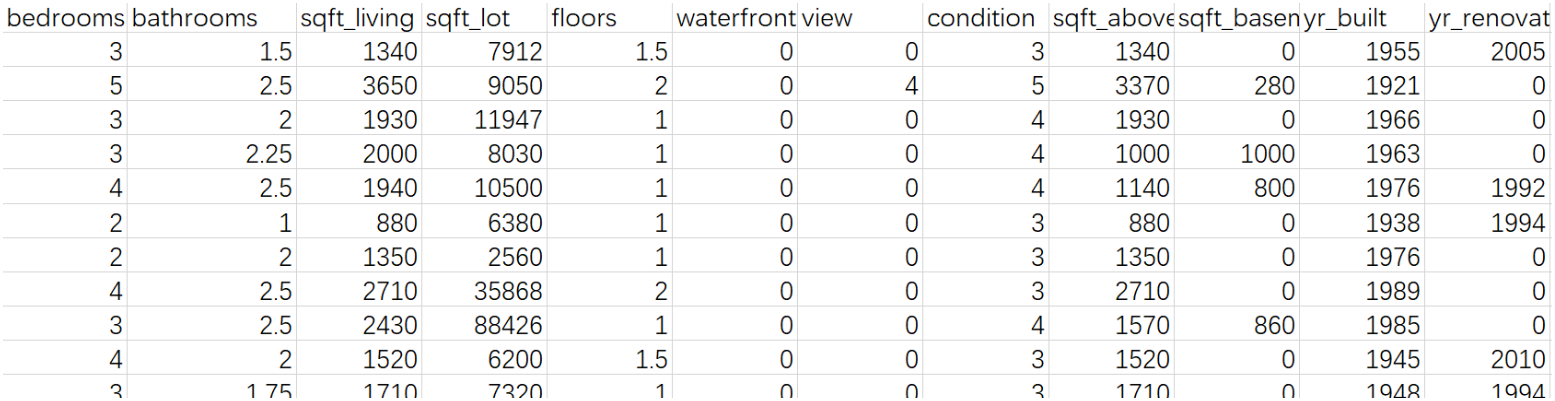
$f($ "How are you" $)=$ "I am good"

Retrieved from *https://speech.ee.ntu.edu.tw/~tlkagk/courses/ML2020*

How to find a function

- Case study of house price prediction

$y = \boldsymbol{f}($



bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basen	yr_built	yr_renovat
3	1.5	1340	7912	1.5	0	0	3	1340	0	1955	2005
5	2.5	3650	9050	2	0	4	5	3370	280	1921	0
3	2	1930	11947	1	0	0	4	1930	0	1966	0
3	2.25	2000	8030	1	0	0	4	1000	1000	1963	0
4	2.5	1940	10500	1	0	0	4	1140	800	1976	1992
2	1	880	6380	1	0	0	3	880	0	1938	1994
2	2	1350	2560	1	0	0	3	1350	0	1976	0
4	2.5	2710	35868	2	0	0	3	2710	0	1989	0
3	2.5	2430	88426	1	0	0	4	1570	860	1985	0
4	2	1520	6200	1.5	0	0	3	1520	0	1945	2010
3	1.75	1710	7320	1	0	0	3	1710	0	1948	1994

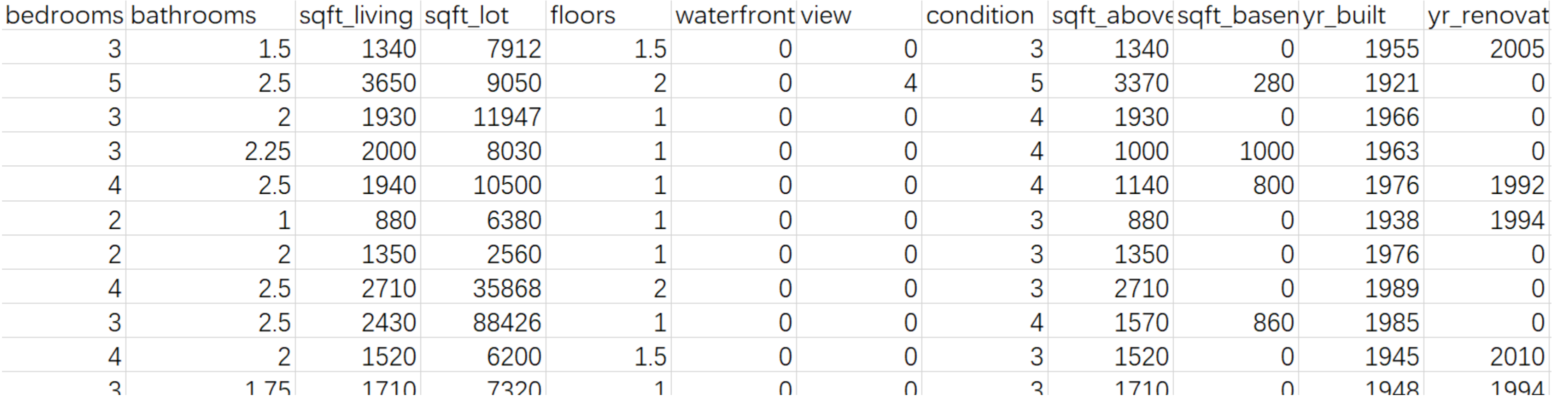
$)$

The function we want to find by machine learning

Step1

- Assume a function with unknown parameters

y = ***f***(



bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basen	yr_built	yr_renovat
3	1.5	1340	7912	1.5	0	0	3	1340	0	1955	2005
5	2.5	3650	9050	2	0	4	5	3370	280	1921	0
3	2	1930	11947	1	0	0	4	1930	0	1966	0
3	2.25	2000	8030	1	0	0	4	1000	1000	1963	0
4	2.5	1940	10500	1	0	0	4	1140	800	1976	1992
2	1	880	6380	1	0	0	3	880	0	1938	1994
2	2	1350	2560	1	0	0	3	1350	0	1976	0
4	2.5	2710	35868	2	0	0	3	2710	0	1989	0
3	2.5	2430	88426	1	0	0	4	1570	860	1985	0
4	2	1520	6200	1.5	0	0	3	1520	0	1945	2010
3	1.75	1710	7320	1	0	0	3	1710	0	1948	1994

)

Domain knowledge

$$y = w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} + b$$

y: *prediction price*

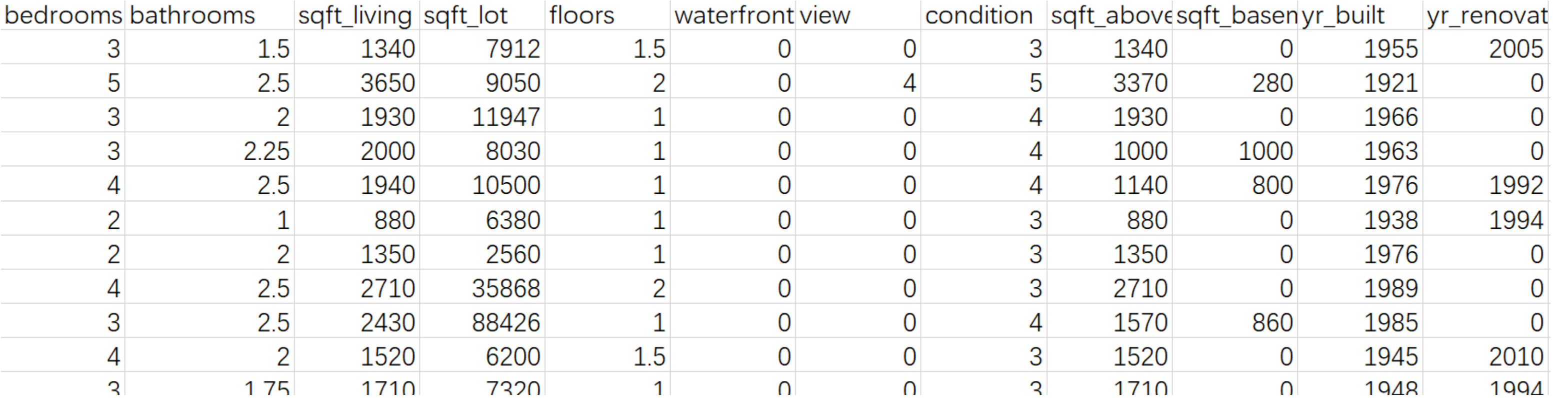
x_i: $\boldsymbol{i}$*th column in house price table*

w_i, b: unkown parameters

Step1

- Assume a function with unknown parameters

y = ***f***(



bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basen	yr_built	yr_renovat
3	1.5	1340	7912	1.5	0	0	3	1340	0	1955	2005
5	2.5	3650	9050	2	0	4	5	3370	280	1921	0
3	2	1930	11947	1	0	0	4	1930	0	1966	0
3	2.25	2000	8030	1	0	0	4	1000	1000	1963	0
4	2.5	1940	10500	1	0	0	4	1140	800	1976	1992
2	1	880	6380	1	0	0	3	880	0	1938	1994
2	2	1350	2560	1	0	0	3	1350	0	1976	0
4	2.5	2710	35868	2	0	0	3	2710	0	1989	0
3	2.5	2430	88426	1	0	0	4	1570	860	1985	0
4	2	1520	6200	1.5	0	0	3	1520	0	1945	2010
3	1.75	1710	7320	1	0	0	3	1710	0	1948	1994

)

Domain knowledge

model

$$\boxed{y = w_1 x_1 + w_2 x_2 + \cdots + w_{12} x_{12} + b}$$

y: prediction price

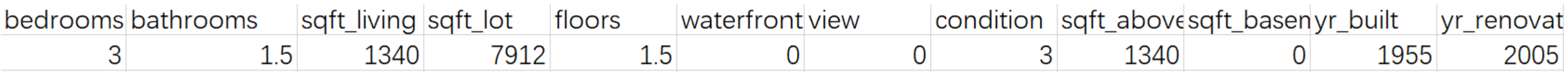
$\boxed{x_i:\ \textit{ith column in house price table}}$ features

weight $\boxed{w_i}$, $\boxed{b}$: unkown parameters

bias

Step2

- Define loss from training data



bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basen	yr_built	yr_renovat
3	1.5	1340	7912	1.5	0	0	3	1340	0	1955	2005

Predicted value

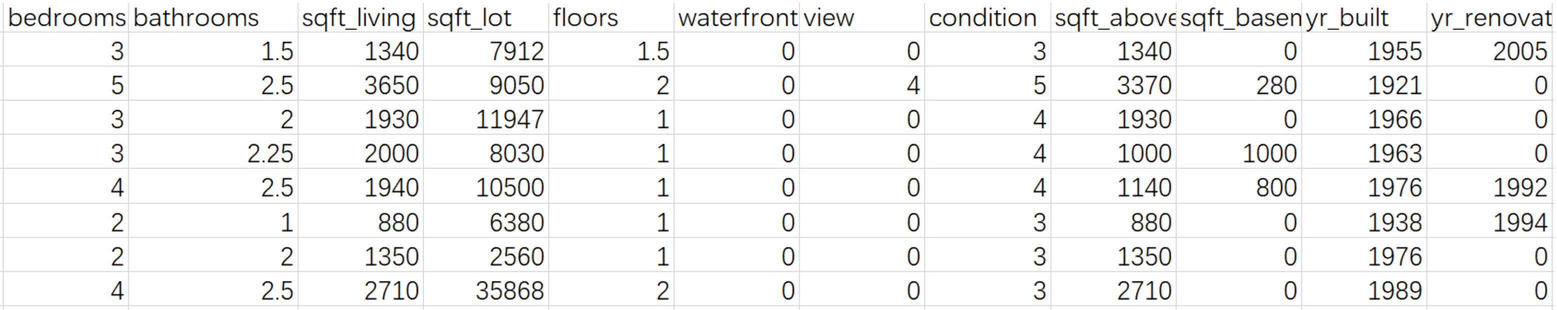
$$y = w_1x_1 + w_2x_2 + \cdots + w_{12}x_{12} + b$$
$$= 3w_1 + 1.5w_2 + \cdots + 2005w_{12} + b$$

Real house price $\bar{y}$

$$\mathrm{e} = |y - \bar{y}|$$

Step2

- Define loss from training data



bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basen	yr_built	yr_renovat
3	1.5	1340	7912	1.5	0	0	3	1340	0	1955	2005
5	2.5	3650	9050	2	0	4	5	3370	280	1921	0
3	2	1930	11947	1	0	0	4	1930	0	1966	0
3	2.25	2000	8030	1	0	0	4	1000	1000	1963	0
4	2.5	1940	10500	1	0	0	4	1140	800	1976	1992
2	1	880	6380	1	0	0	3	880	0	1938	1994
2	2	1350	2560	1	0	0	3	1350	0	1976	0
4	2.5	2710	35868	2	0	0	3	2710	0	1989	0

$$e_n = |y_n - \overline{y_n}|$$

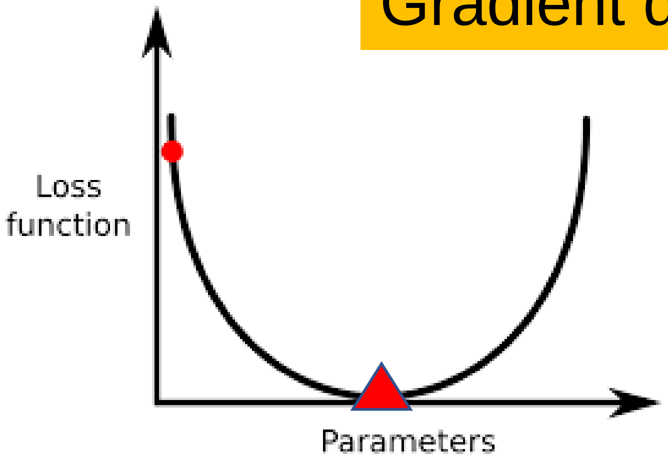
(1) Loss: How good a set of value is?
(2) Loss is a function of parameter(weight and bias).

$$Loss = \frac{1}{N}\sum_n e_n = L(weight, bias)$$

Step3

- Optimization

Gradient descent

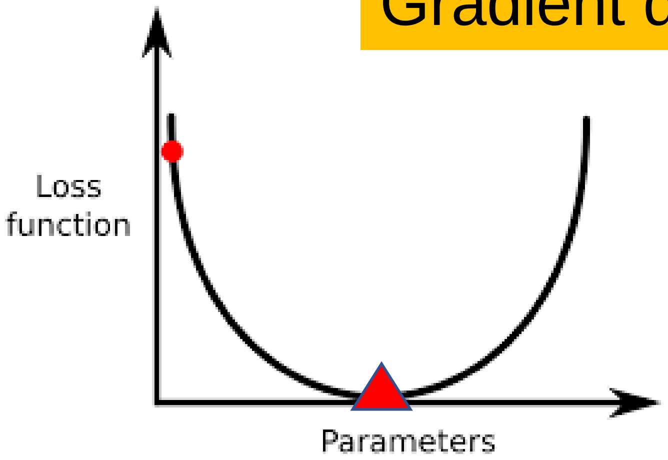


$$w^*, b^* = \arg\min_{w,b} L$$

Step3

- Optimization

Gradient descent

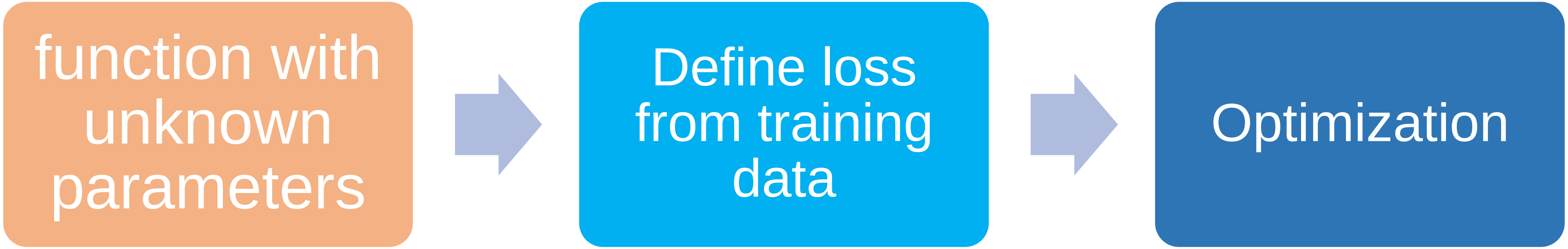


$$w^*, b^* = \arg\min_{w,b} L$$

model $y = w_1{}^* x_1 + w_2{}^* x_2 + \cdots + w_{12}{}^* x_{12} + b^*$ ✓

Prediction

Machine Learning Process



function with unknown parameters

Define loss from training data

Optimization

Video

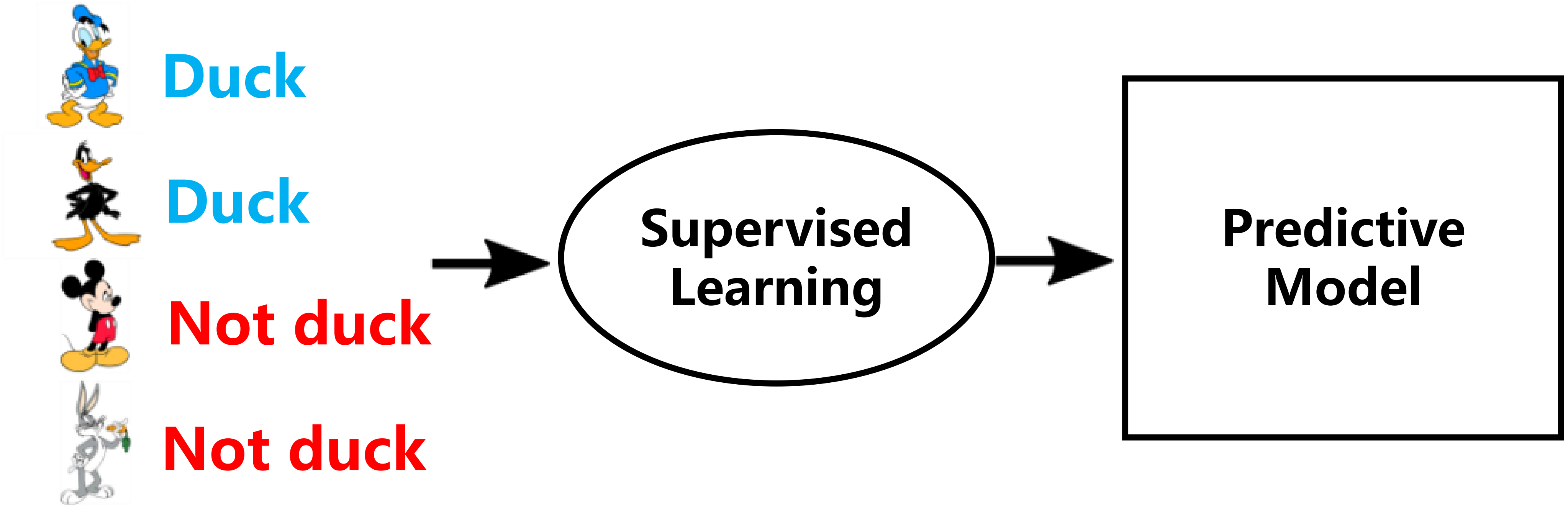
Introduction to Supervised Learning

Types of Learning

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning

Types of Learning

- Supervised Learning



Duck

Duck

Not duck

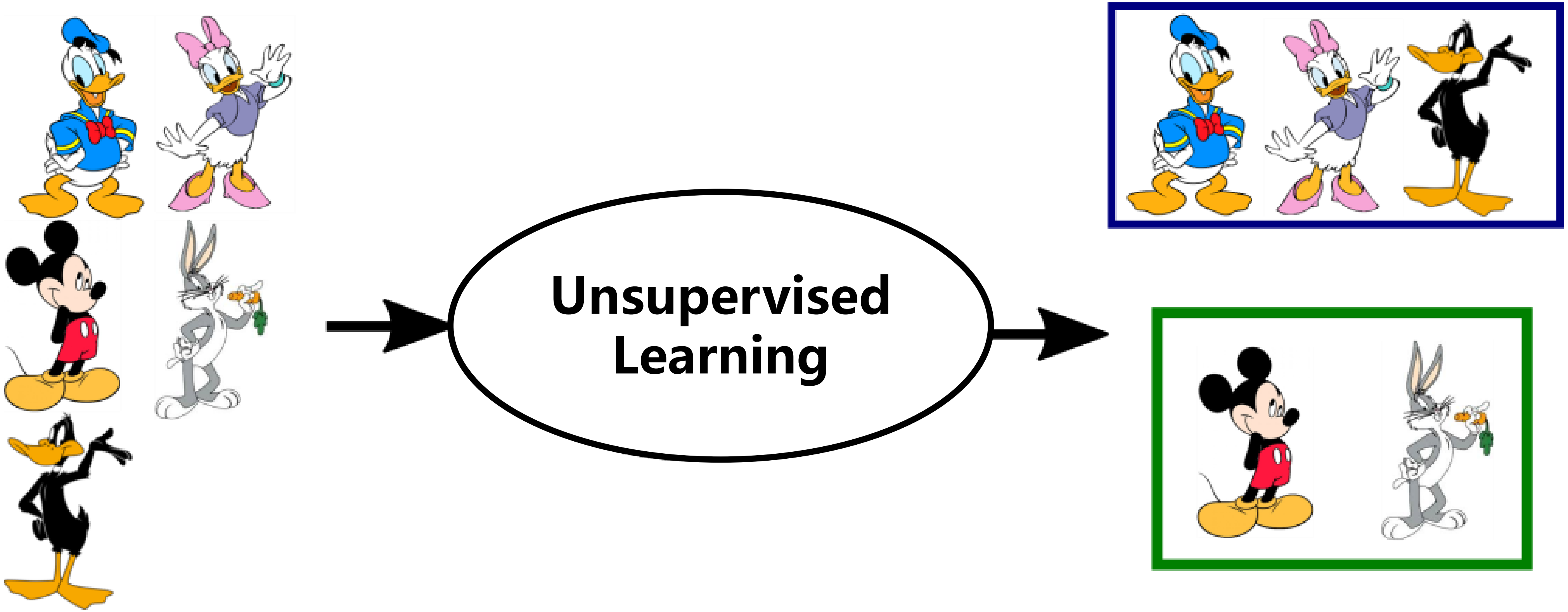
Not duck

Supervised Learning

Predictive Model

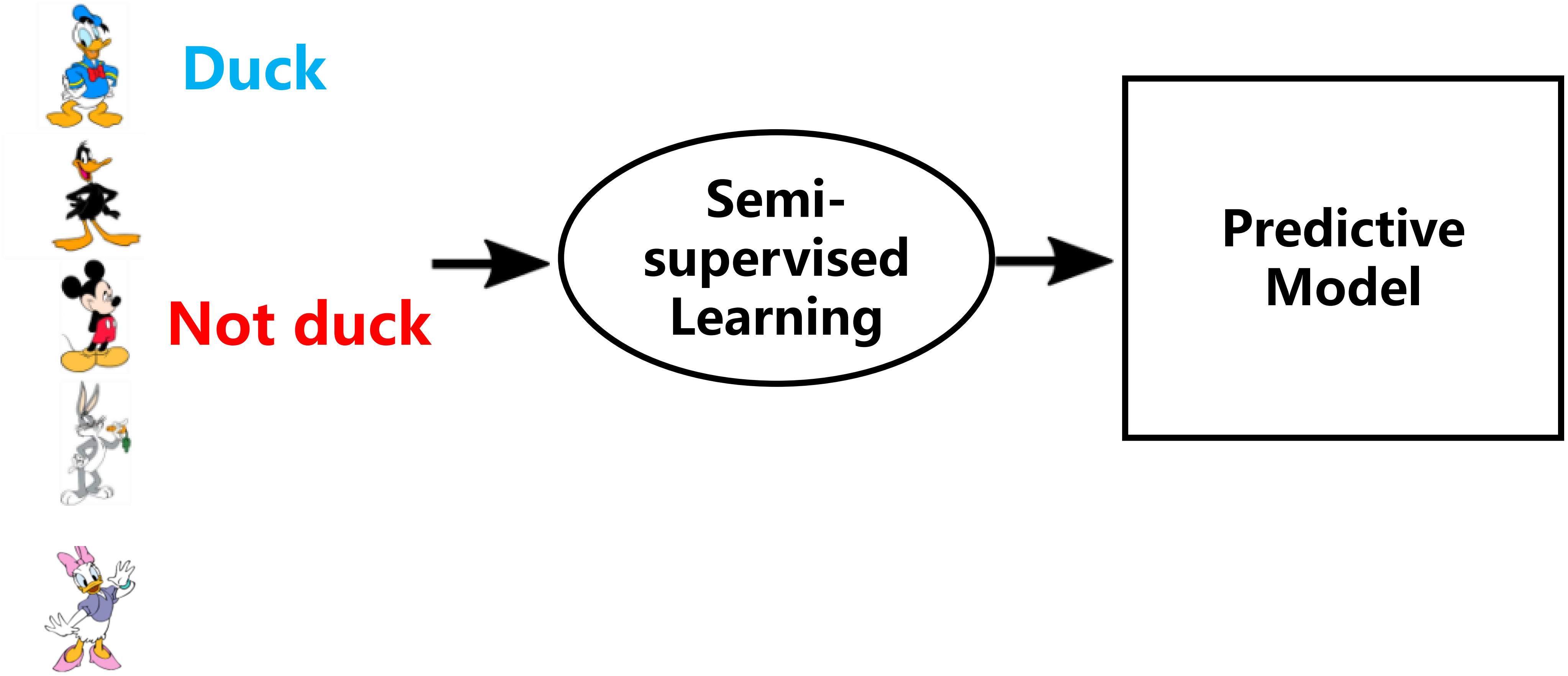
Types of Learning

- Unsupervised Learning



Types of Learning

- Semi-supervised Learning



Duck

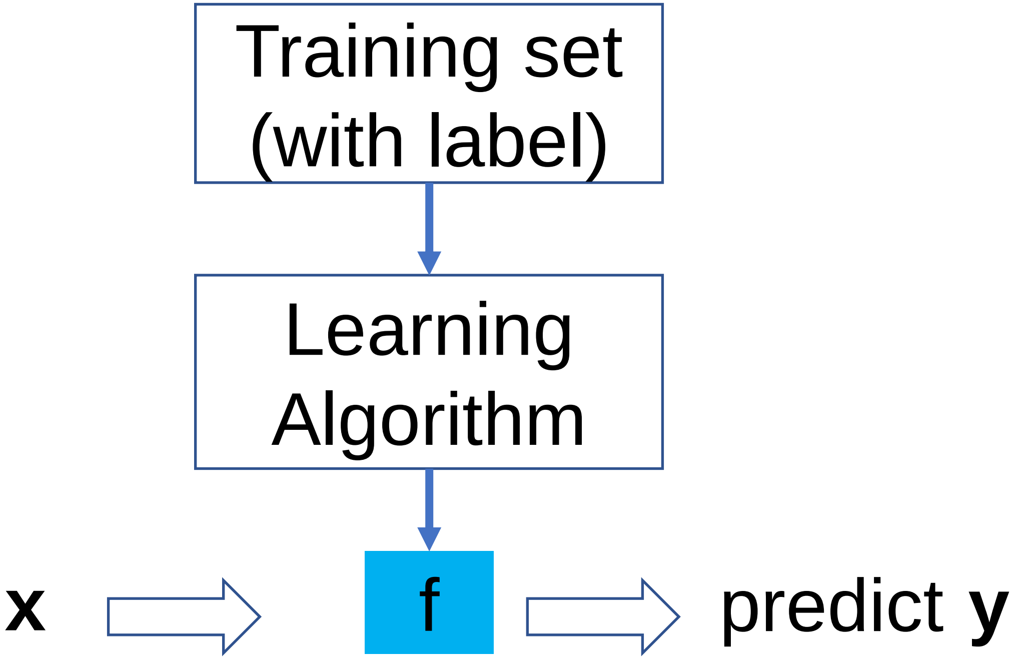
Not duck

Semi-supervised Learning

Predictive Model

Supervised Learning

- Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.



Training set (with label)

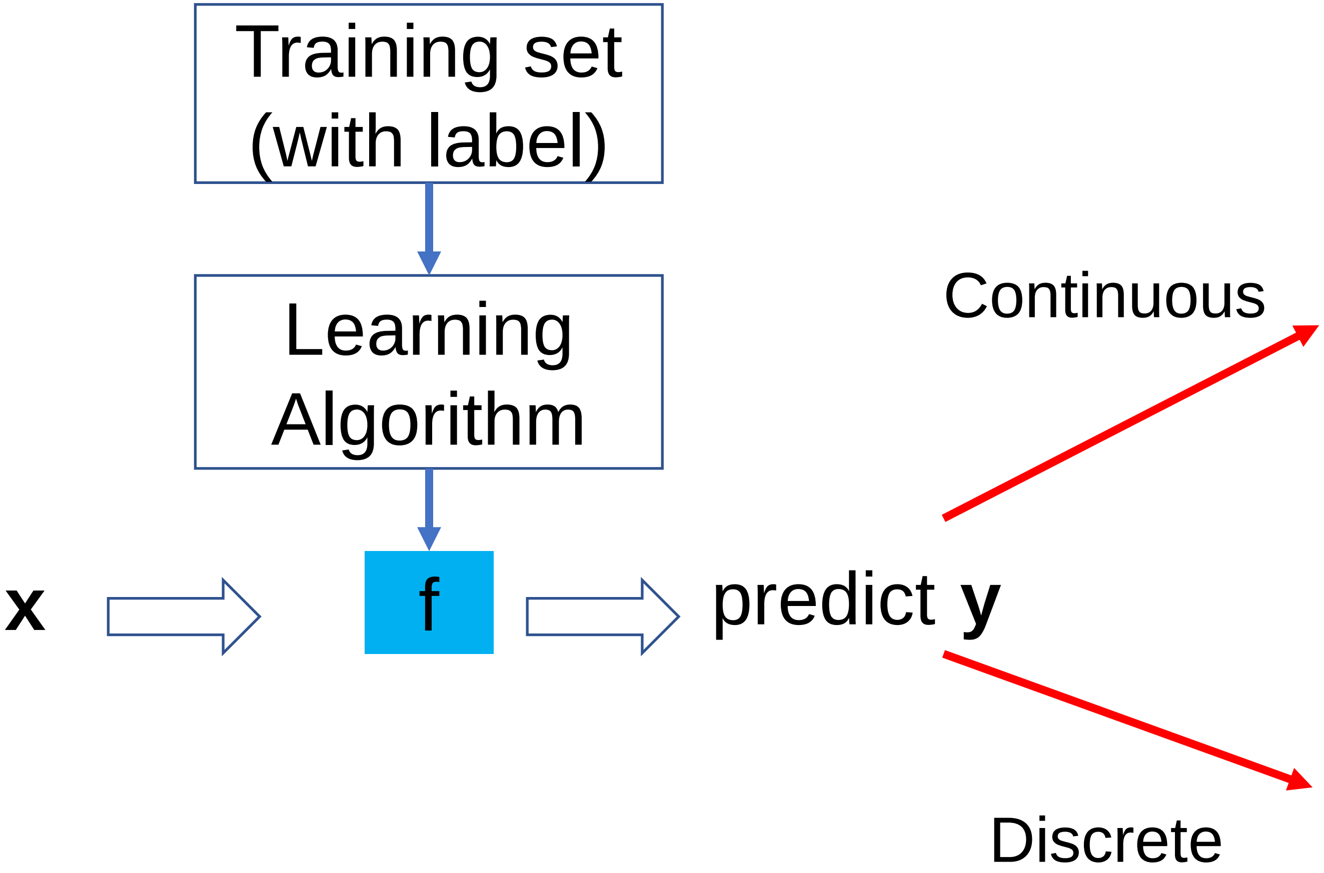
Learning Algorithm

x

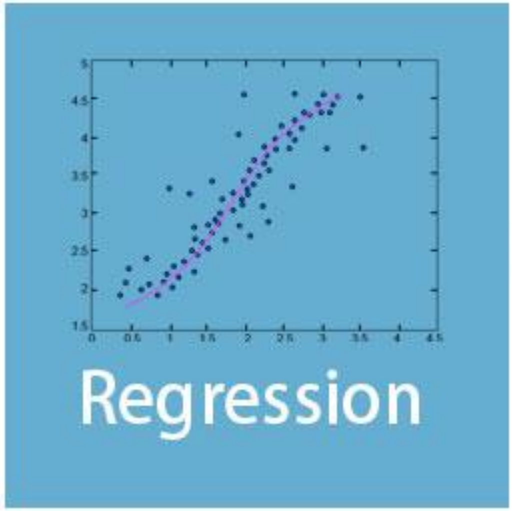
f

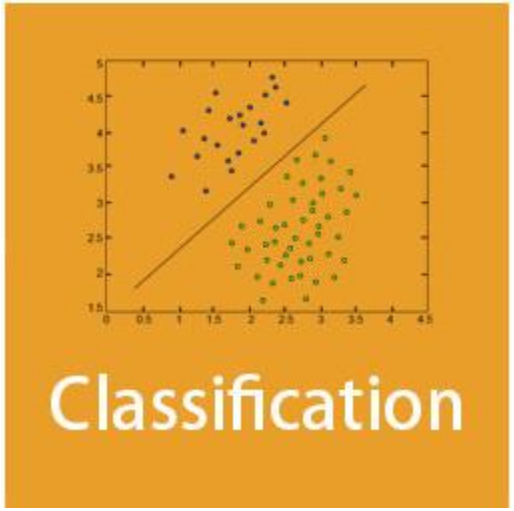
predict **y**

Supervised Learning



Two Tasks

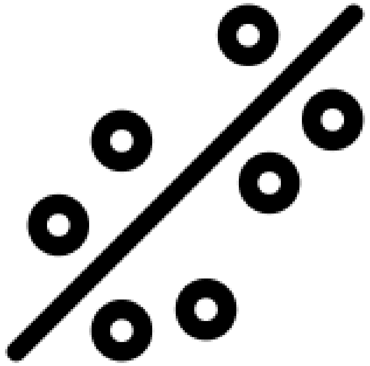


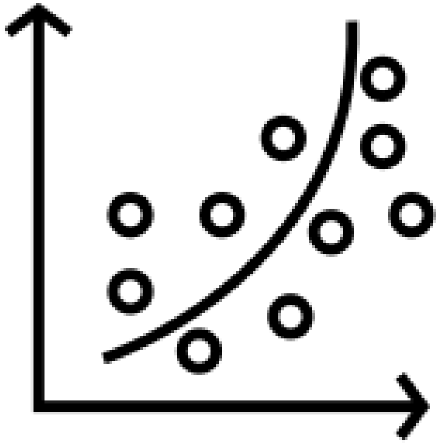


Regression

Regression

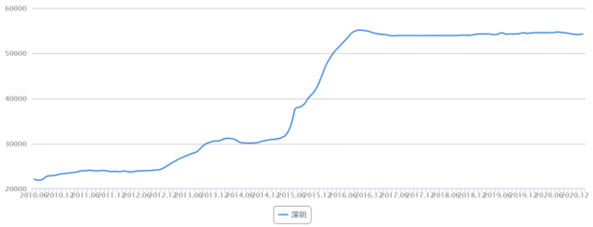
- Predict a value of a given **continuous** valued variable based on the values of other variables, assuming a linear or nonlinear model of dependency.





Regression

- House Price Forecast:

$f($  $)$= Price

- Self-driving Car:

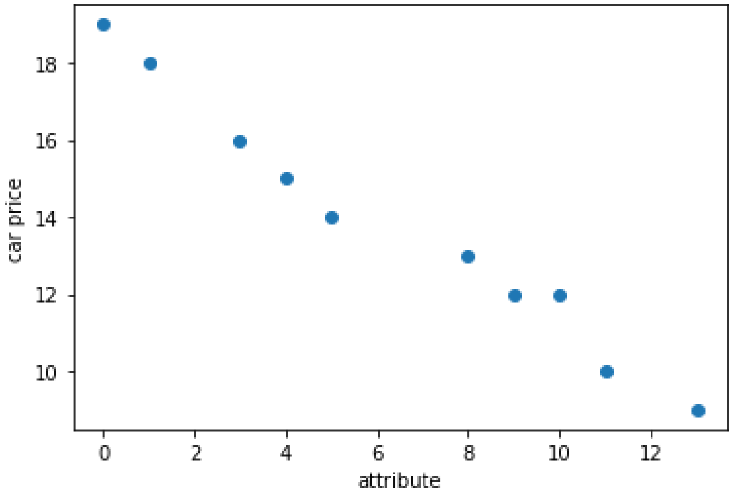
$f($ $)$= Steering Angle

Regression

- Types of Regression Algorithm:
  - Simple Linear Regression
  - Multiple Linear Regression
  - Polynomial Regression
  - Support Vector Regression
  - Decision Tree Regression
  - Random Forest Regression

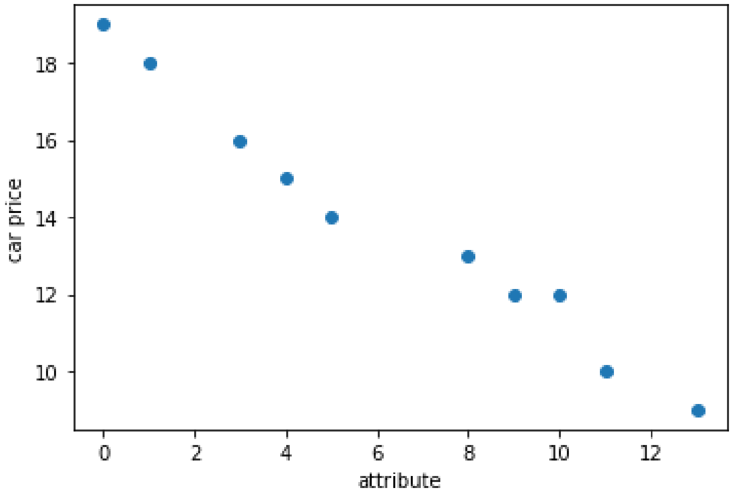
Regression

- Price of a used car



Regression

- Price of a used car



Step1: Model $y = wx + b$

Function Set: f1, f2, f3…

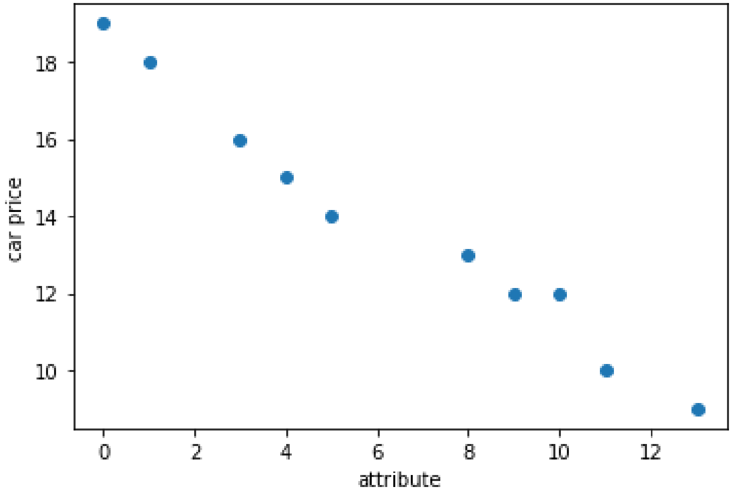
x: $attribute\ of\ car$

y: $price$

w, b: $parameters$

Regression

- Price of a used car



Step1: Model $y = wx + b$

Function Set: f1, f2, f3…

x: *attribute of car*

y: *price*

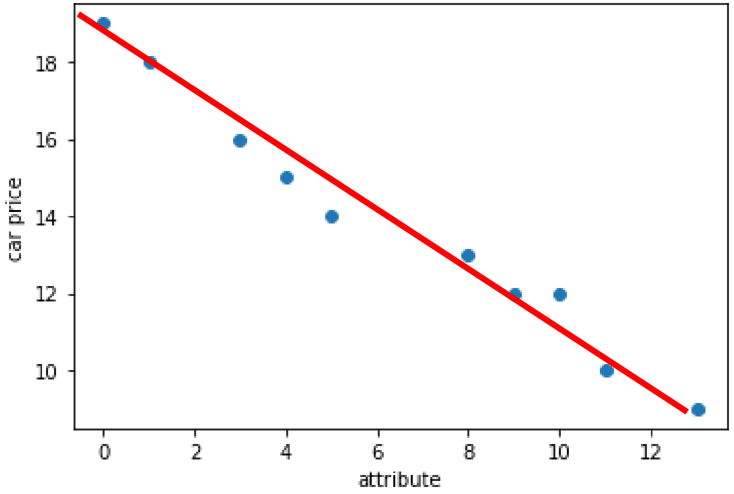
w, b: *parameters*

Step2: Goodness of Function $L(f) = \sum_{n=1}^{10} \boxed{(\bar{y}^n - f(x^n))^2}$ Estimation error

$$L(f) = \sum_{n=1}^{10} (\bar{y}^n - (b + wx^n))^2$$

Simple Linear Regression

- Price of a used car



Step1: Model $y = wx + b$

Function Set: f1, f2, f3…

x: *attribute of car*

y: *price*

w, b: *parameters*

Step2: Goodness of Function $L(f) = \sum_{n=1}^{10} \boxed{(\bar{y}^n - f(x^n))^2}$ Estimation error

$$L(f) = \sum_{n=1}^{10} (\bar{y}^n - (b + wx^n))^2$$

Step3: Pick the "Best Function"

$$\boxed{w^*, b^*} = \arg\min_{w,b} L(w, b)$$

$$= \arg\min_{w,b} \sum_{n=1}^{10} (\bar{y}^n - (b + wx^n))^2$$

Gradient Descent

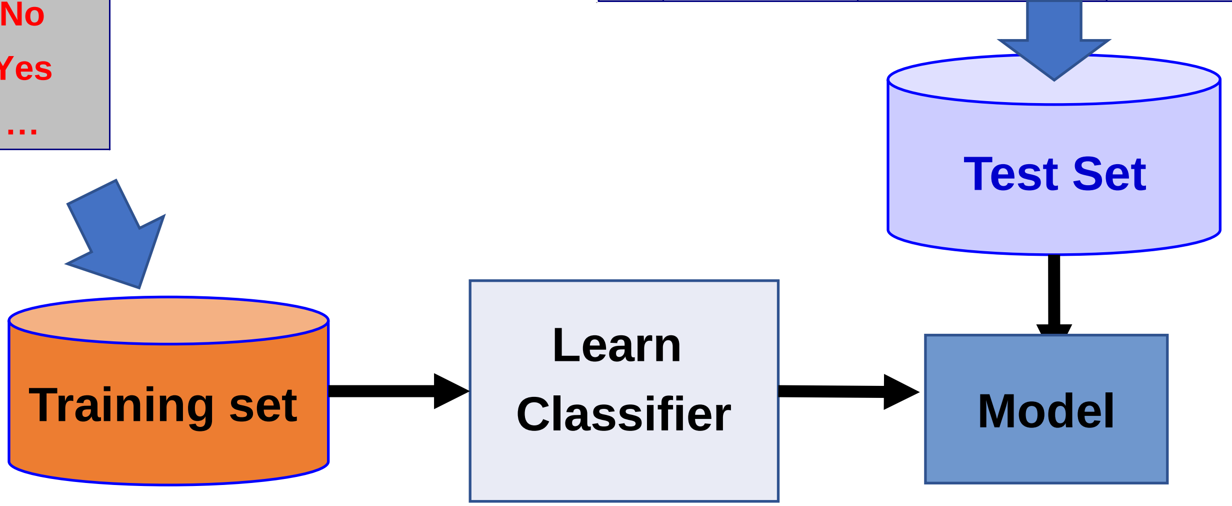
Classification

Classification

- Given a collection of records (training set)
  - Each record contains a set of attributes, one of the attributes is the class.
- Find a model for class attribute as a function of the values of other attributes.

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
…	…	…	…	…

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Undergrad	7	?
2	No	Graduate	3	?
3	Yes	High School	2	?
…	…	…	…	…



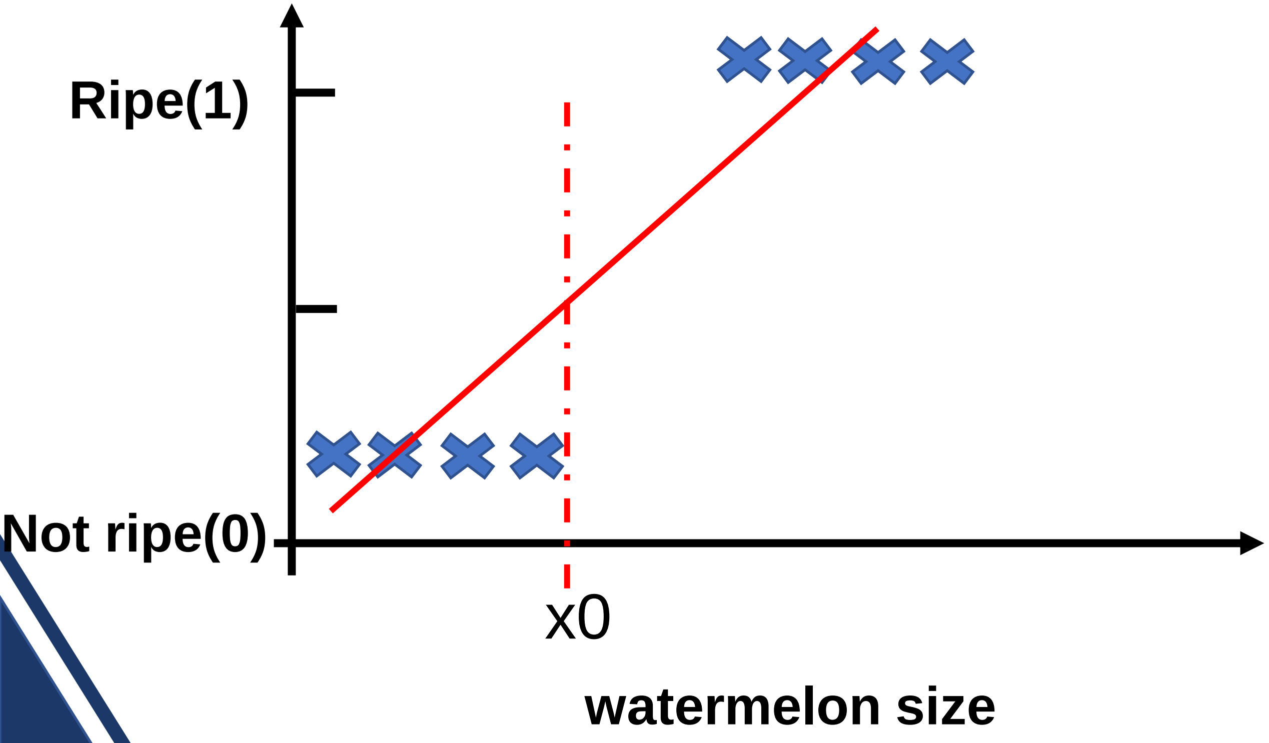
Classification

- Base Classifiers
  - Logistic Regression
  - Decision Tree based Methods
  - Rule-based Methods
  - Nearest-neighbor
  - Neural Networks, Deep Neural Nets
  - Naïve Bayes and Bayesian Belief Networks
  - Support Vector Machines

- Ensemble Classifiers
  - Boosting, Bagging, Random Forests

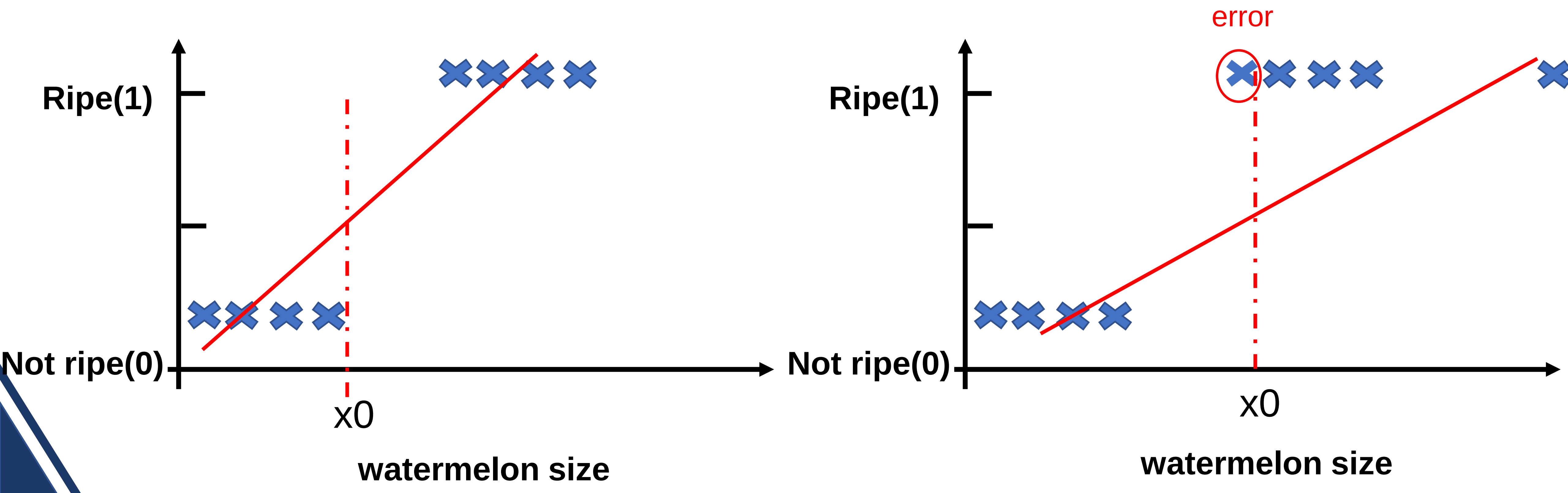
Logistic Regression

- The linear regression model can work well for regression, but fails for classification.



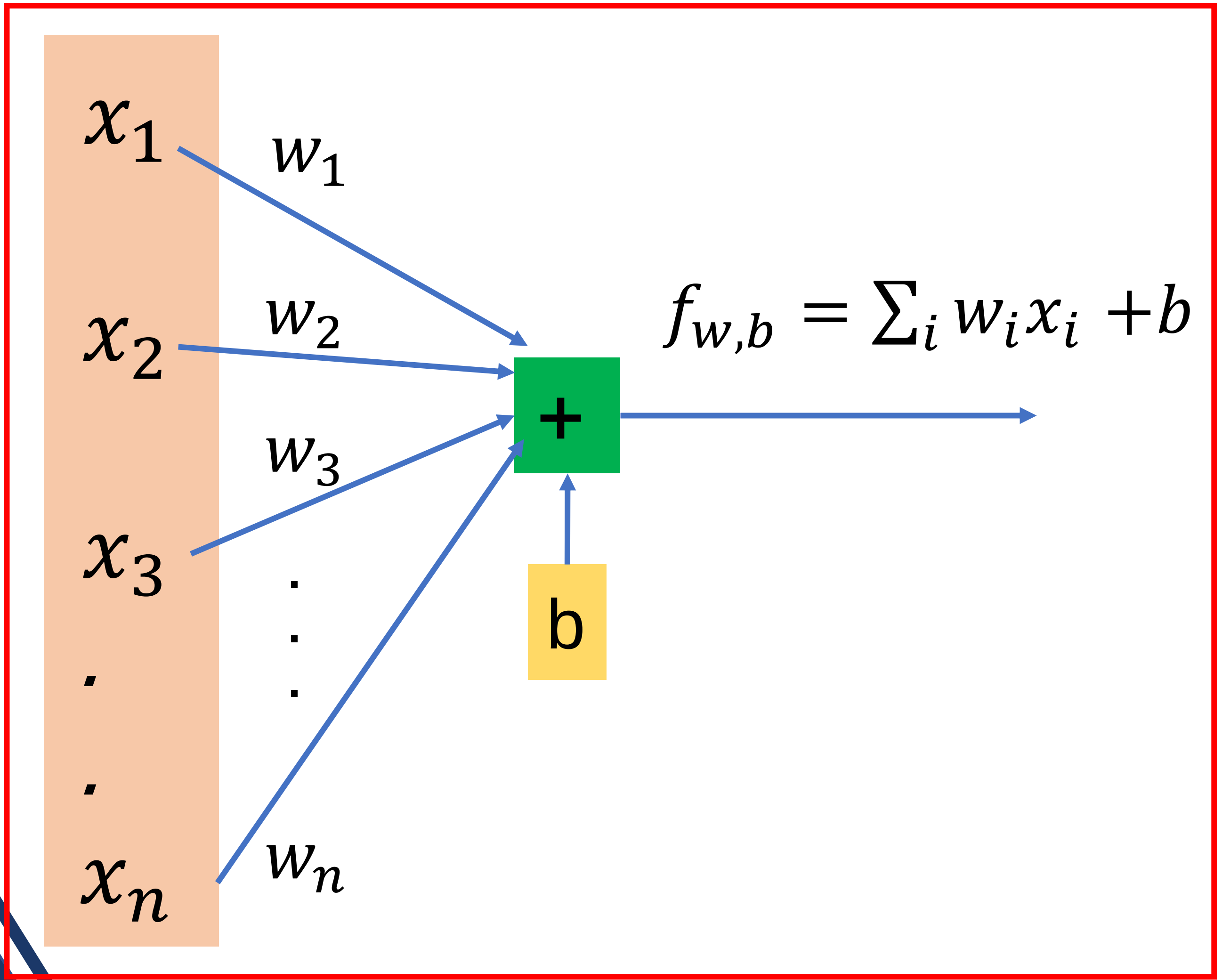
Logistic Regression

- The linear regression model can work well for regression, but fails for classification.



Logistic Regression

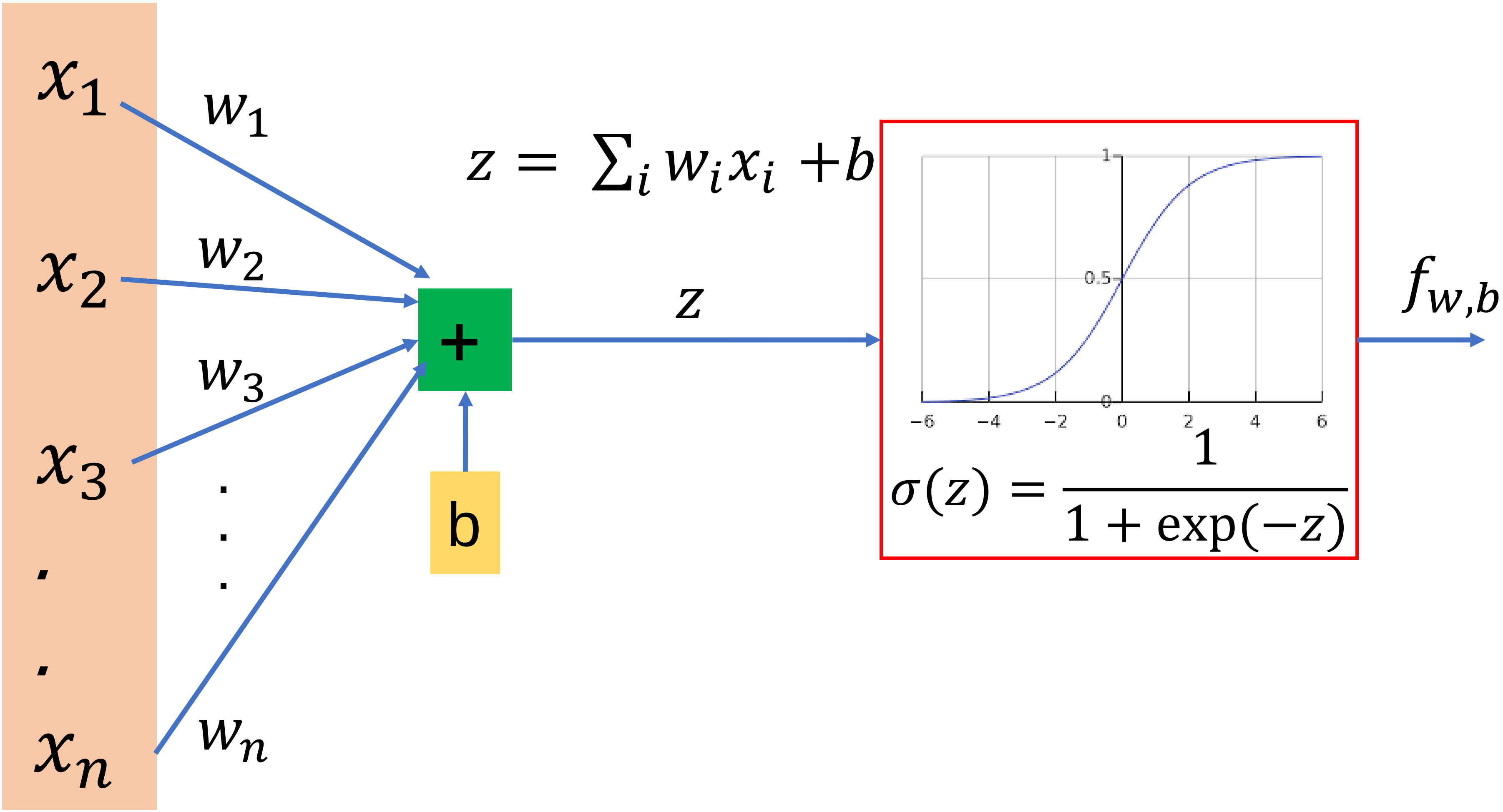
Linear Regression **output: any value**



$$f_{w,b} = \sum_i w_i x_i + b$$

Logistic Regression

output: between 0 and 1



$$z = \sum_i w_i x_i + b$$

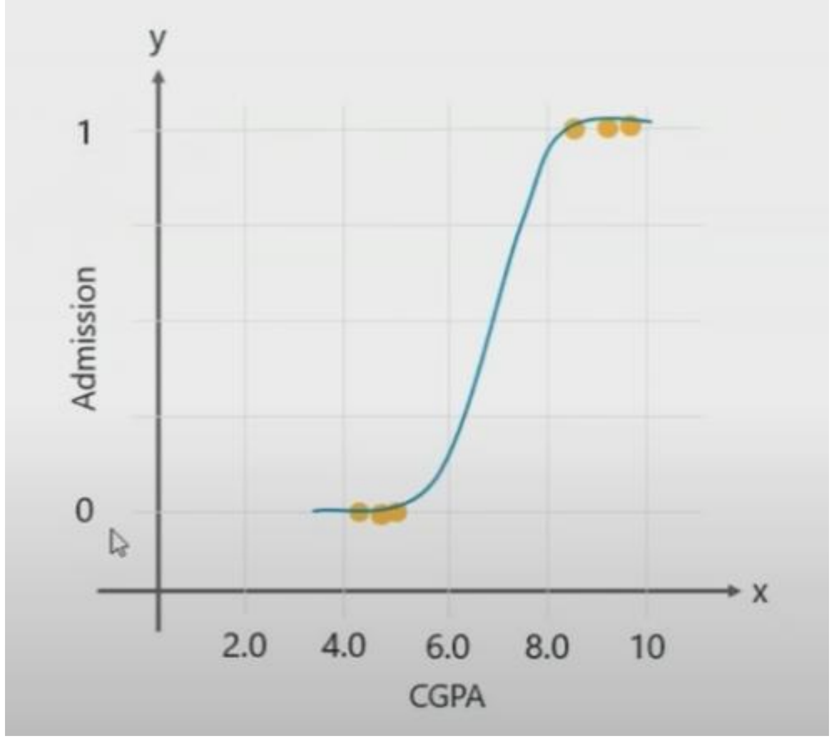
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

$$f_{w,b} = \sigma(\sum_i w_i x_i + b)$$

Logistic Regression Use Case

To predict if a student will be admitted based on his/her CGPA

Admission	CGPA
0	4.2
0	5.1
0	5.5
1	8.2
1	9.0
1	9.1



Retrieved from:
https://www.youtube.com/watch?v=OCwZyYH14uw&ab_channel=edureka%21

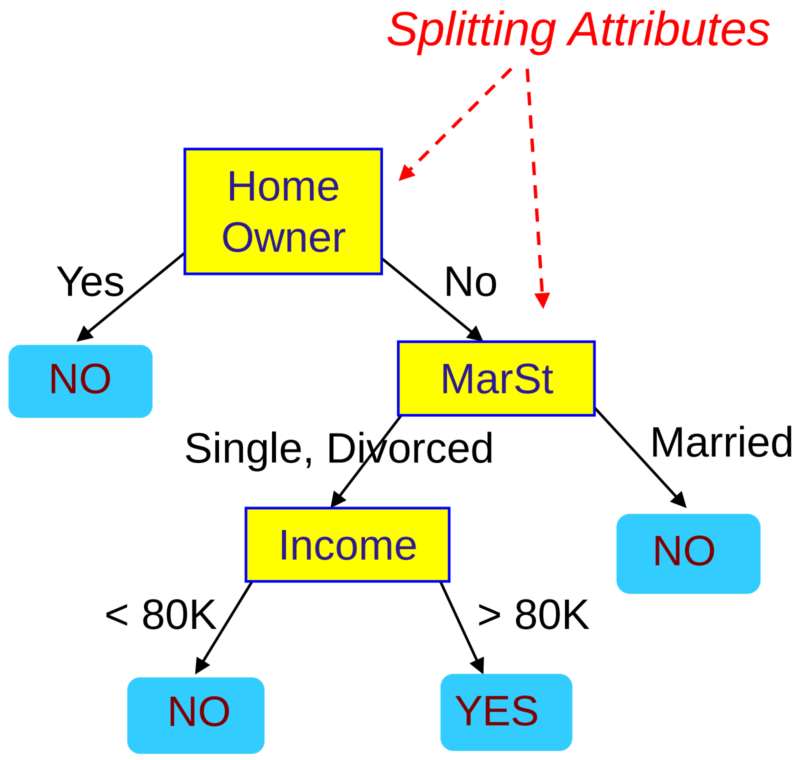
Classification: Decision Tree

- They do classification: predict a categorical output from categorical and/or real inputs
- Decision trees are the single most popular data mining tool
  - Easy to understand
  - Easy to implement
  - Easy to use
  - Computationally cheap
- Mature, Easy-to-use software package freely available
- NO programming needed!

Example of Decision Tree

	categorical	categorical	continuous	class
ID	**Home Owner**	**Marital Status**	**Annual Income**	**Defaulted Borrower**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

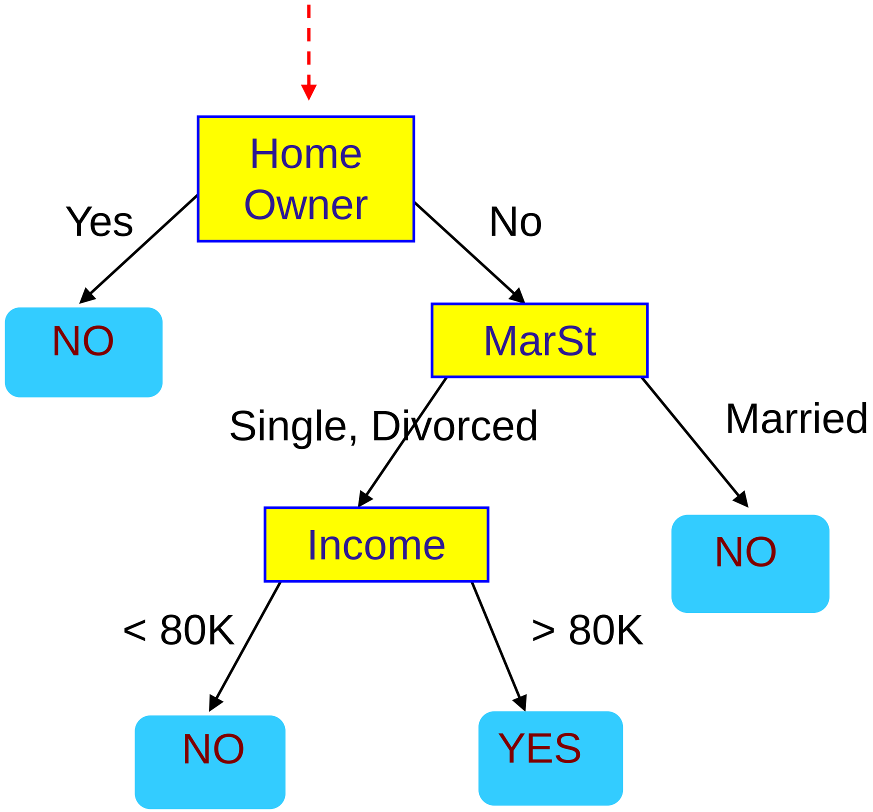
Training Data



Model: Decision Tree

Apply model to test data

Start from the root of tree.



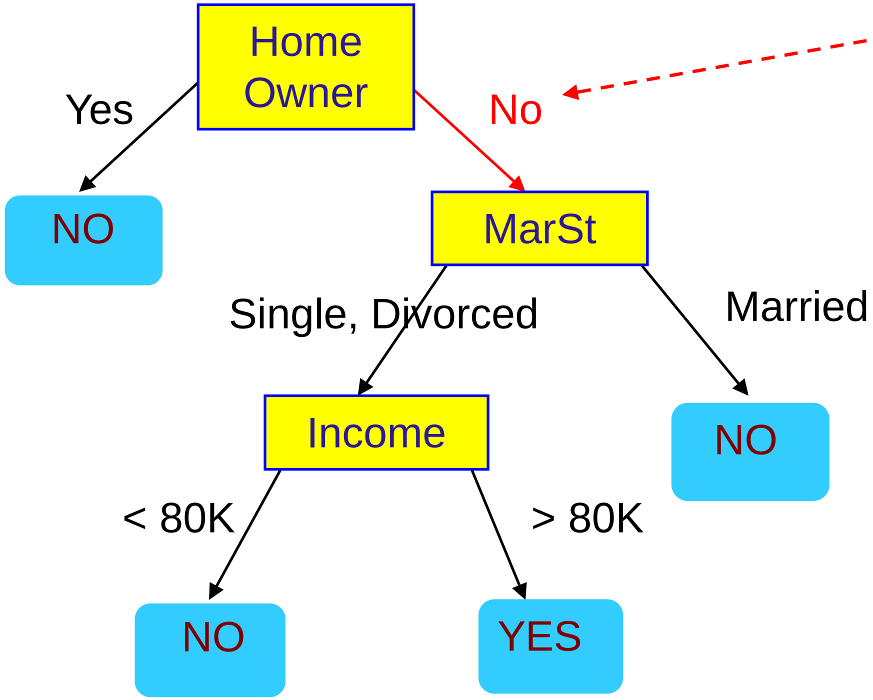
Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

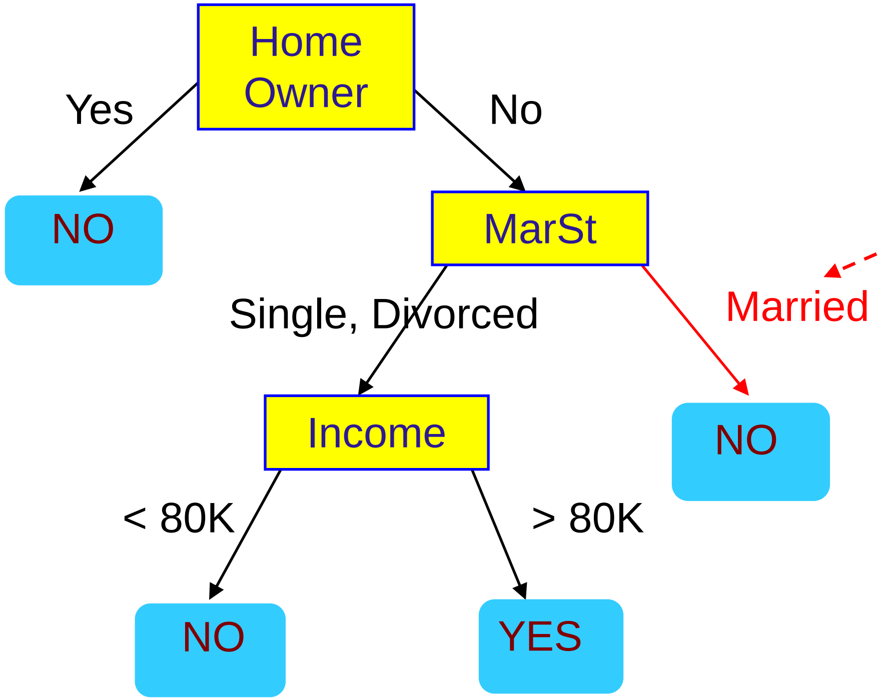
Apply model to test data

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



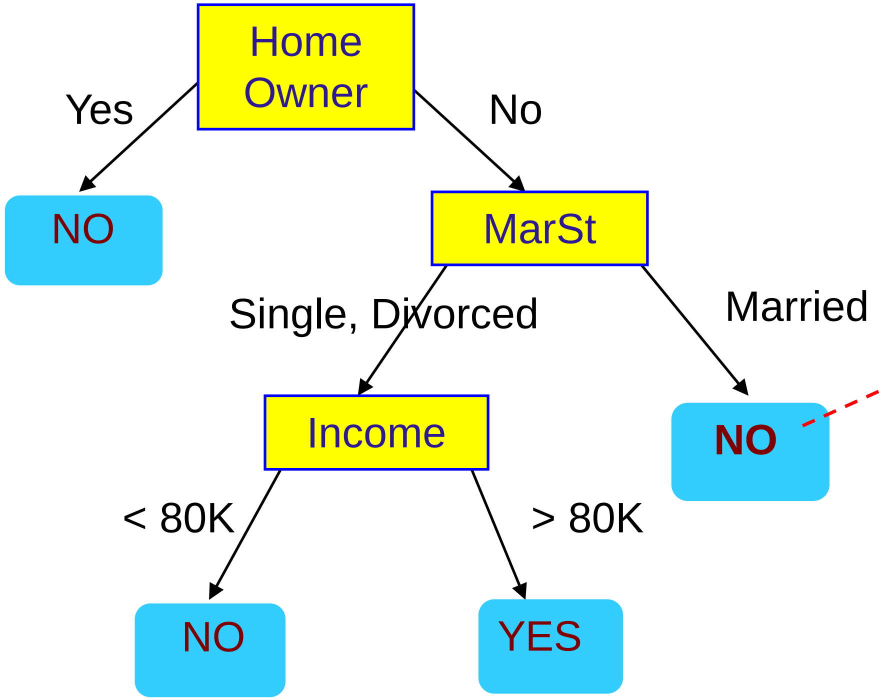
Apply model to test data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Apply model to test data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Assign Defaulted to “No”

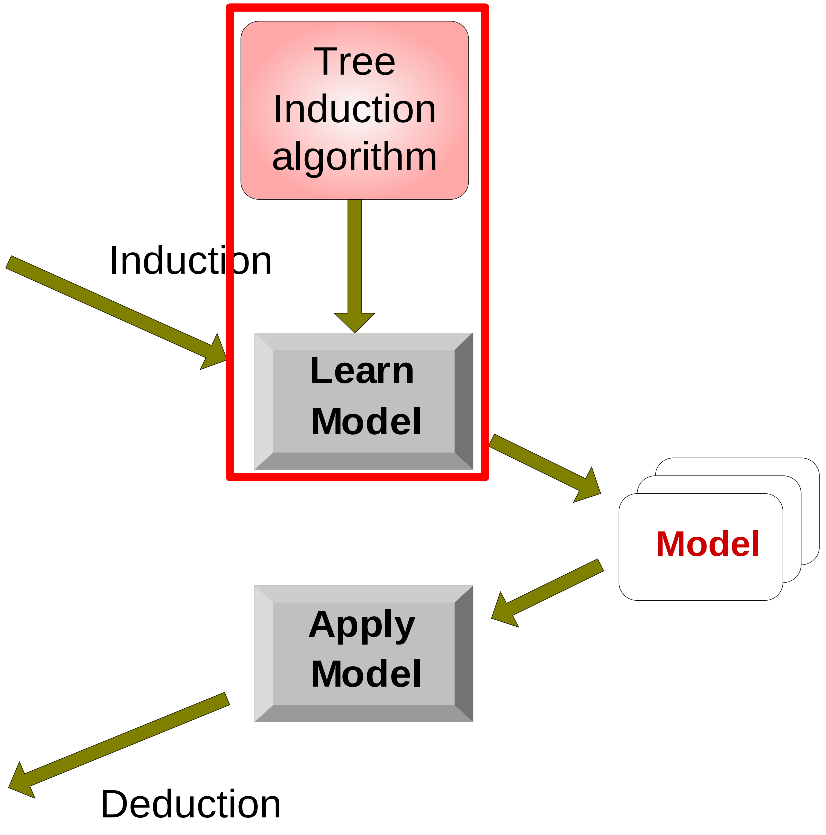
Decision Tree Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Decision Tree Based Classification

Advantages:

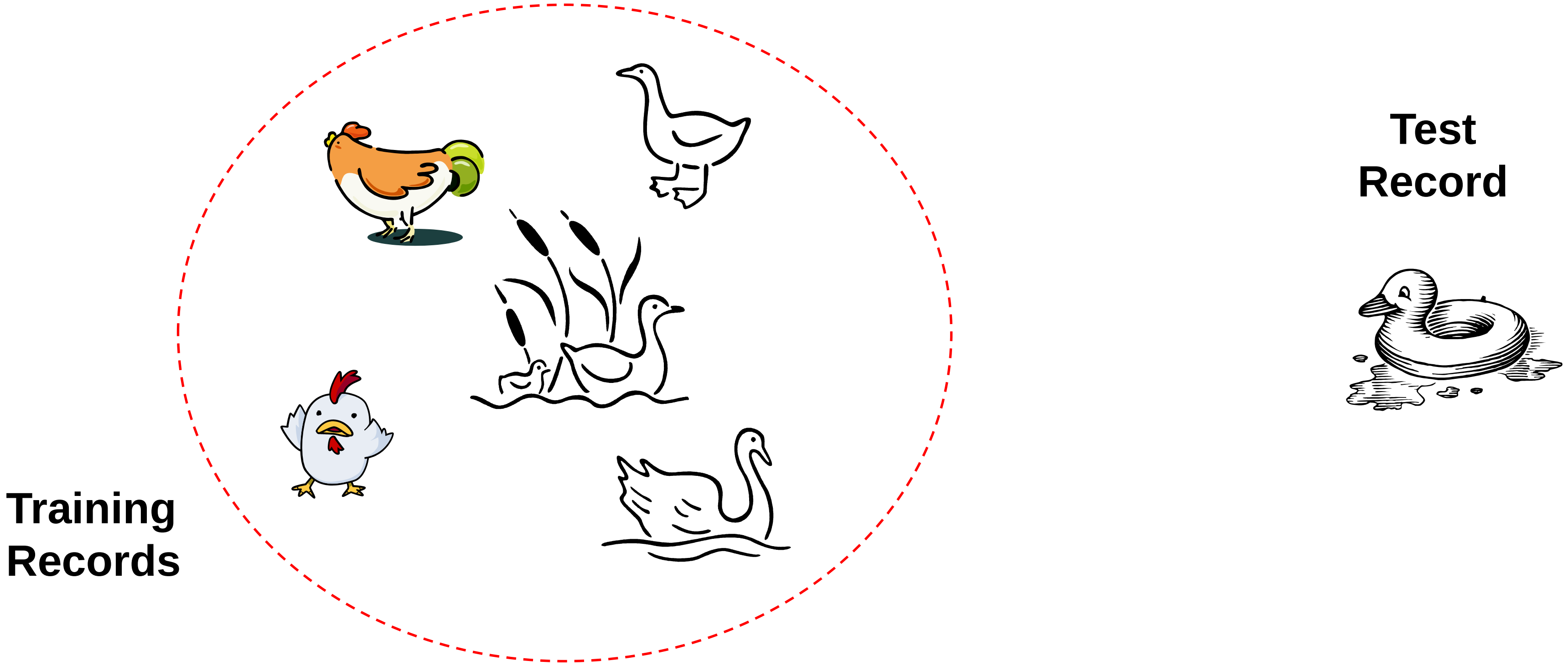
- Relatively inexpensive to construct
- Extremely fast at classifying unknown records
- Easy to interpret for small-sized trees
- Robust to noise (especially when methods to avoid overfitting are employed)
- Can easily handle redundant or irrelevant attributes (unless the attributes are interacting)

Disadvantages: .

- Due to the greedy nature of splitting criterion, interacting attributes (that can distinguish between classes together but not individually) may be passed over in favor of other attributed that are less discriminating.
- Each decision boundary involves only a single attribute

Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it’s probably a duck

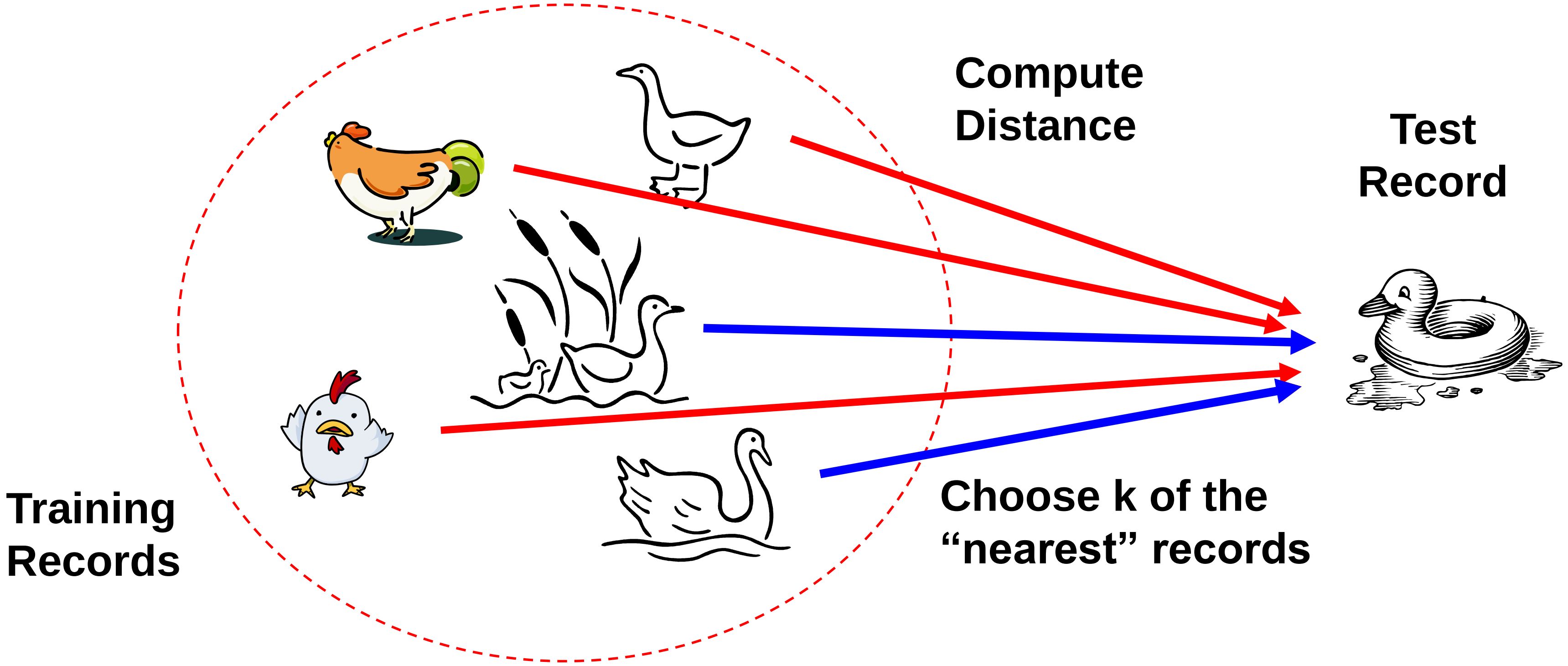


Training Records

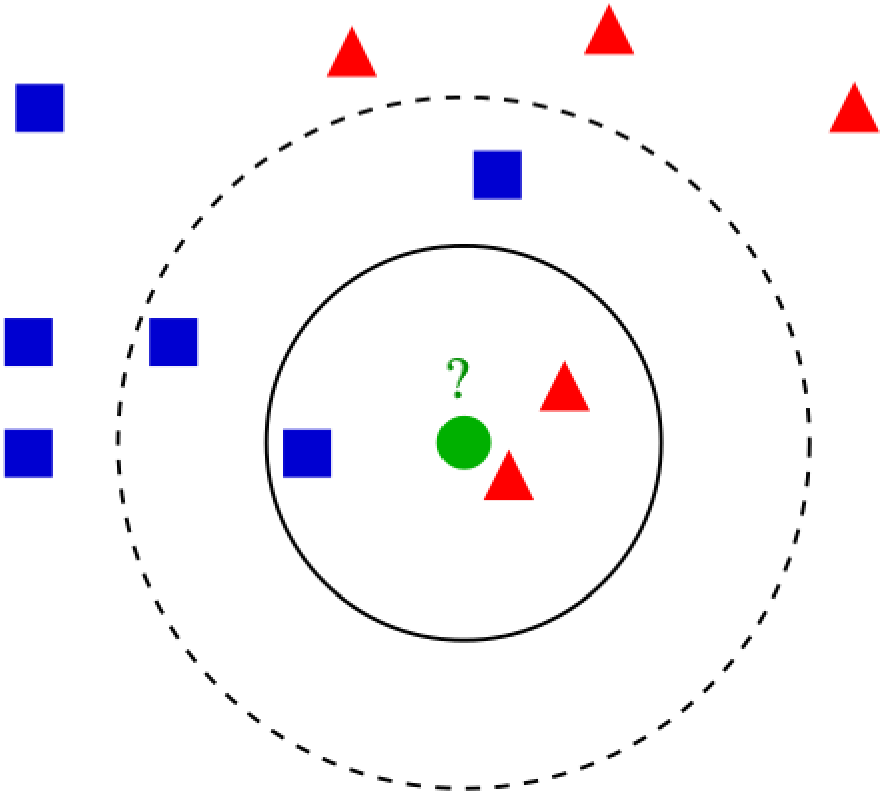
Test Record

Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



k Nearest Neighbor (kNN) Classification

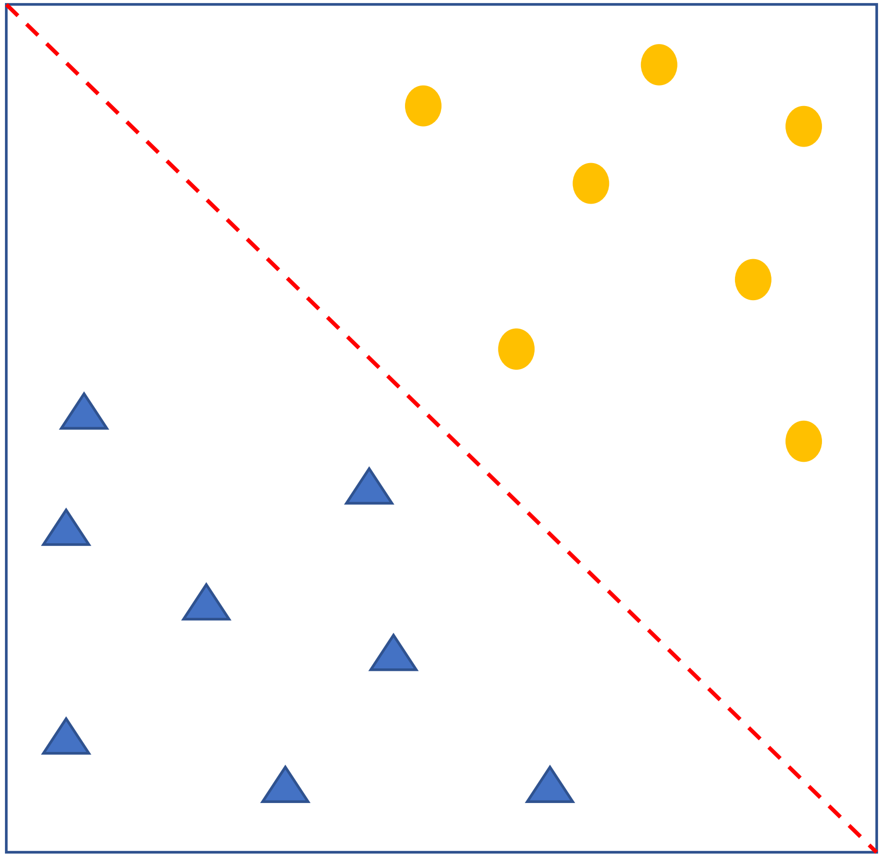


□Requires the following:

- A set of labeled records
- Proximity metric to compute distance/similarity between a pair of records
  - e.g., Euclidean distance
- The value of *k*, the number of nearest neighbors to retrieve
- A method for using class labels of K nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

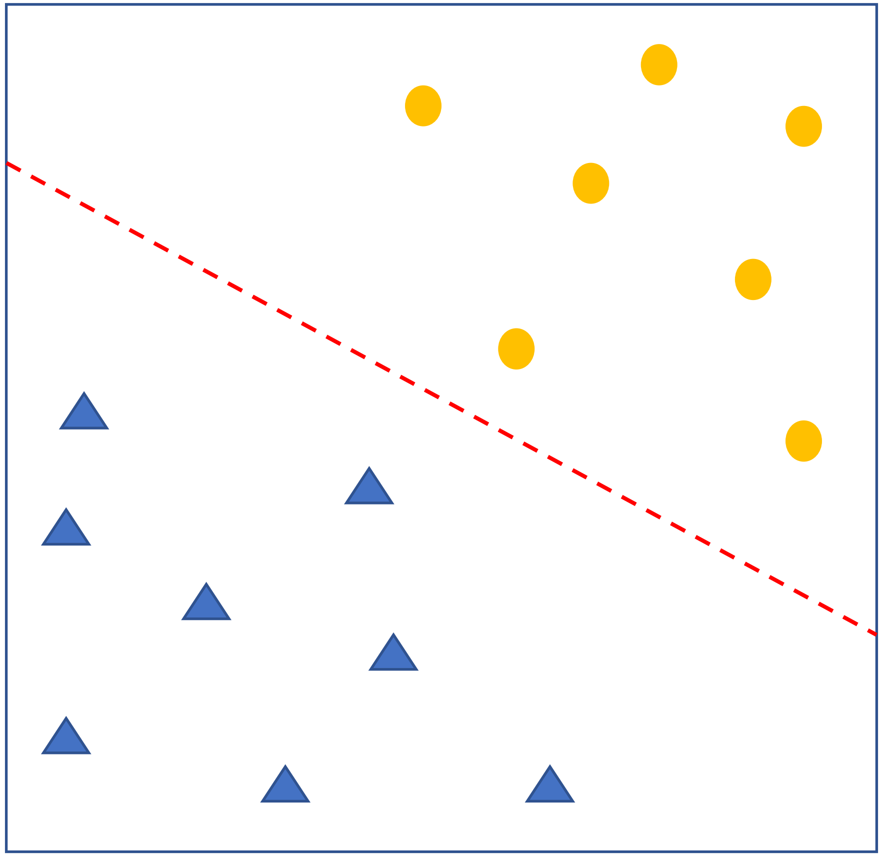
Support Vector Machines

One possible solution



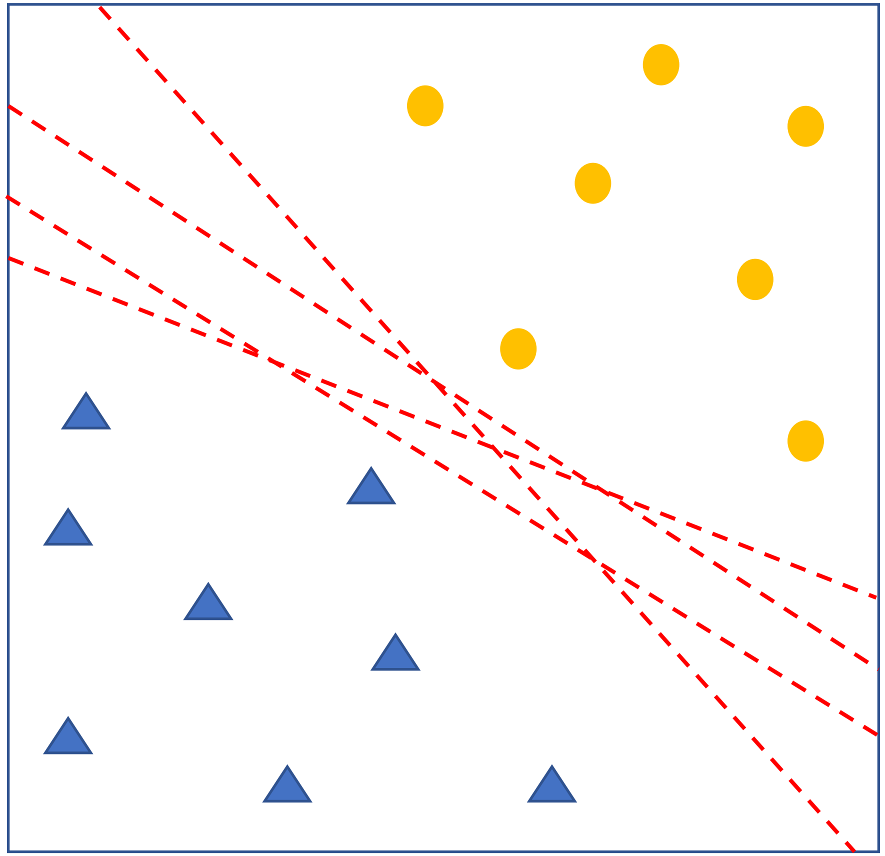
Support Vector Machines

Another possible solution

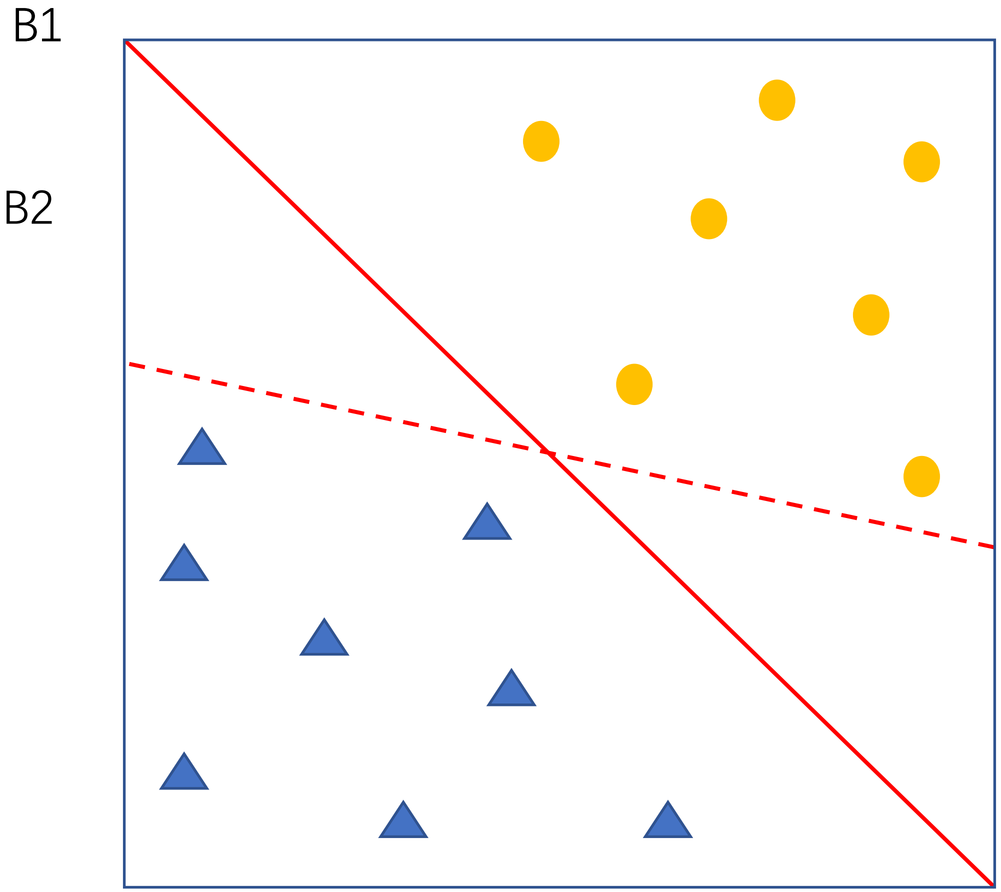


Support Vector Machines

Other possible solutions

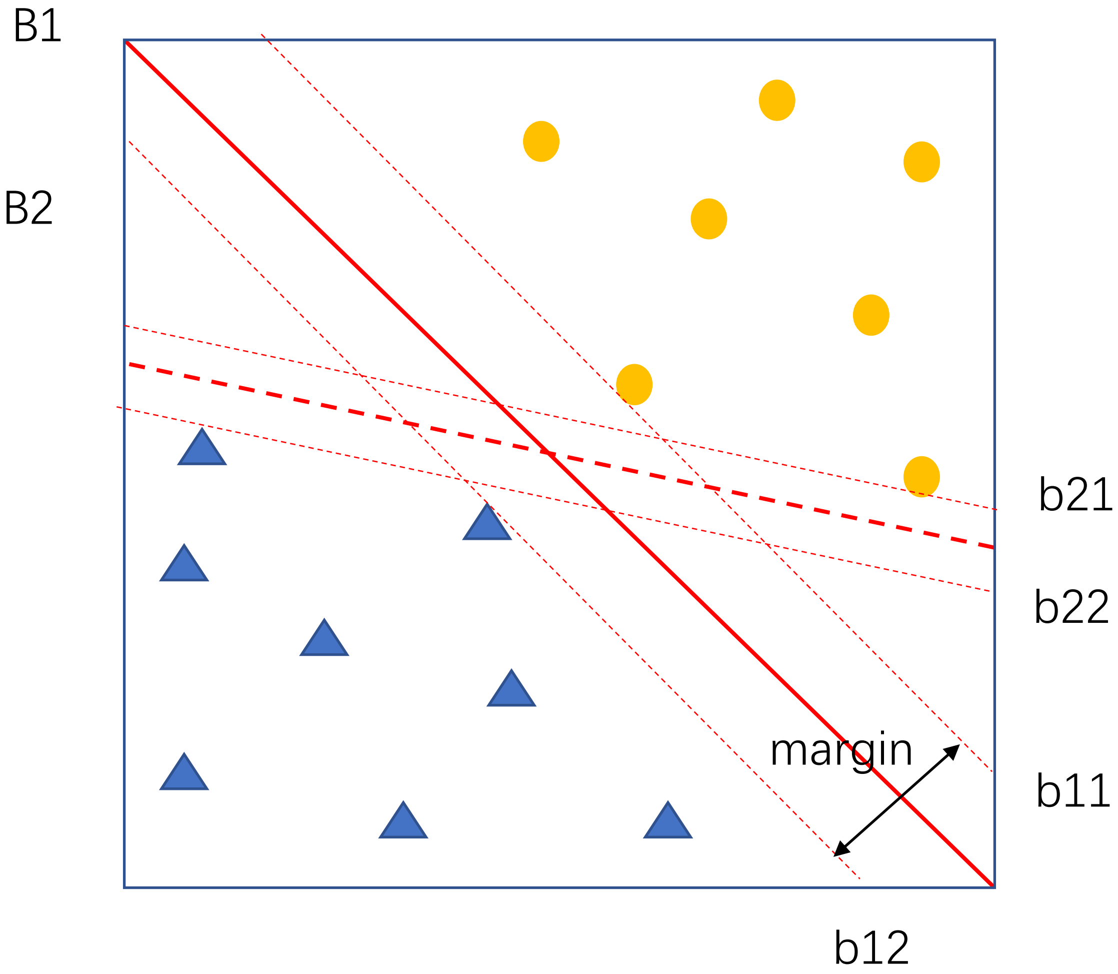


Support Vector Machines



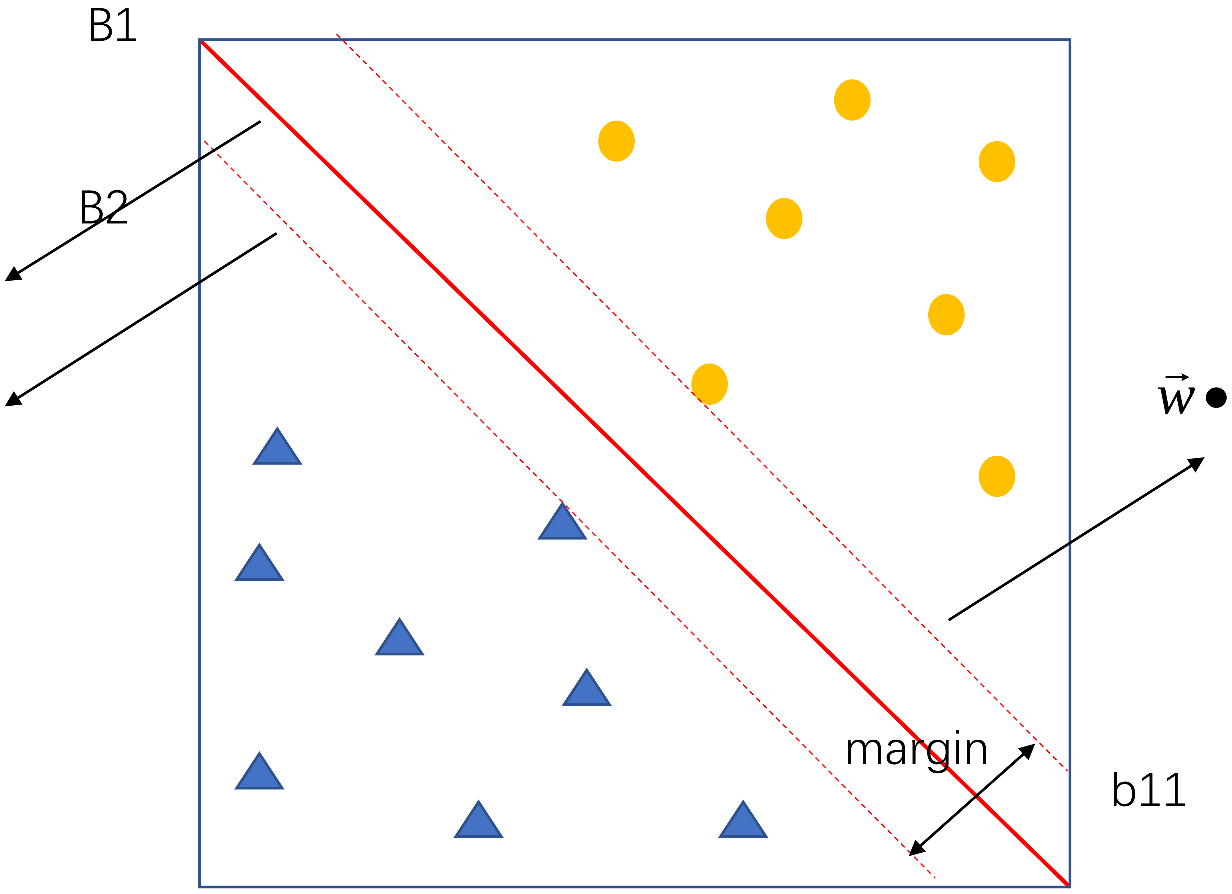
Which one is better? B1 or B2?
How do you define better?

Support Vector Machines



- Find hyperplane maximizes the margin => B1 is better than B2

Support Vector Machines



$\vec{w} \bullet \vec{x} + b = 0$

$\vec{w} \bullet \vec{x} + b = -1$

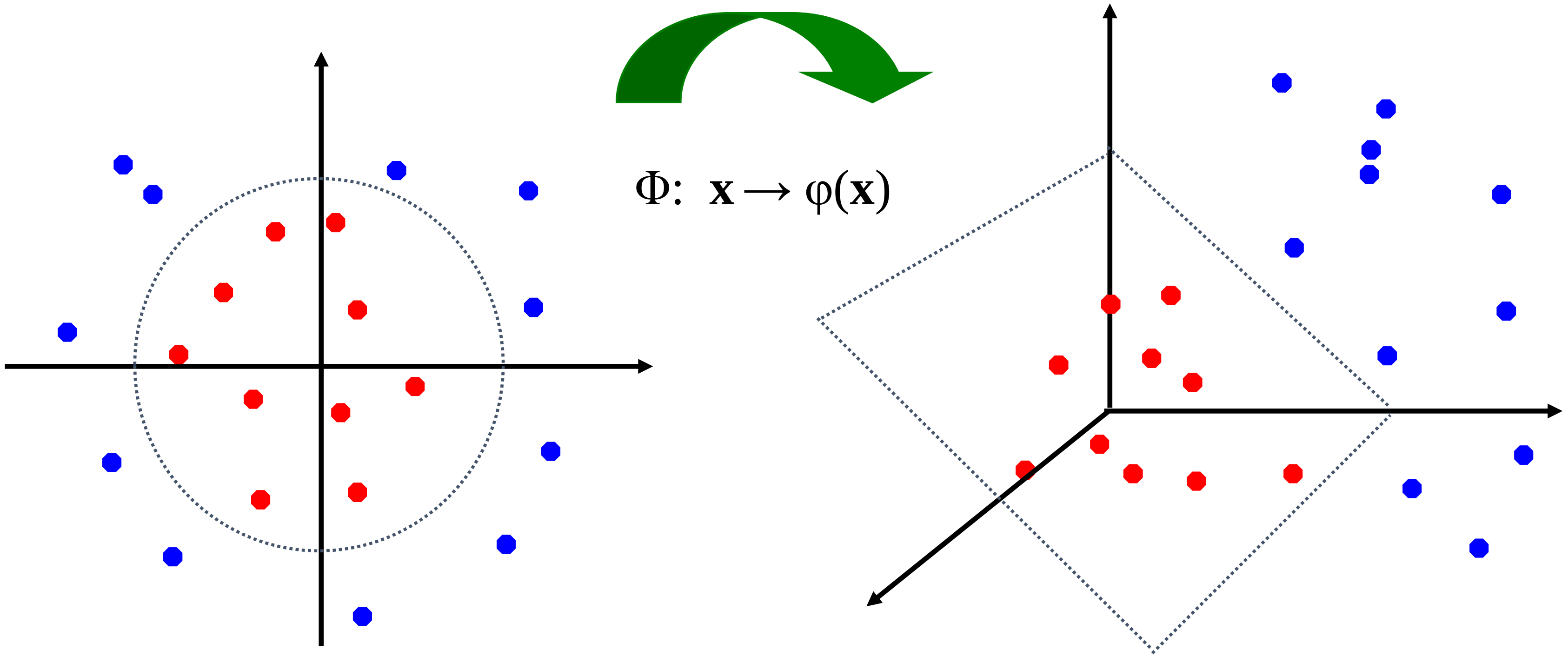
$\vec{w} \bullet \vec{x} + b = +1$

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \leq -1 \end{cases}$$

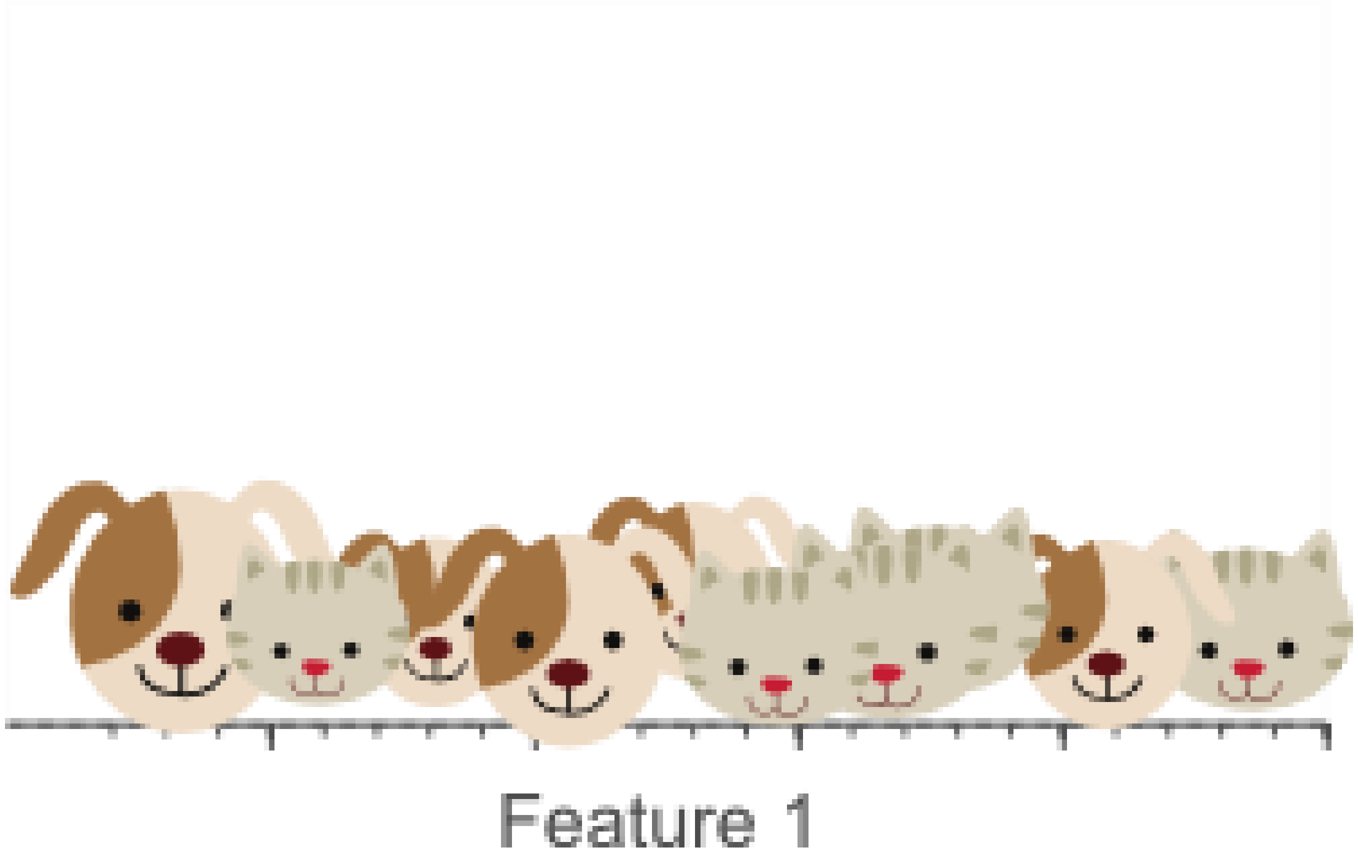
$$\text{Margin} = \frac{2}{\| \vec{w} \|}$$

Non-linear SVM: Feature spaces

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable:

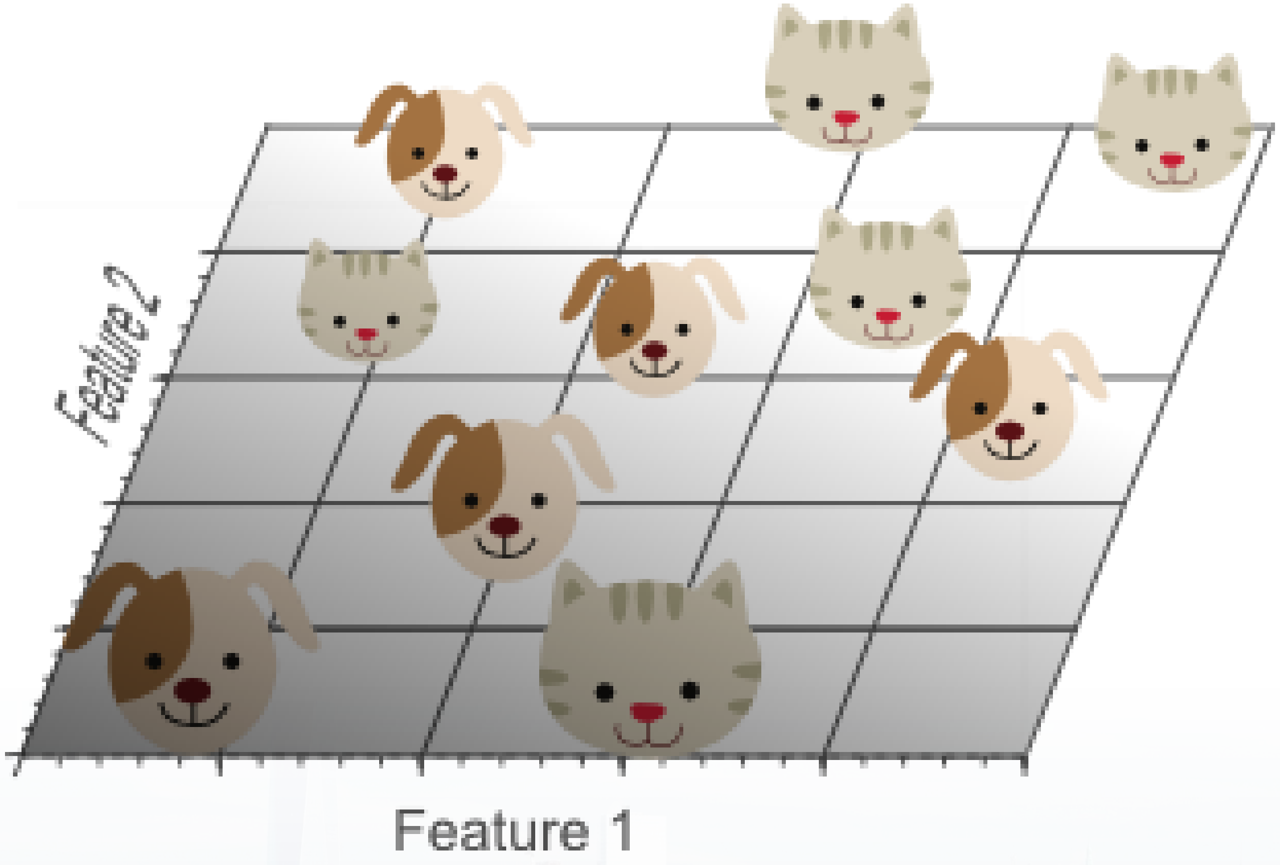


Example



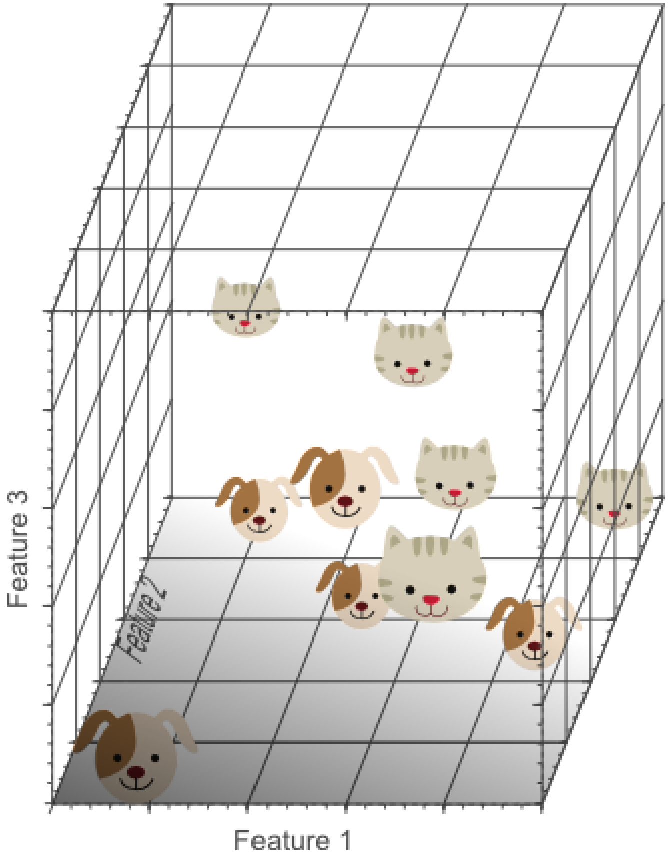
1) Retrieved from https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

Example



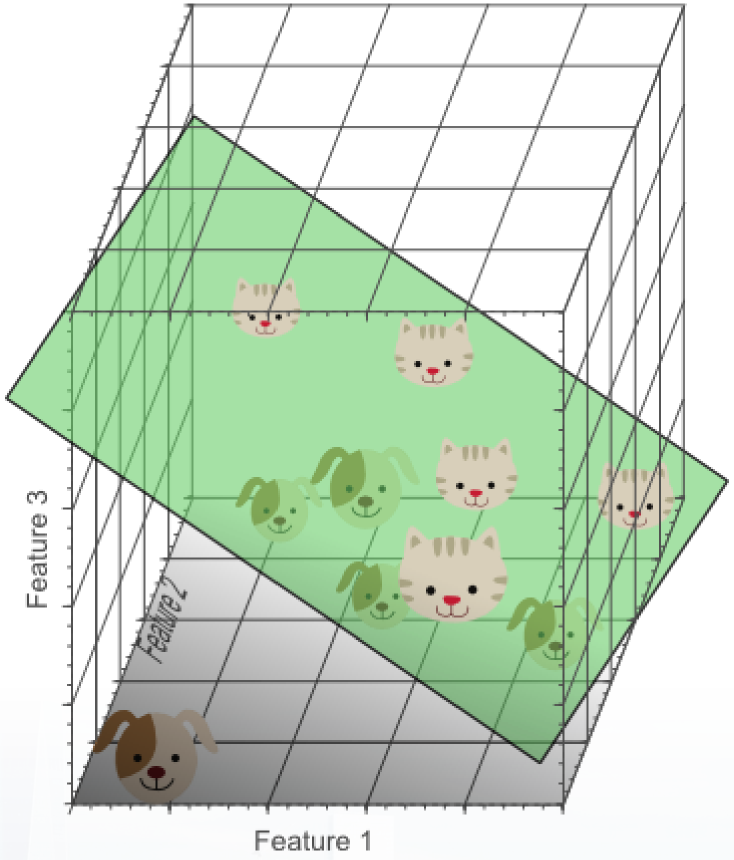
1) Retrieved from https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

Example



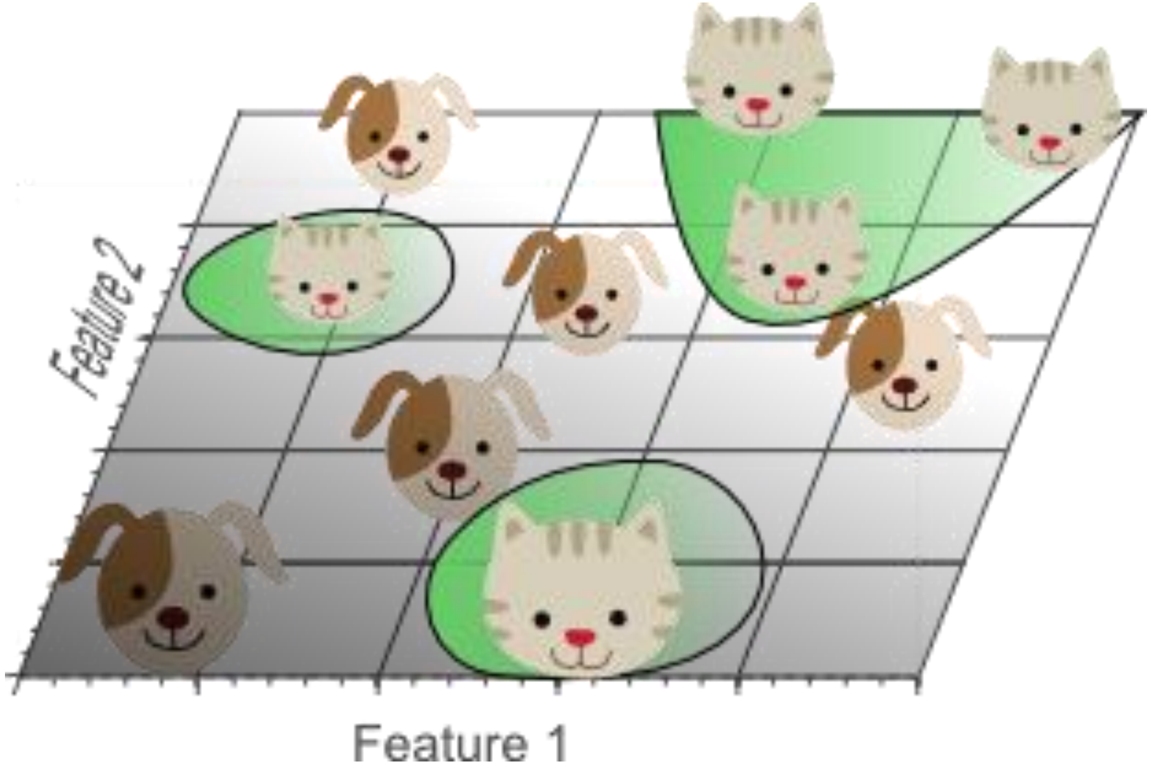
1) Retrieved from https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

Example



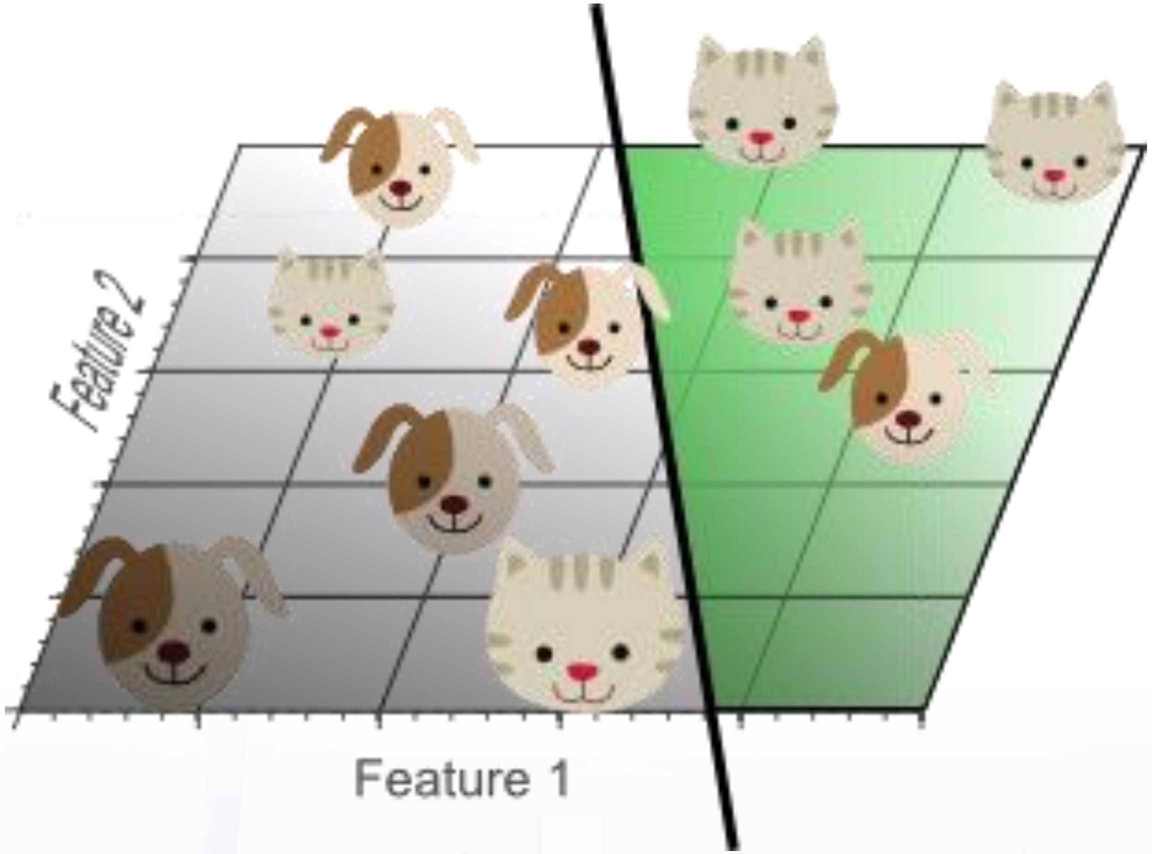
1) Retrieved from https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

Example



1) Retrieved from https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

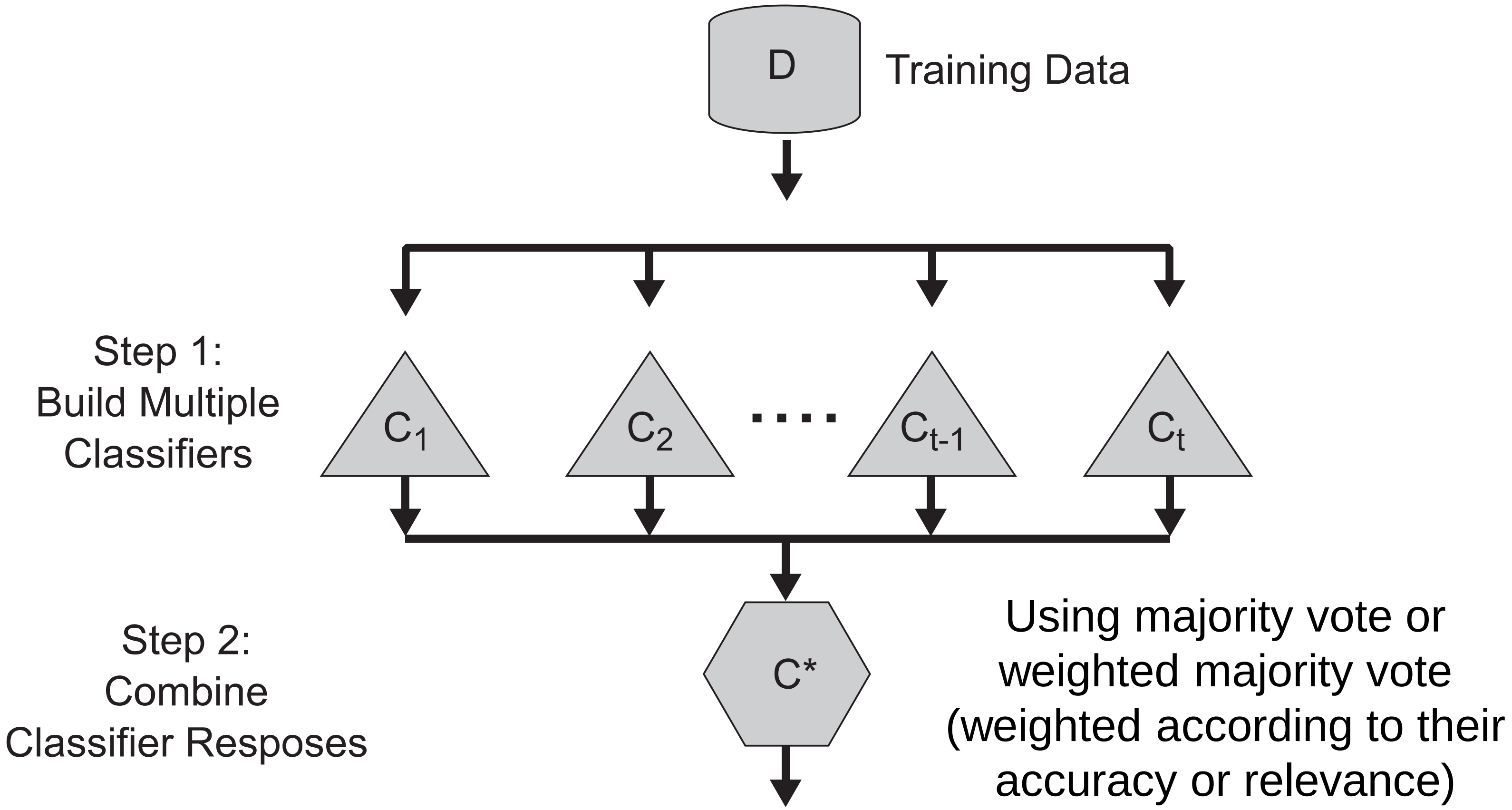
Example



1) Retrieved from https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

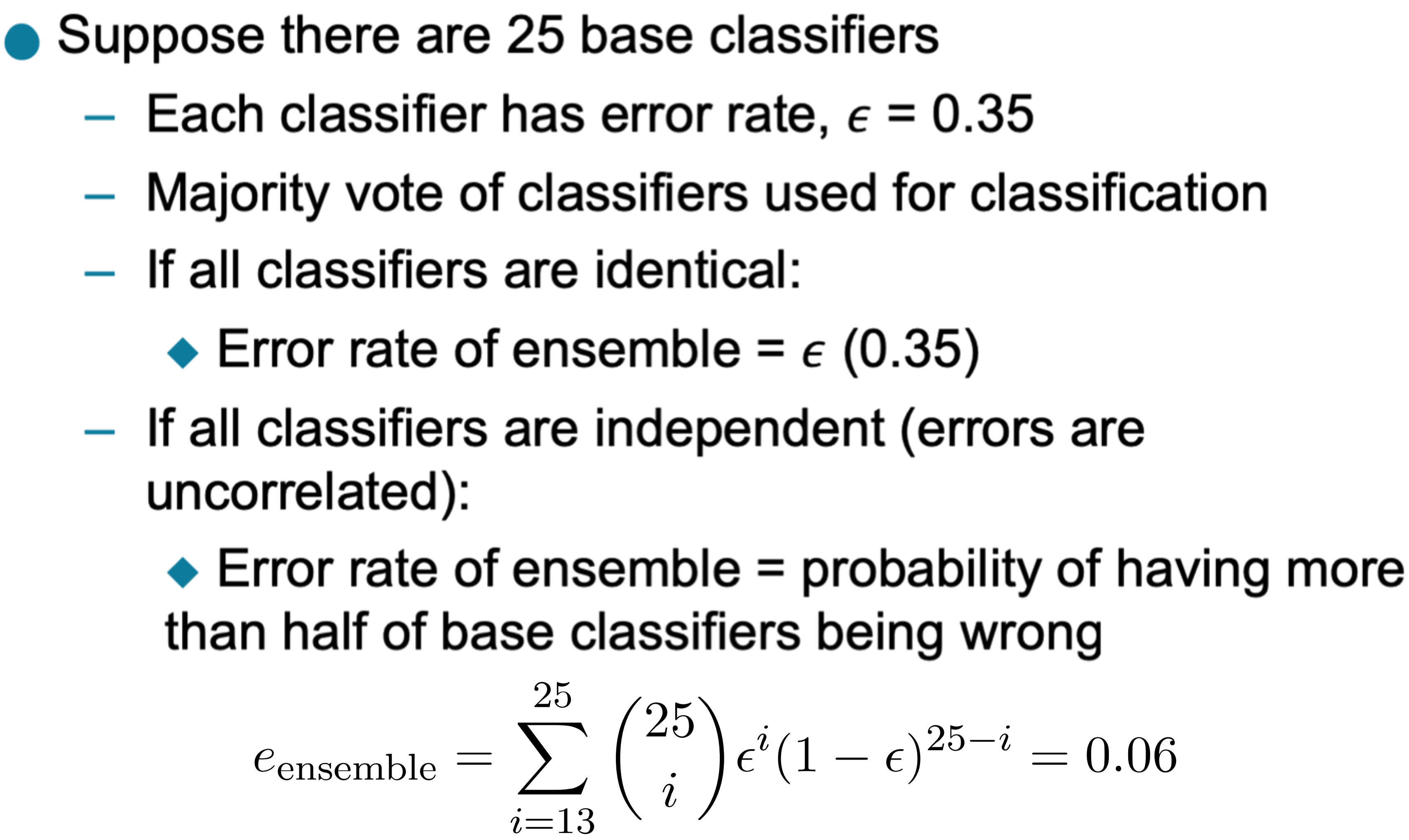
Ensemble Techniques

- Construct a set of base classifiers learned from the training data
- Predict class label of test records by combining the predictions made by multiple classifiers (e.g., by taking majority vote)



Ensemble Techniques

- Why ensemble?



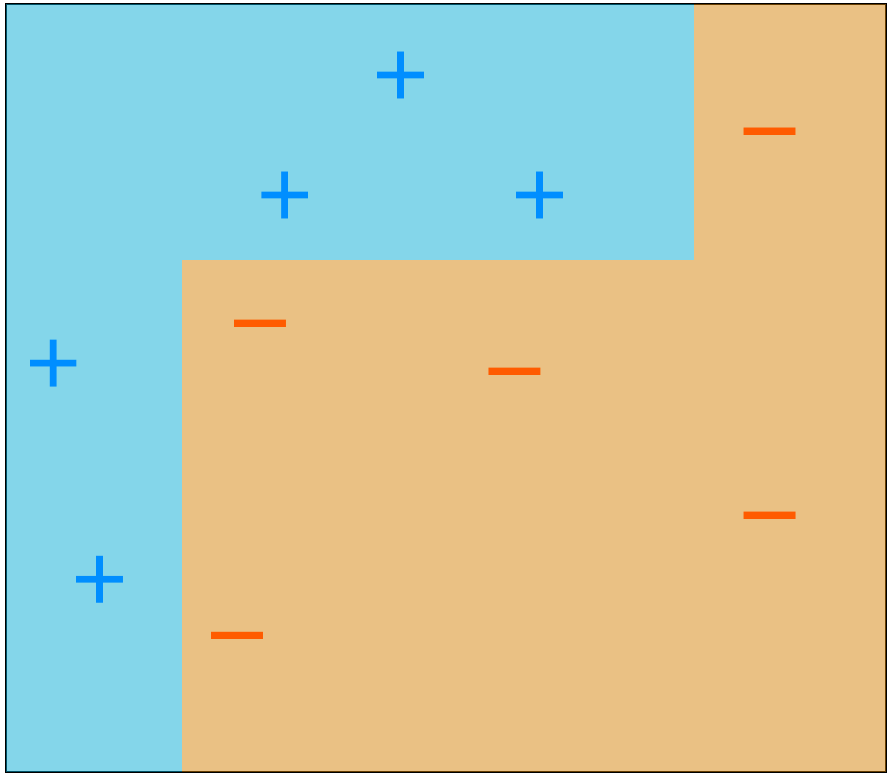
- Suppose there are 25 base classifiers
  - Each classifier has error rate, ϵ = 0.35
  - Majority vote of classifiers used for classification
  - If all classifiers are identical:
    - Error rate of ensemble = ϵ (0.35)
  - If all classifiers are independent (errors are uncorrelated):
    - Error rate of ensemble = probability of having more than half of base classifiers being wrong

$$e_{\text{ensemble}} = \sum_{i=13}^{25} \binom{25}{i} \epsilon^i (1-\epsilon)^{25-i} = 0.06$$

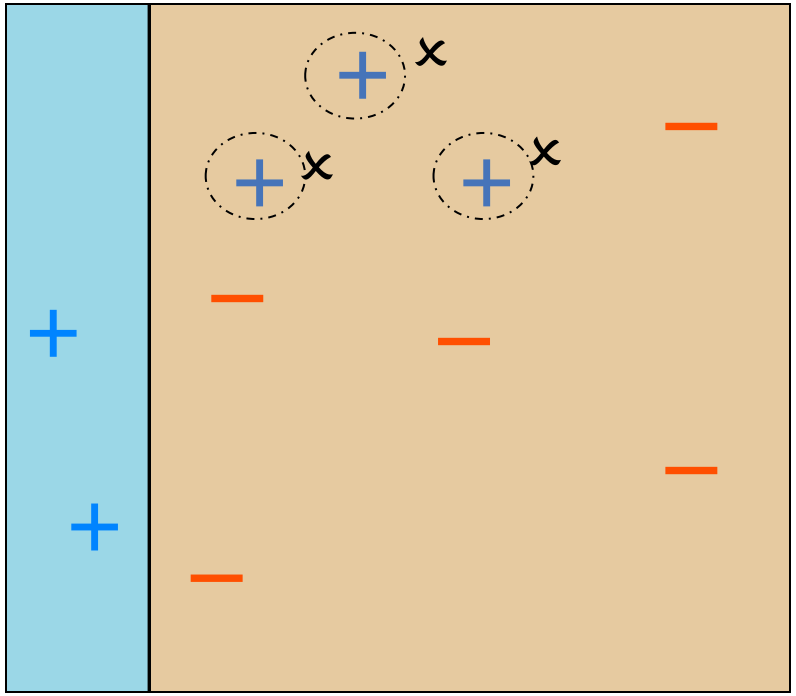
Boosting

- A family of methods:
  - **AdaBoost** (Freund & Schapire, 1996)
- Training
  - Produce a sequence of classifiers (the same base learner)
  - Each classifier is dependent on the previous one, and focuses on the previous one's errors
  - Examples that are incorrectly predicted in previous classifiers are given higher weights
- Testing
  - For a test case, the results of the series of classifiers are combined to determine the final class of the test case.

Example of a Good Classifier

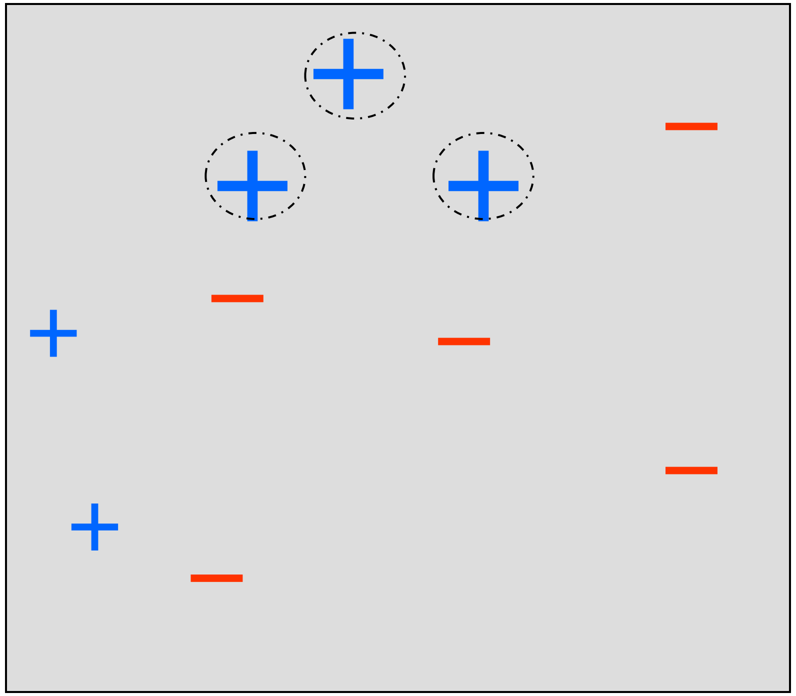


Round 1 of 3



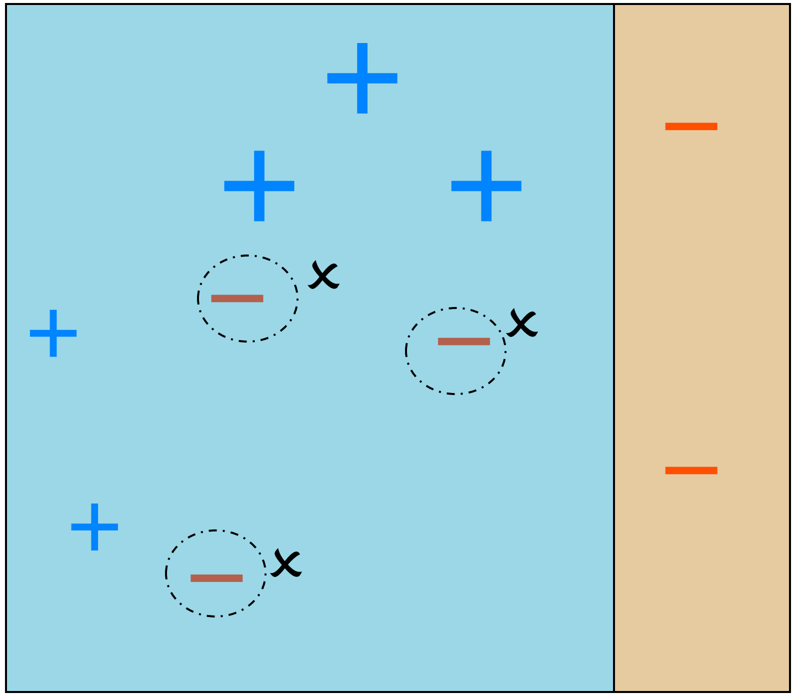
h_1 $\varepsilon_1 = 0.300$

α_1=0.424



D_2

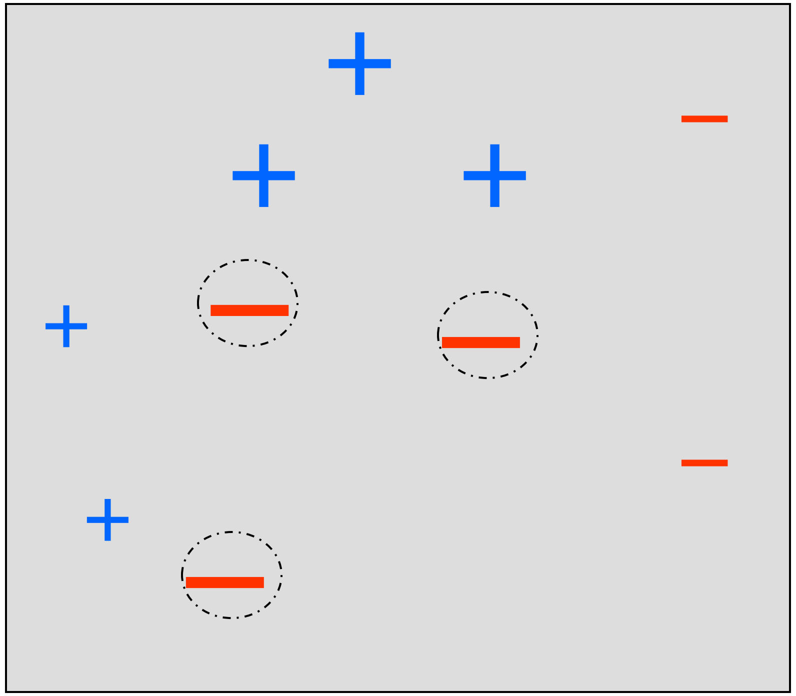
Round 2 of 3



$\varepsilon_2 = 0.196$

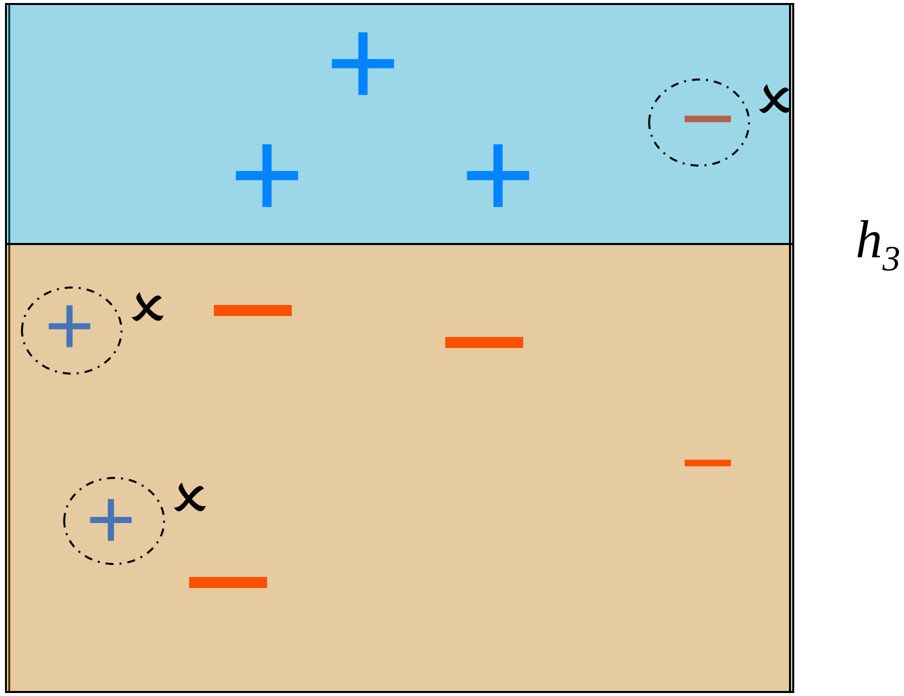
h_2

$\alpha_2 = 0.704$



D_2

Round 3 of 3

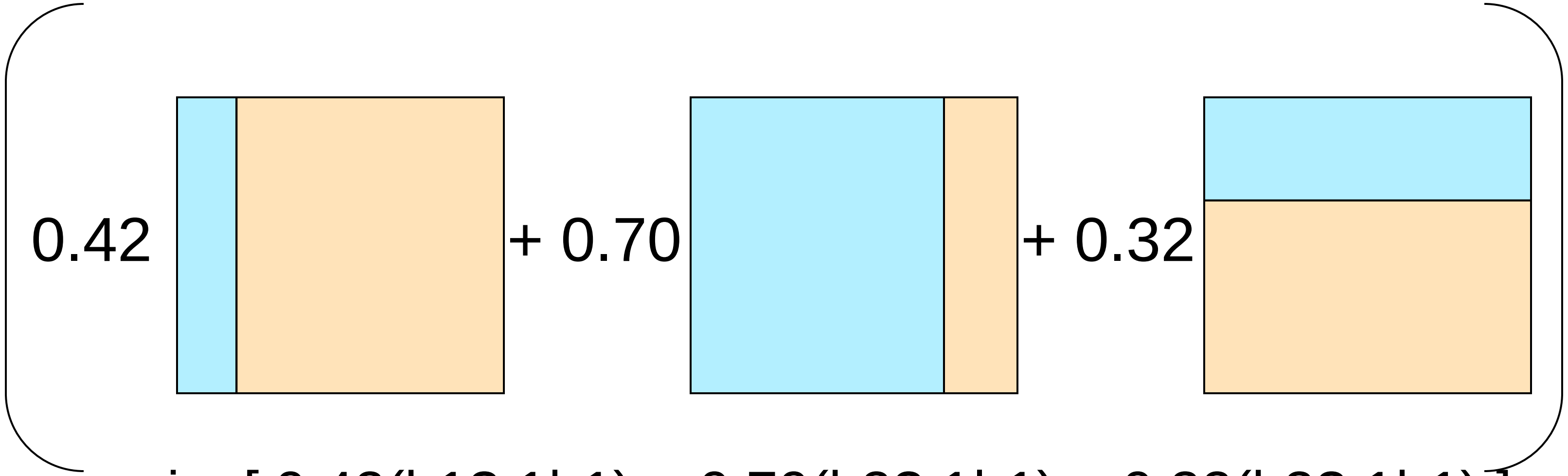


$\varepsilon_3 = 0.344$

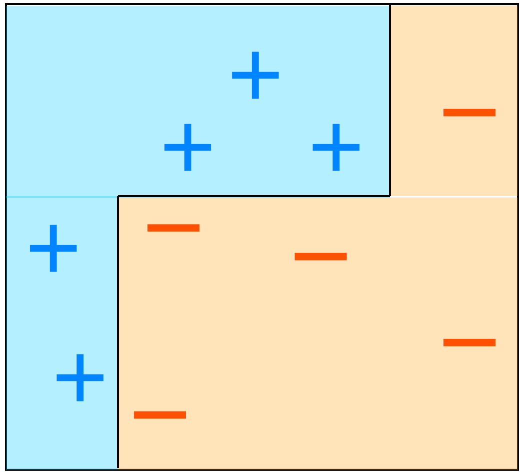
$\alpha_2 = 0.323$

STOP

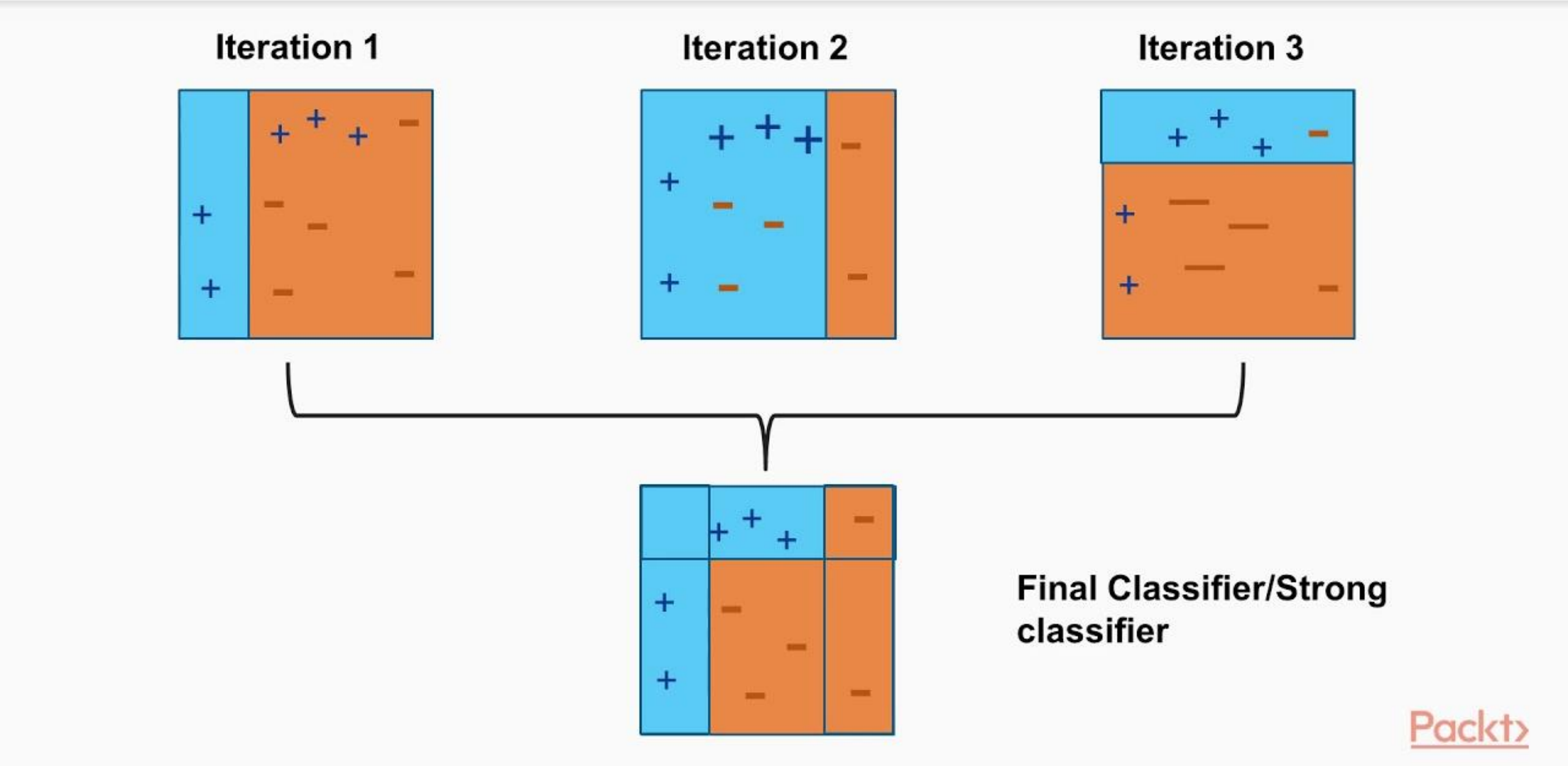
Final Hypothesis



H_{final} = sign[0.42(h1? 1|-1) + 0.70(h2? 1|-1) + 0.32(h3? 1|-1)]



Boosting



https://www.youtube.com/watch?v=BoGNyWW9-mE&ab_channel=PacktVideo

Resource of Machine Learning

- 李宏毅 机器学习2020
  - https://www.youtube.com/watch?v=c9TwBeWAj_U&ab_channel=Hung-yiLee
  - https://www.bilibili.com/video/av94519857/

End of Class 3