Introduction to Machine Learning CS182

Lu Sun

School of Information Science and Technology

ShanghaiTech University

September 26, 2023

Today:

- Course logistics
- Introduction to machine learning
- Overview of machine learning
- Overview of supervised learning I

Readings:

- The Elements of Statistical Learning (ESL), Chapters 1--2
- Pattern Recognition and Machine Learning (PRML), Chapter 1
- Deep Learning (DL), Chapters 1--3

Course Logistics

About Me: SUN Lu（孙 露）

- Assistant Professor in SIST
  - Since Nov., 2019
  - Email: sunlu1@shanghaitech.edu.cn
  - Homepage: https://faculty.sist.shanghaitech.edu.cn/sunlu/

- Teaching
  - CS182 Introduction to Machine Learning
    - 2022 Spring, 2022 Fall, 2023 Spring
  - SI151 Optimization and Machine Learning
    - 2020 Spring, 2021 Spring
  - CS150A Database
    - 2021 Fall, 2022 Fall
  - CS150 Database and Data Mining
    - 2020 Fall

TAs

– CAO Tianxiao (曹 天笑)
caotx@shanghaitech.edu.cn

– JIANG Haoran (蒋 浩然)
jianghr1@shanghaitech.edu.cn

– MAO Zhanwang (茅 展望)
maozhw@shanghaitech.edu.cn

– SHAO Yuanming (邵 元明)
shaoym1@shanghaitech.edu.cn

Week	Date	Lec.	Topic	TA course	HW out	HW in
1	Sept. 26	1	Overview of Supervised Learning			
	Sept. 28		**University Anniversary**			
2	Oct. 3		**National Day**			
	Oct. 7	2	Linear Methods for Regression	TAC1		
3	Oct. 10	3	Linear Methods for Regression			
	Oct. 12	4	Linear Methods for Classification		HW1	
4	Oct. 17	5	Linear Methods for Classification			
	Oct. 19	6	Probability and Estimation	TAC2		
5	Oct. 24	7	Naive Bayes			
	Oct. 26	8	Graphical Models			HW1
6	Oct. 31	9	Graphical Models		HW2	
	Nov. 2	10	Mixture Models and EM	TAC3		
7	Nov. 7	11	Ensemble Learning			
	Nov. 9	12	Ensemble Learning			
8	Nov. 14	13	Kernel Methods			HW2
	Nov. 16	14	Support Vector Machines	TAC4	HW3	
9	Nov. 21	15	Support Vector Machines			
	Nov. 23	16	Semi-Supervised Learning			
10	Nov. 28	17	Active Learning			
	Nov. 30	18	Clustering	TAC5		HW3
11	Dec. 5	19	Dimensionality Reduction		HW4	
	Dec. 7	20	Dimensionality Reduction			
12	Dec. 12	21	Neural Networks			
	Dec. 14	22	Neural Networks	TAC6		
13	Dec. 19	23	Supervised Deep Learning			HW4
	Dec. 21	24	Supervised Deep Learning			
14	Dec. 26	25	Unsupervised Deep Learning		HW5	
	Dec. 28	26	Unsupervised Deep Learning	TAC7		
15	Jan. 2	27	Nonparametric Methods			
	Jan. 4	28	Model Assessment and Selection			
16	Jan. 9	29	Project Presentation			HW5
	Jan. 11	30	Project Presentation	TAC8		

Introduction to Machine Learning CS182

General information

- Time: Tue. & Thu., 13:00-14:40
- Online: Blackboard, Piazza & Gradescope
- 16 weeks (64 credit hours)

All class communication via Piazza

- https://piazza.com/shanghaitech.edu.cn/fall2023/cs182
- announcements and discussion
- read it regularly
- post all questions/comments there
- direct email is not a good idea

Introduction to Machine Learning CS182

Grading

- Homework: 30%
- Course project: 30%
- Final exam: 40%

Highlights

- Please write your HW, project and exam in English
- Submitted to GradeScope: https://www.gradescope.com/courses/632125
  Entry Code: **8EXX4K**
- For late HW or project, the score will be exponentially decreased
- Once any plagiarism or cheating is confirmed, relevant assignments or exams will receive 0 points

Introduction to Machine Learning CS182

Recommended textbooks

- **The Elements of Statistical Learning: Data Mining, Inference and Prediction**, Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman
- **Pattern Recognition and Machine Learning**, Christopher Bishop
- **Machine Learning**, Tom M. Mitchell
- **Introduction to Machine Learning**, Ethem Alpaydin
- **Deep Learning**, Ian Goodfellow and Yoshua Bengio and Aaron Courville
- **Convex Optimization**, Stephen Boyd and Lieven Vandenberghe

Some useful online resources

- CMU, machine learning course

http://www.cs.cmu.edu/~ninamf/courses/601sp15/lectures.shtml

- Stanford, convex optimization course

https://web.stanford.edu/~boyd/cvxbook/

Introduction to Machine Learning

Machine Learning

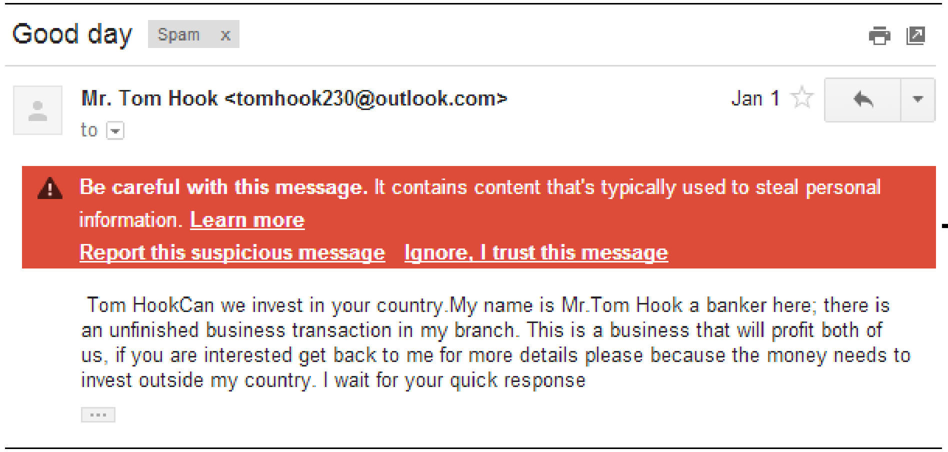
"Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead."

------Wikipedia

ML: Study of algorithms that

- improve their performance P
- at some task T
- with experience E

Well-defined ML task: <P, T, E>



spam vs email

Learning to Detect Spam Emails

- Data:
    - 4601 email messages
    - Each is labeled by email (+) or spam (-)
    - The relative frequencies of the 57 most commonly occurring words and punctuation marks in the message
- Classify:
    - label future messages email (+) or spam (-)
- Supervised learning problem on categorical data:

Binary classification problem

Table: Words with largest difference between spam and email shown.

	spam	email
george	0.00	1.27
you	2.26	1.27
your	1.38	0.44
hp	0.02	0.90
free	0.52	0.07
hpl	0.01	0.43
!	0.51	0.11
our	0.51	0.18
re	0.13	0.42
edu	0.01	0.29
remove	0.28	0.01

Learning to Detect Spam Emails

- Examples of rules for prediction:
  - If (%george<0.6) and (%you>1.5)
    then `spam`
    else `email`
  - If (0.2 %you-0.3 %george)>0
    then `spam`
    else `email`
- Tolerance to errors:
  - Tolerant to letting through some spam (false positive)
  - No tolerance towards throwing out email (false negative)

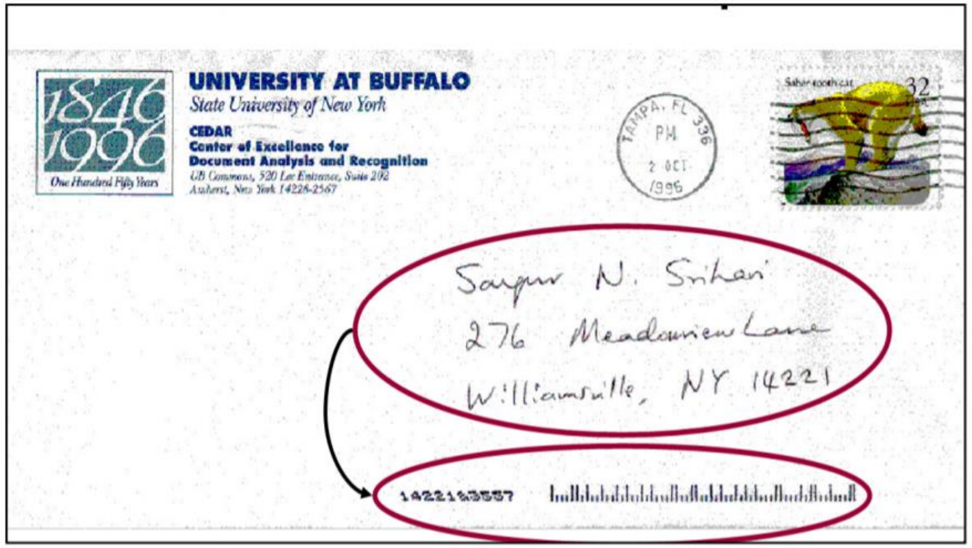
Table: Words with largest difference between spam and email shown.

	spam	email
george	0.00	1.27
you	2.26	1.27
your	1.38	0.44
hp	0.02	0.90
free	0.52	0.07
hpl	0.01	0.43
!	0.51	0.11
our	0.51	0.18
re	0.13	0.42
edu	0.01	0.29
remove	0.28	0.01

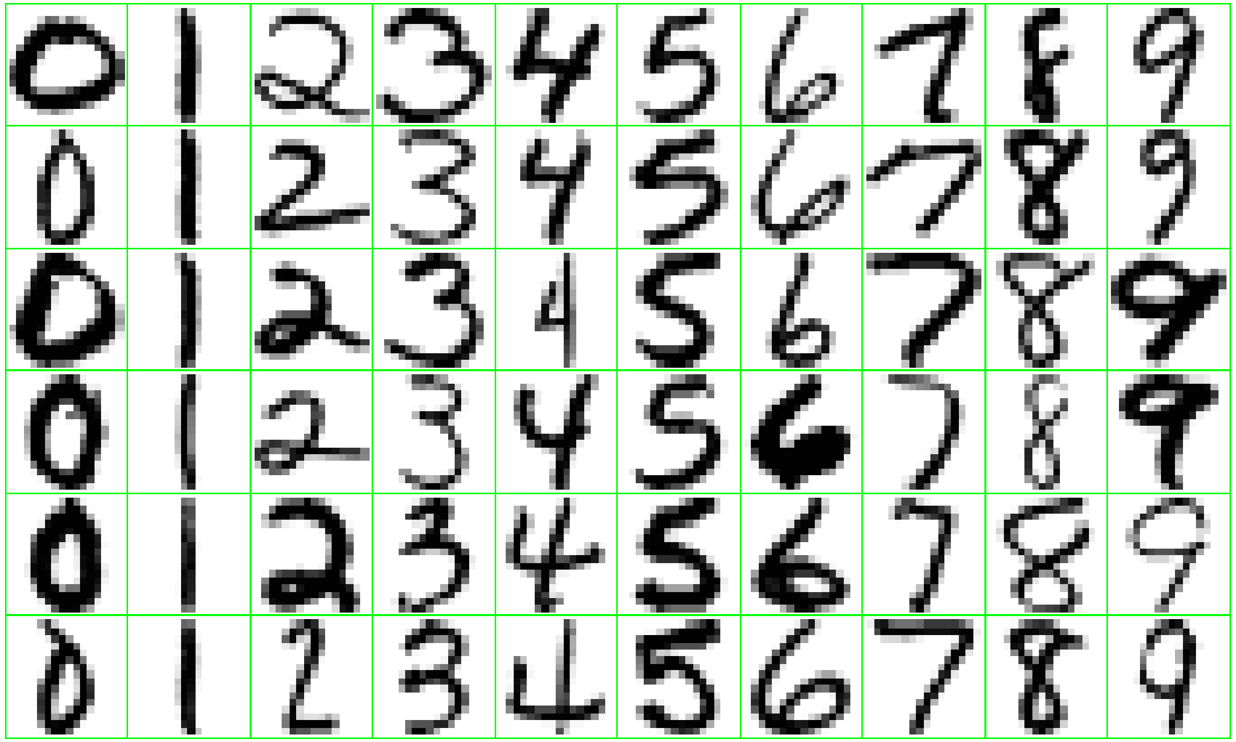
Learning to Recognize Handwritten Digits

Data: images are single digits 16x16 8-bit gray-scale, normalized for size and orientation

Classify: newly written digits



https://cedar.buffalo.edu/~srihari/CSE574/Chap1/1.1%20ML-Overview.pdf



- Non-binary classification problem
- Low tolerance to misclassifications

Learning to Diagnose Prostate Cancer

- Data (by Stamey et al. 1989):
  - Given:

`lcavol`	log cancer volume
`lweight`	log prostate weight
`age`	age
`lbph`	log benign hyperplasia amount
`svi`	seminal vesicle invasion
`lcp`	log capsular penetration
`gleason`	gleason score
`pgg45`	percent gleason scores 4 or 5

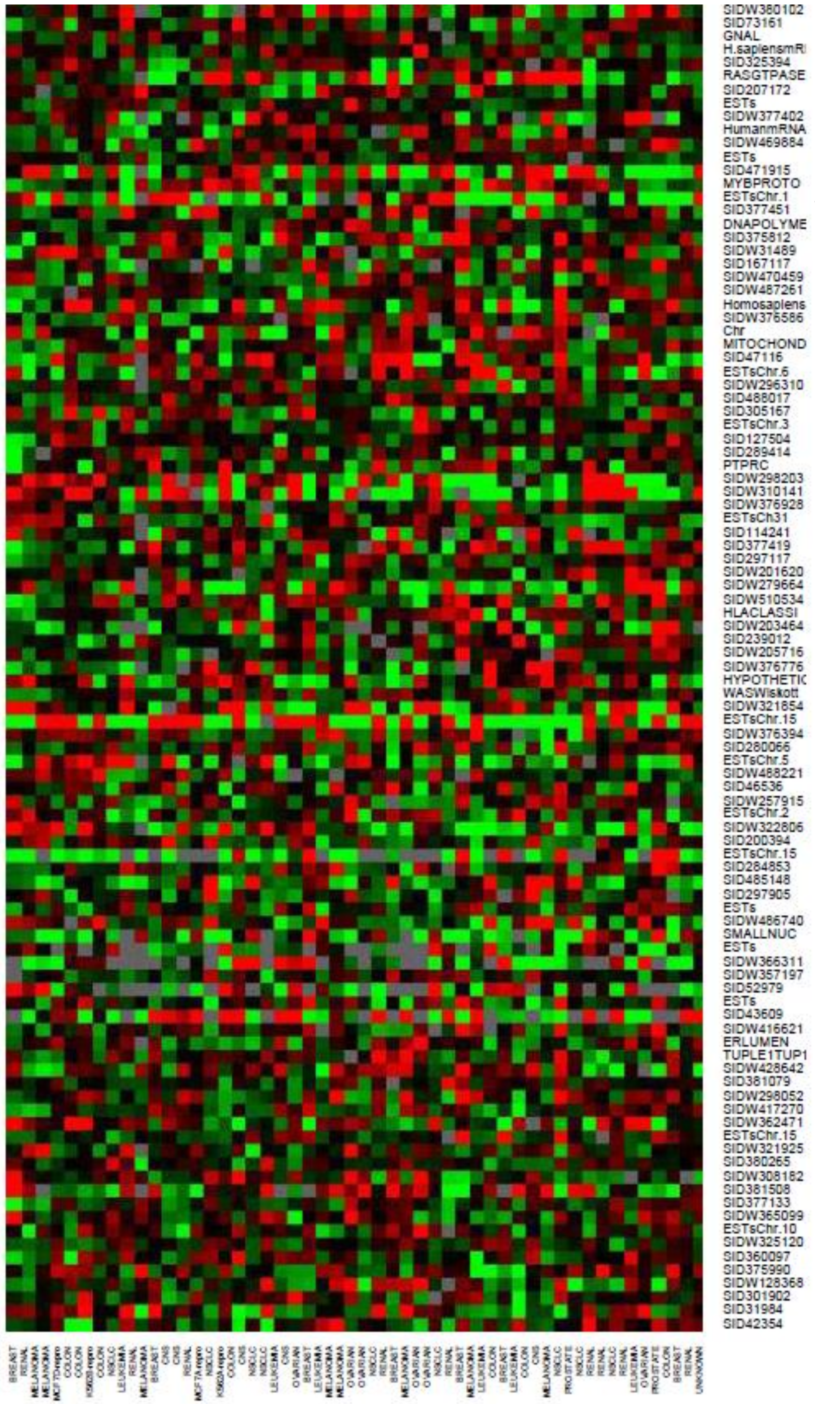
  - Predict:

`lpsa`	log of prostate specific antigen

- Supervised learning problem on quantitative data: Regression problem.

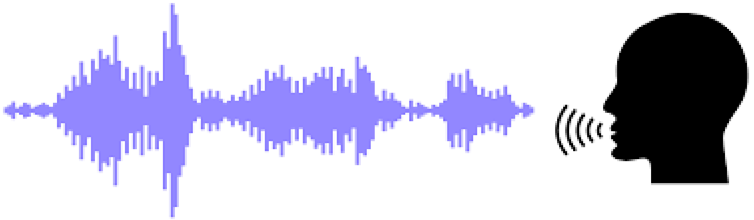
Learning to Analyze DNA Data

- Data:
  - Color intensities signifying the abundance levels of mRNA for a number of genes (6830) in several (64) different cell states (samples).
  - Red: over-expressed gene
  - Green: under-expressed gene
  - Gray: gene with missing values
  - Black: normally expressed gene (according to some predefined background)
- Questions:
  1. Which genes show similar expression over the samples – Unsupervised learning
  2. Which samples show similar expression over the genes – Unsupervised learning
  3. Which genes are highly over or under expressed in certain cancers – Supervised learning



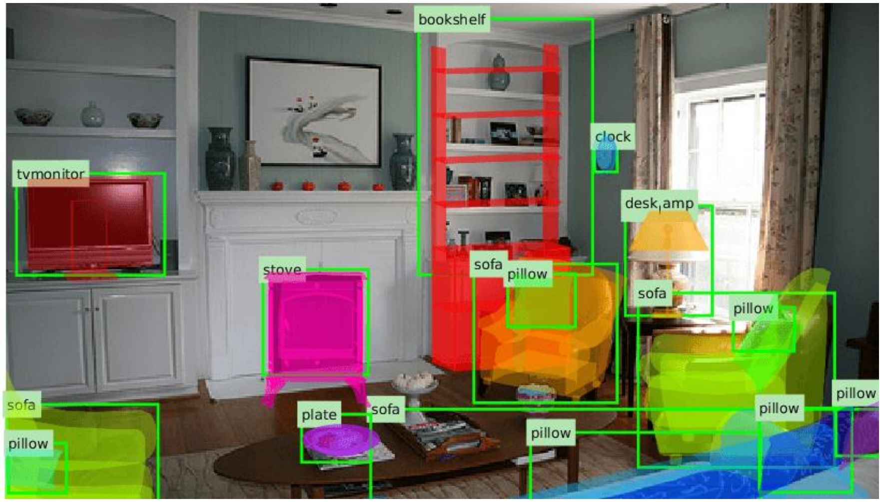
Machine Learning – Practice

Text analysis

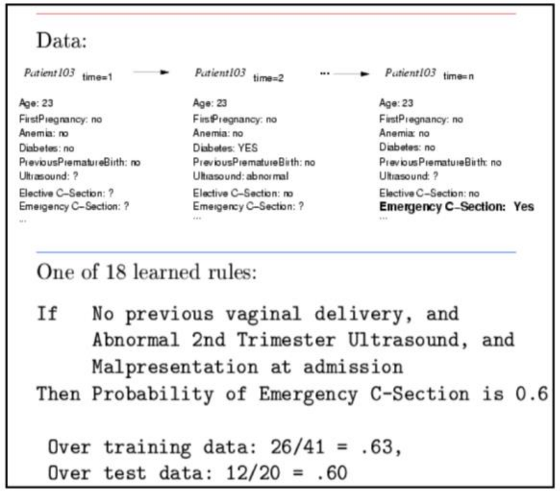


Speech recognition

Control learning



Object recognition



Data:

Patient103 time=1 → Patient103 time=2 ... → Patient103 time=n

Patient103 time=1	Patient103 time=2	Patient103 time=n
Age: 23	Age: 23	Age: 23
FirstPregnancy: no	FirstPregnancy: no	FirstPregnancy: no
Anemia: no	Anemia: no	Anemia: no
Diabetes: no	Diabetes: YES	Diabetes: no
PreviousPrematureBirth: no	PreviousPrematureBirth: no	PreviousPrematureBirth: no
Ultrasound: ?	Ultrasound: abnormal	Ultrasound: ?
Elective C–Section: ?	Elective C–Section: no	Elective C–Section: no
Emergency C–Section: ?	Emergency C–Section: ?	**Emergency C–Section: Yes**
...	...	...

One of 18 learned rules:

```
If   No previous vaginal delivery, and
     Abnormal 2nd Trimester Ultrasound, and
     Malpresentation at admission
Then Probability of Emergency C-Section is 0.6

 Over training data: 26/41 = .63,
 Over test data: 12/20 = .60
```

Mining databases

- Logistic regression
- SVM
- Neural networks
- Hidden Markov models
- Reinforcement learning
- Bayesian methods
- ……

Machine Learning – Theory

PAC Learning Theory

(by Leslie Valiant, 1984)

examples (m)

hypothesis
complexity (H)

failure
probability (δ)

error rate (ε)

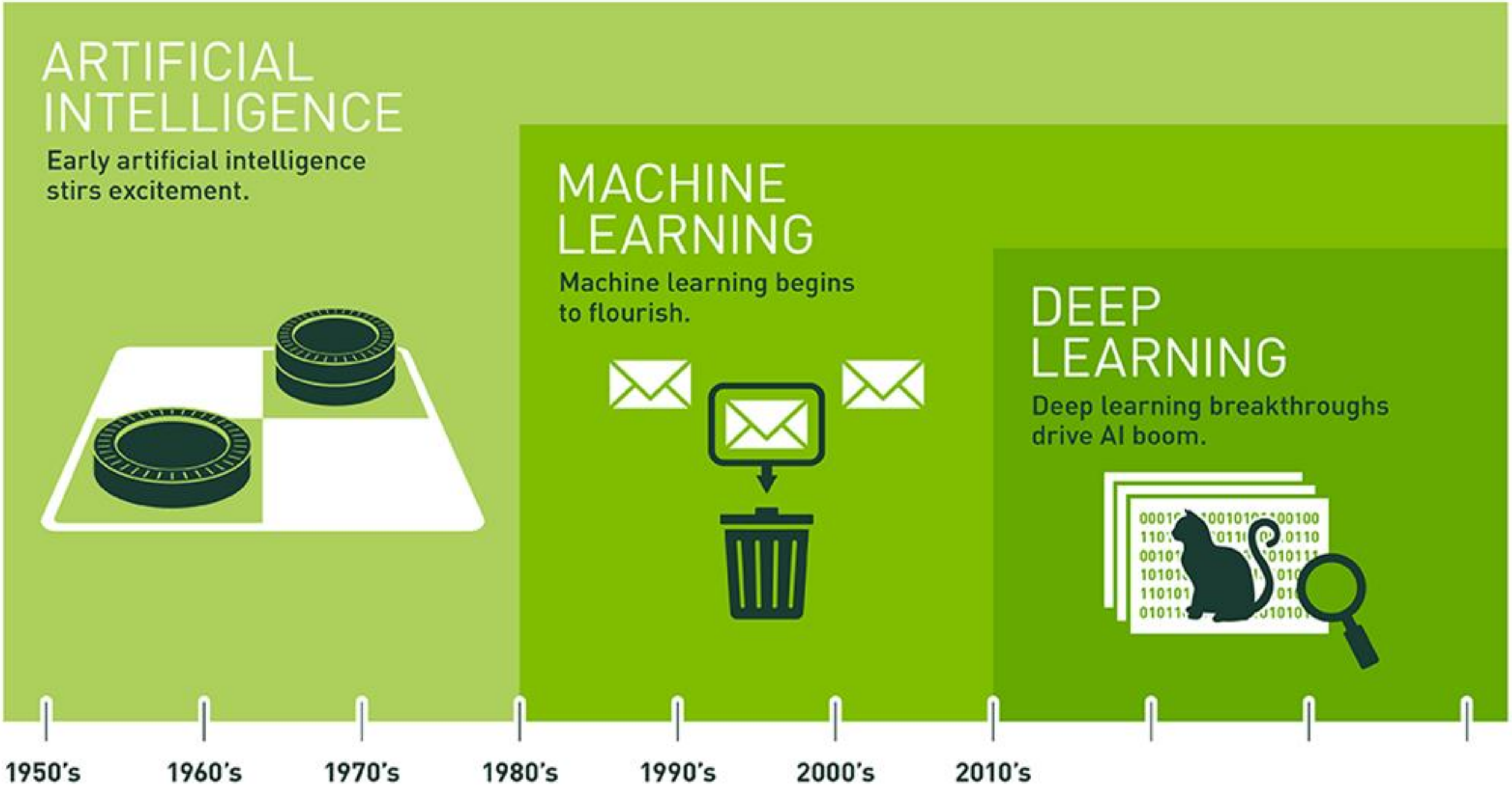
$$m \geq \frac{1}{\varepsilon}\left(ln|H| + \ln(\frac{1}{\delta})\right)$$

PAC: Probably Approximately Correct

Other theories for

- Reinforcement learning
- Semi-supervised learning
- ……

Defining Artificial Intelligence



What You Will Learn in This Course

- The primary machine learning and optimization algorithms
  - Ridge regression, lasso, logistic regression, SVM, neural networks, graphical models, unsupervised learning, deep learning, reinforcement learning…
  - Convex optimization, gradient methods, proximal methods, ADMM, …

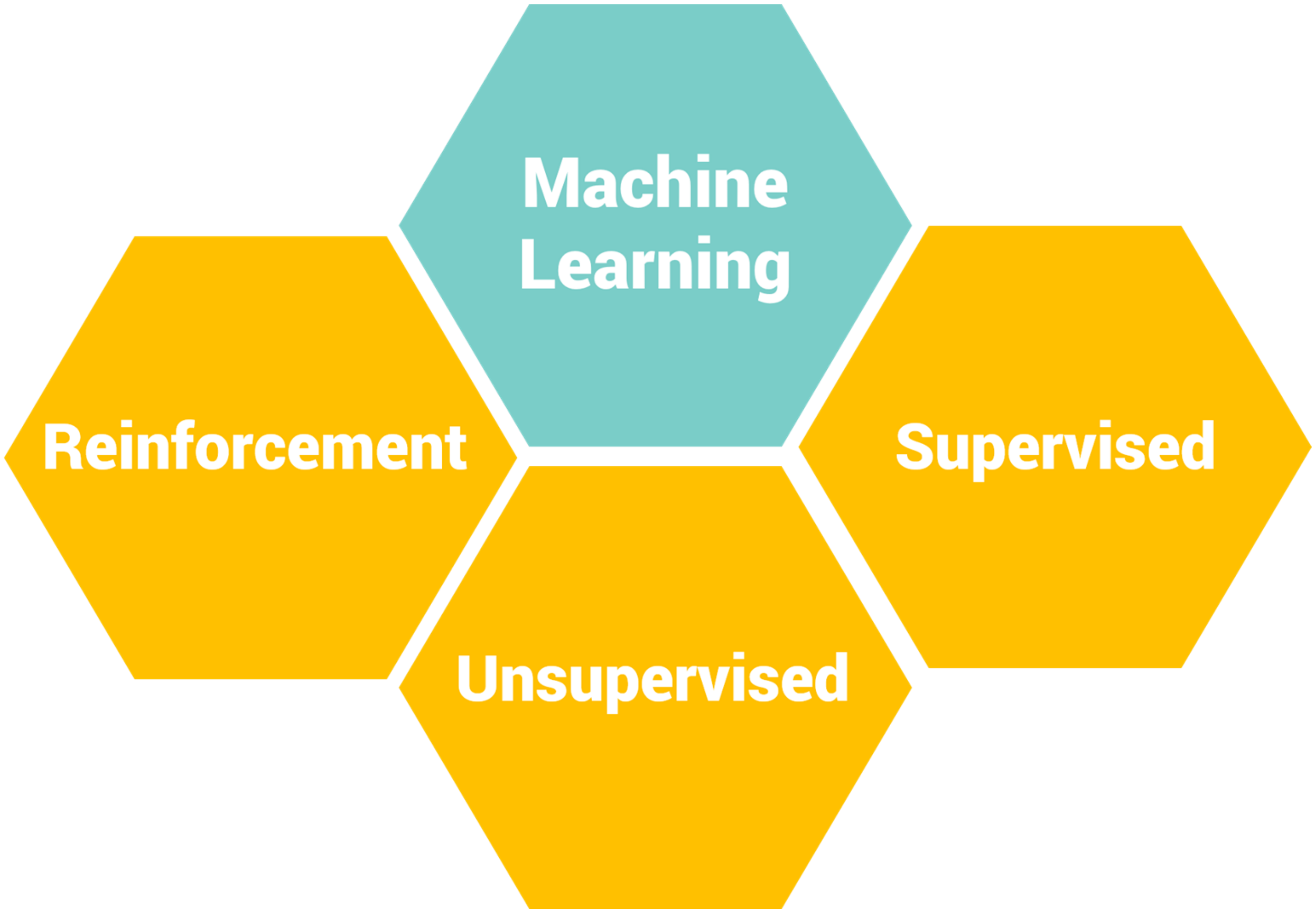
- Underlying statistical and computational theory

- Enable to apply the algorithms to solve practical problems

- Enough to read and understand related research papers.

Overview of Machine Learning

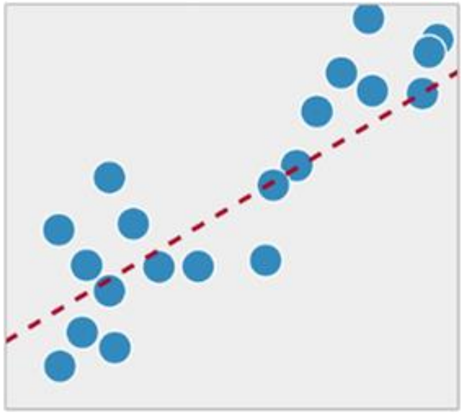
Different Classes of Machine Learning Problems



Supervised Learning

Train your model to map the input to the prediction output based on the **ground truth** labels in the training data

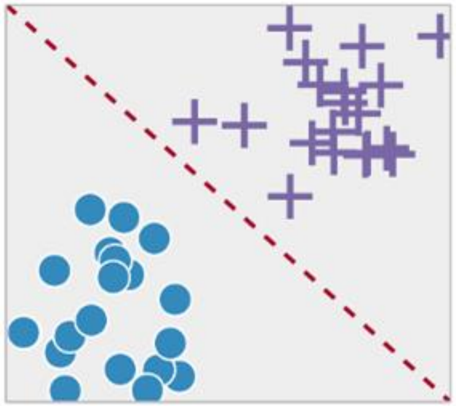
Regression



Learning a function for a **continuous** output

Eg. Predicting sales price of house.

Classification



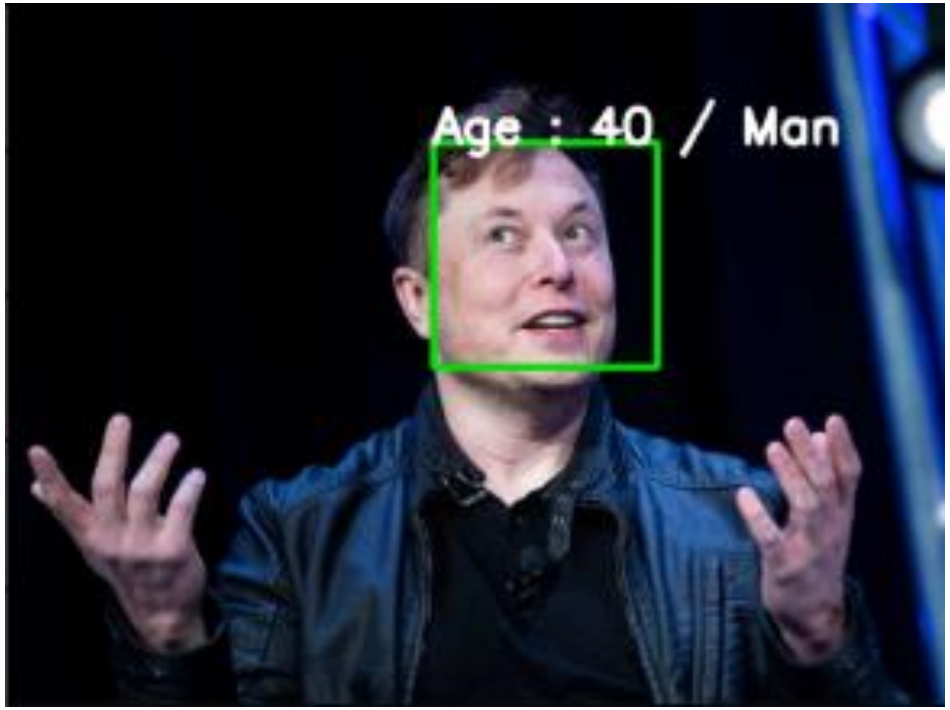
Learning a function for a **categorical** output

Eg. Classifying cats vs dogs in images.

Regression

Gives a **continuous output**.

Example: Age and Gender Prediction



https://github.com/ChibaniMohamed/cnn_age_gender

Classification

Gives a **discrete output.**

Example: Fruit Classification

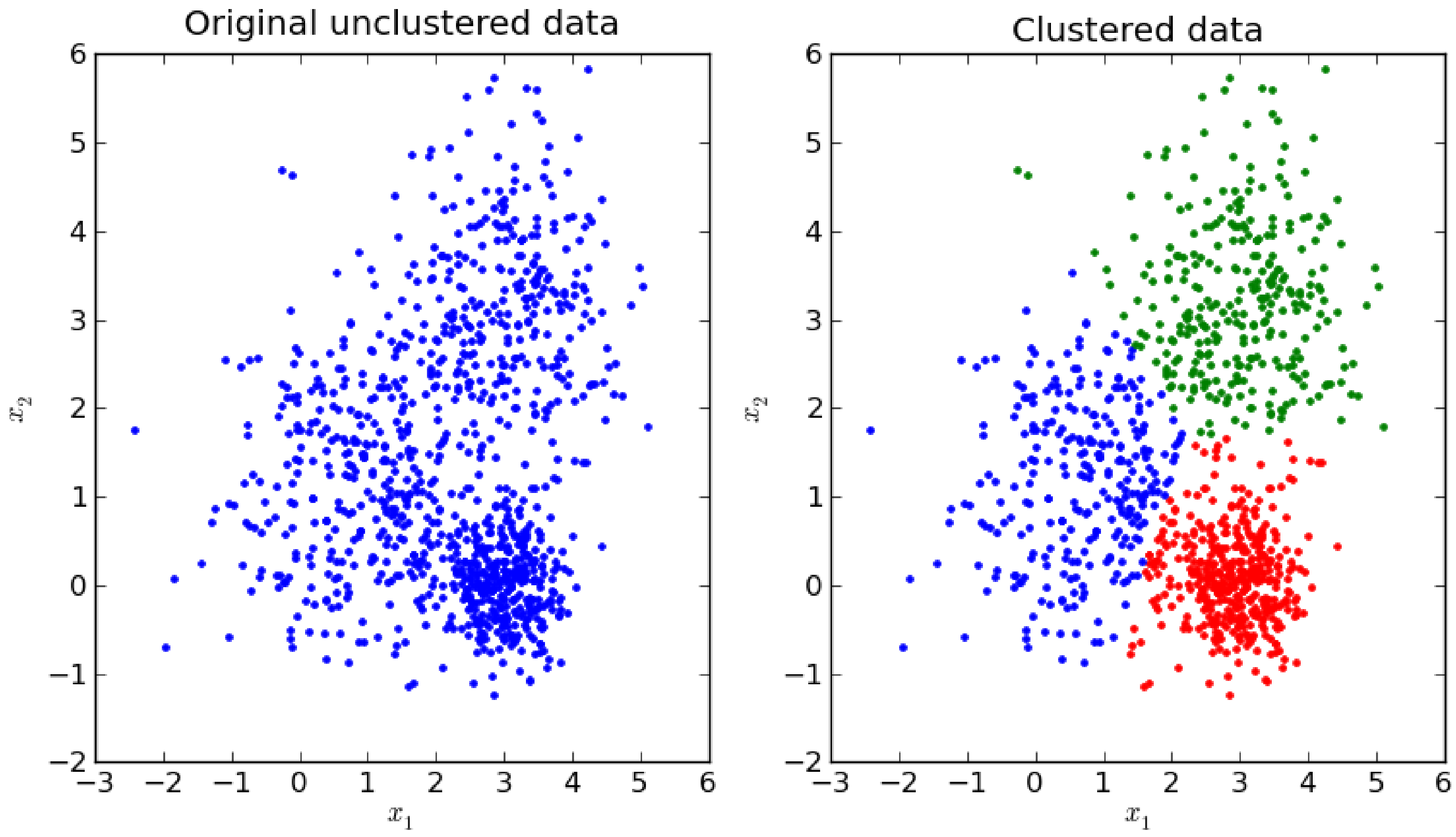
Some Basic Terminology

Features/ Attributes					**Target Variable**
Colour	**Mass**	**Shape**	**Seeds**	**Country**	**Fruit**
Red	100g	Round	Yes	Canada	Apple
Yellow	647 g	Curved	No	Australia	Banana

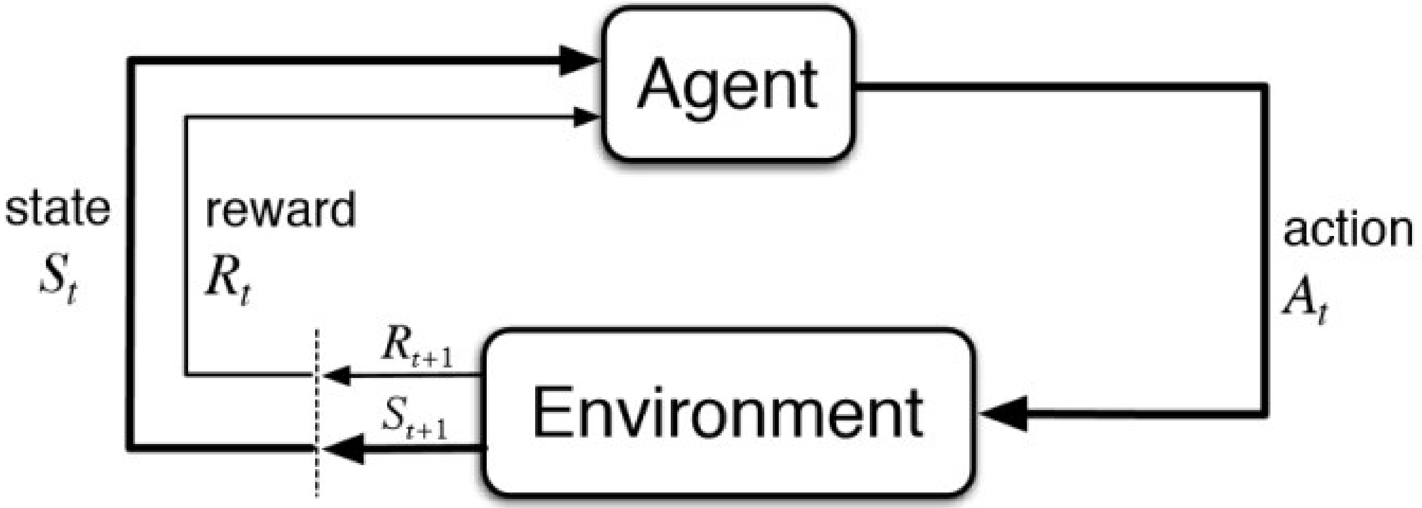
Features / attributes: how you would describe the fruit
Target variable: how you want to teach your model to recognize the fruit. (ground truth)

Unsupervised Learning

Train your model to learn how to difference input data, and make prediction on its own without training labels.



Reinforcement Learning



Your system learns to behave in an evolving environment and make prediction by learning from the outcome of specific actions.

Goal: learn the actions (Good) that **maximize** the reward.

Machine Learning Pipeline

1 Identify Problem

Carefully define the problem you want to solve. What specific question are you trying to answer?

2 Gather Data

Figure out what data is needed and where to retrieve it. Does similar data exist or do we need to generate it?

3 Process Data

Format data that can be interpreted by a computer. That includes cleaning, manipulating and extracting important features to feed into the training model.

4 Train Model

Training the dataset on your selected model. In practice, datasets are split into train, validation and test sets in order to measure model performance.

5 Evaluate Results

Does the trained model solve your initial problem? Does it satisfy your performance requirements?

6 Repeat!

Improve your model by reiterating the process!

Overview of Supervised Learning I

--- Variable Types and Terminology

Variable Types and Terminology

Input: a variable X. If X is a vector, its j-th element is X_j

an observation x_i
(scalar or vector)

Typically, we use i to denote the index of observations, while use j to denote the index of variables.

Model $f_\theta(\cdot)$

$$N \text{ observations} \left\{ \begin{array}{c} X \\ \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_N \end{bmatrix} \end{array} \right. \quad \mathbf{x} \in \mathbb{R}^N$$

$$\begin{array}{c} X_1 \quad \cdots \quad X_j \quad \cdots \quad X_p \\ N \text{ observations} \left\{ \begin{bmatrix} x_1^T \\ \vdots \\ x_i^T \\ \vdots \\ x_N^T \end{bmatrix} \right. \\ \underbrace{\qquad\qquad\qquad\qquad}_{p \text{ variables}} \end{array} \quad \mathbf{X} \in \mathbb{R}^{N \times p}$$

Variable Types and Terminology

Input: a variable X. If X is a vector, its j-th element is X_j

an observation x_i

Binary classification

coding

$= 2$ a binary digit. $\{0,1\}$ or $\{-1,1\}$

> 2 K binary digits. One is "on" at a time.

Model $f_\theta(\cdot)$

Output

- qualitative: G (categorical or discrete) → **classification**
- quantitative: Y → **regression**

Multi-class classification

Variable Types and Terminology

Input: a variable X. If X is a vector, its j-th element is X_j

an observation x_i

Model $f_\theta(\cdot)$

Main question of this course:
Given the value of an input vector X,
make a good prediction $\hat{Y}$ of the output Y.

Output

- qualitative: G (categorical or discrete) $\longrightarrow$ **classification**
- quantitative: Y $\longrightarrow$ **regression**

Overview of Supervised Learning I

--- Least Squares and Nearest Neighbors

Simple Approach 1: Least Squares

- Given inputs:

$$X^T = (X_1, X_2, \dots, X_p)$$

- Predict output Y via the model

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

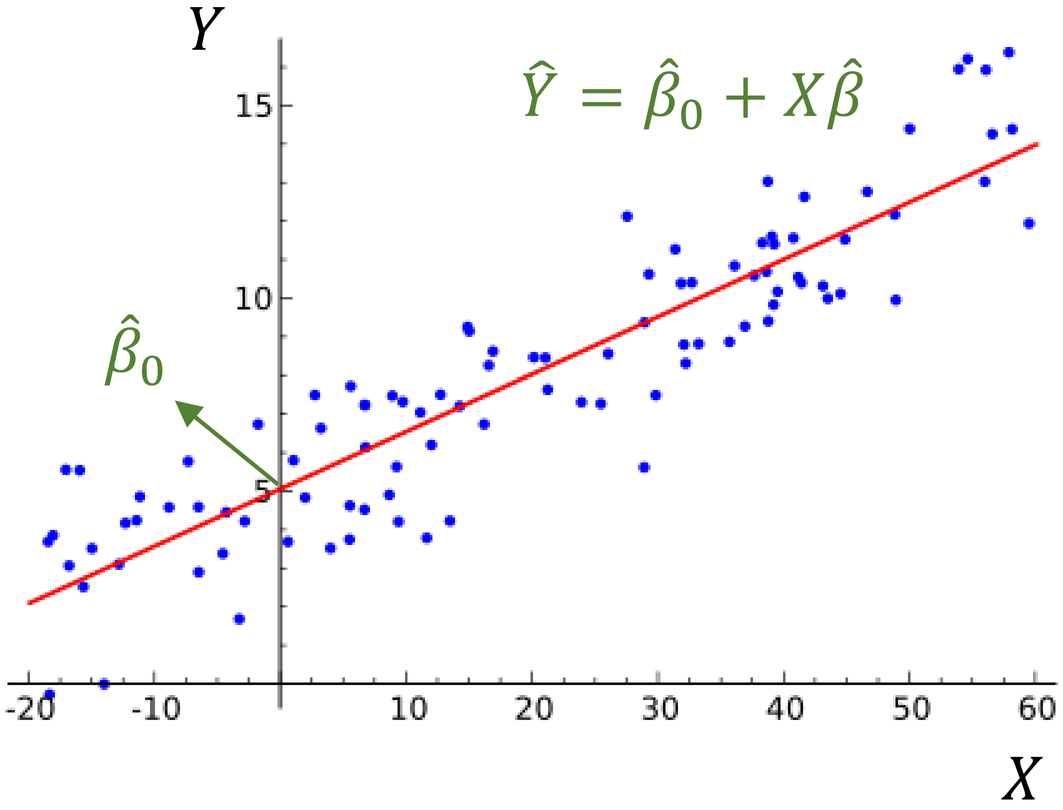
$\hat{\beta}_0$: bias or intercept

- Include the constant variable 1 in X

$$\hat{Y} = X^T \hat{\beta}$$

- Here $\hat{Y}$ is a scalar. If the output $\hat{Y}$ is K-vector, then $\hat{\beta}$ is a $p \times K$ matrix of coefficients.

Multi-output regression



Simple Approach 1: Least Squares

- Given inputs:

$$X^T = (X_1, X_2, \dots, X_p)$$

- Predict output Y via the model

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

$\hat{\beta}_0$: bias or intercept

- Include the constant variable 1 in X

$$\hat{Y} = X^T \hat{\beta}$$

- Here $\hat{Y}$ is a scalar. If the output $\hat{Y}$ is K-vector, then $\hat{\beta}$ is a $p \times K$ matrix of coefficients.

- In the $(p+1)$-dimensional input-output space, $(X, \hat{Y})$ represents a hyperplane
- If the constant is included in X, then the hyperplane goes through the origin

$$f(X) = X^T \beta$$

is a linear function

- Its gradient

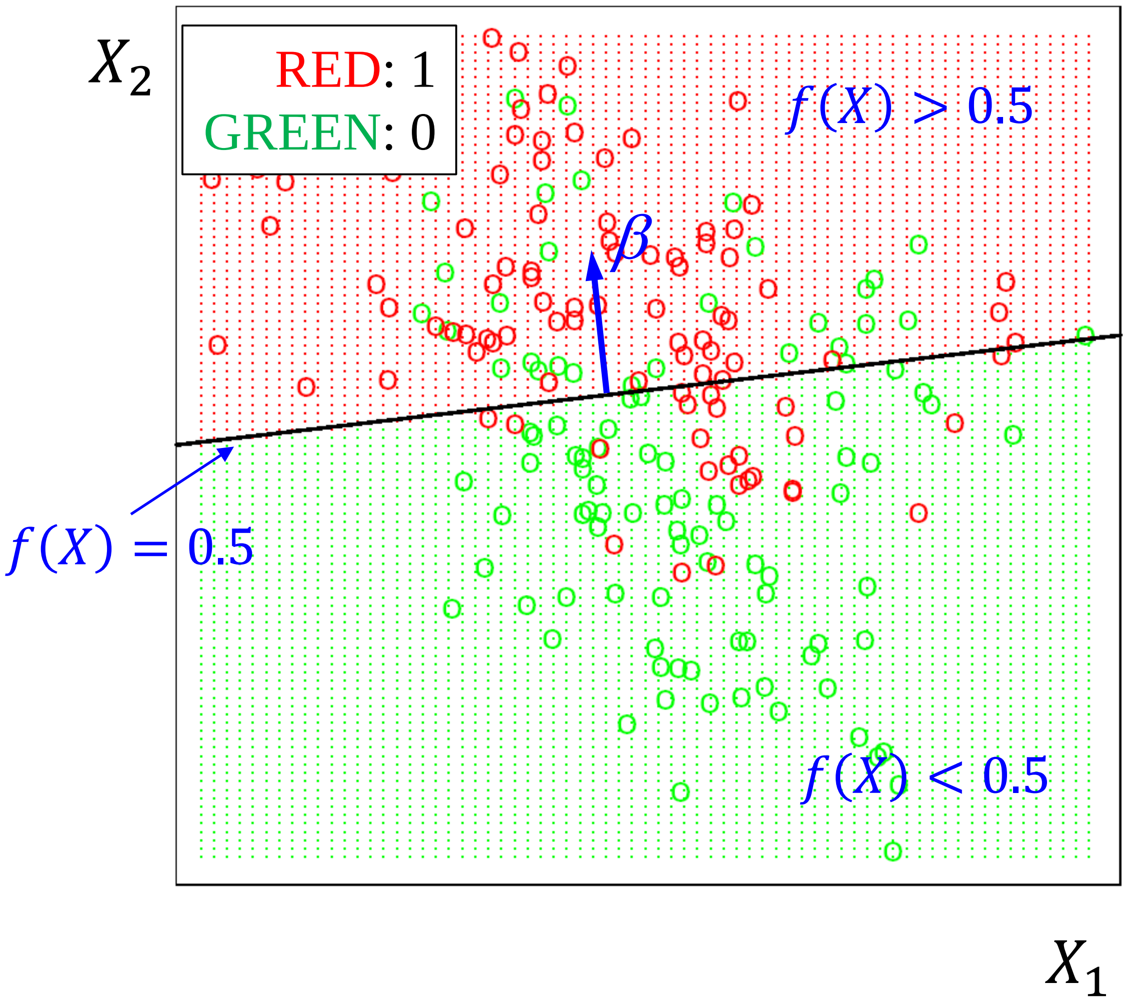
$$\boxed{f'(X) = \beta}$$

is a vector that points in the steepest uphill direction.

For the derivatives of vectors and matrices, please refer to:

- **The Matrix Cookbook**. Kaare Brandt Petersen and Michael Syskind Pedersen

Simple Approach 1: Least Squares



- In the $(p+1)$-dimensional input-output space, $(X, \hat{Y})$ represents a hyperplane
- If the constant is included in X, then the hyperplane goes through the origin

$$f(X) = X^T \beta$$

is a linear function
- Its gradient

$$f'(X) = \beta$$

is a vector that points in the steepest uphill direction.

Simple Approach 1: Least Squares

- Training procedure:
  Method of *least-squares*
- N = #observations
- Minimize the *residual sum of squares*

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2$$

Or equivalently,

$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$
$$= \|\mathbf{y} - \mathbf{X}\beta\|_2^2$$

- This quadratic function always has a
  global minimum, but it may not be unique.

Note: for an arbitrary vector $\boldsymbol{a}$, we have the squared ℓ_2-norm $\|\boldsymbol{a}\|_2^2 = \boldsymbol{a}^T \boldsymbol{a}$.

Simple Approach 1: Least Squares

- Training procedure:
  Method of *least-squares*
- N = #observations
- Minimize the *residual sum of squares*

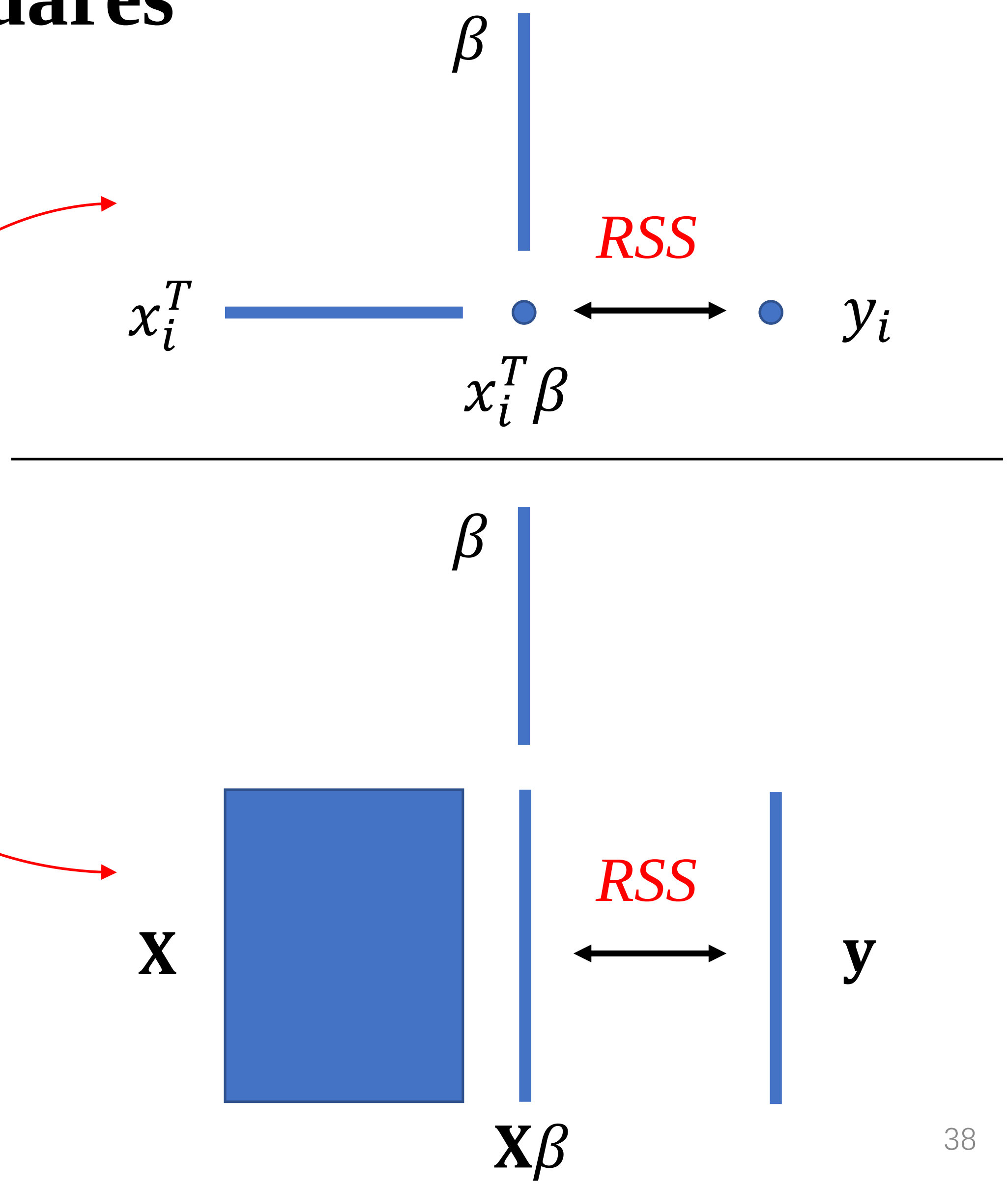
$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} \boxed{(y_i - x_i^T\beta)^2}$$

Or equivalently,

$$\mathrm{RSS}(\beta) = \boxed{(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)}$$
$$= \|\mathbf{y} - \mathbf{X}\beta\|_2^2$$

- This quadratic function always has a global minimum, but it may not be unique.

Q: What is the difference among x_i, x_i^T, $\mathbf{x}$, X and $\mathbf{X}$?



Simple Approach 1: Least Squares

- Training procedure:
  Method of *least-squares*
- N = #observations
- Minimize the *residual sum of squares*

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2$$

  Or equivalently,

$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$
$$= \|\mathbf{y} - \mathbf{X}\beta\|_2^2$$

- This quadratic function always has a global minimum, but it may not be unique.

- Differentiating w.r.t. β yields the *normal equations*

$$\mathbf{X}^T (\mathbf{y} - \mathbf{X}\beta) = 0$$

- If $\mathbf{X}^T\mathbf{X}$ is nonsingular, then the unique solution is

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- The fitted value at an arbitrary input x_0 is

$$\hat{y}(x_0) = x_0^T \hat{\beta}$$

- The entire fitted surface is characterized by $\hat{\beta}$.

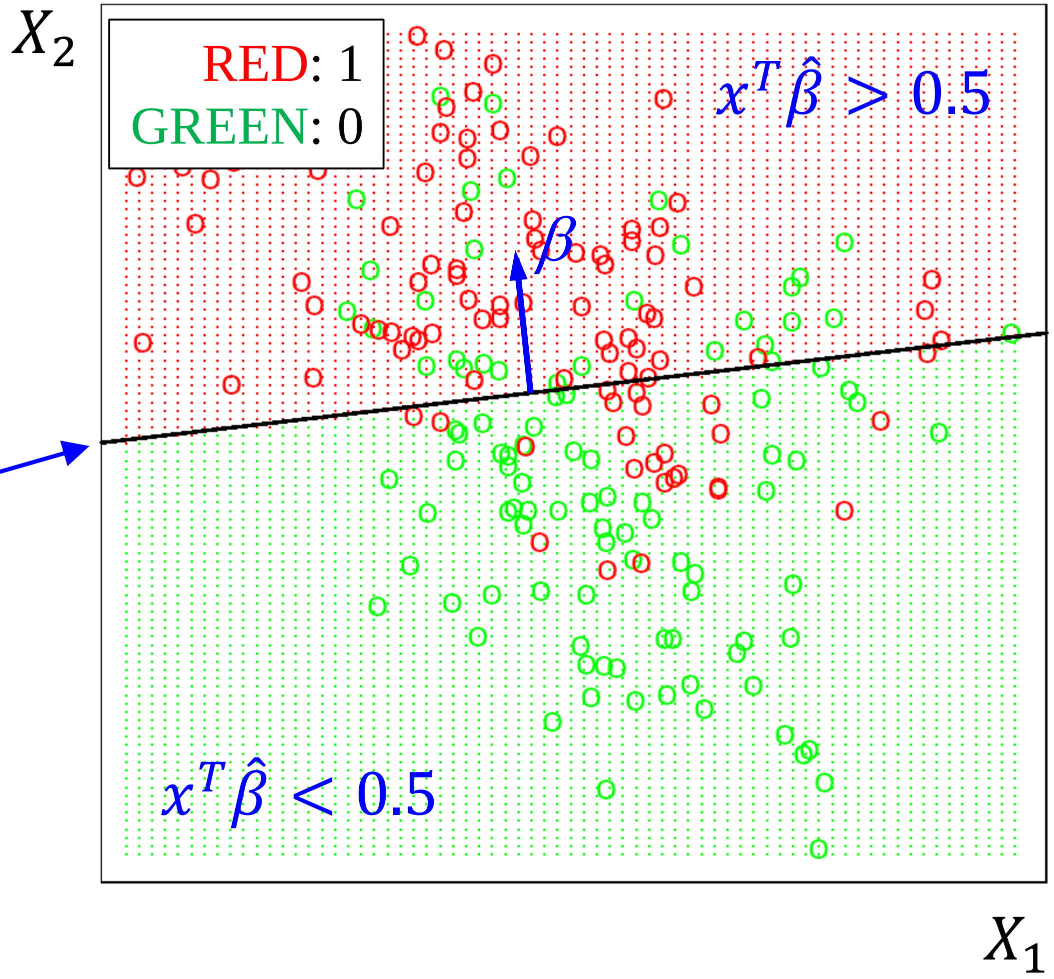
Simple Approach 1: Least Squares

Example:

- Data on two inputs X_1 and X_2.
- Output variable has values GREEN (coded 0) and RED (coded 1).
- 100 points per class.
- Regression line is defined by

$$x^T\hat{\beta} = 0.5.$$

- Easy but many misclassifications if the problem is not linear.

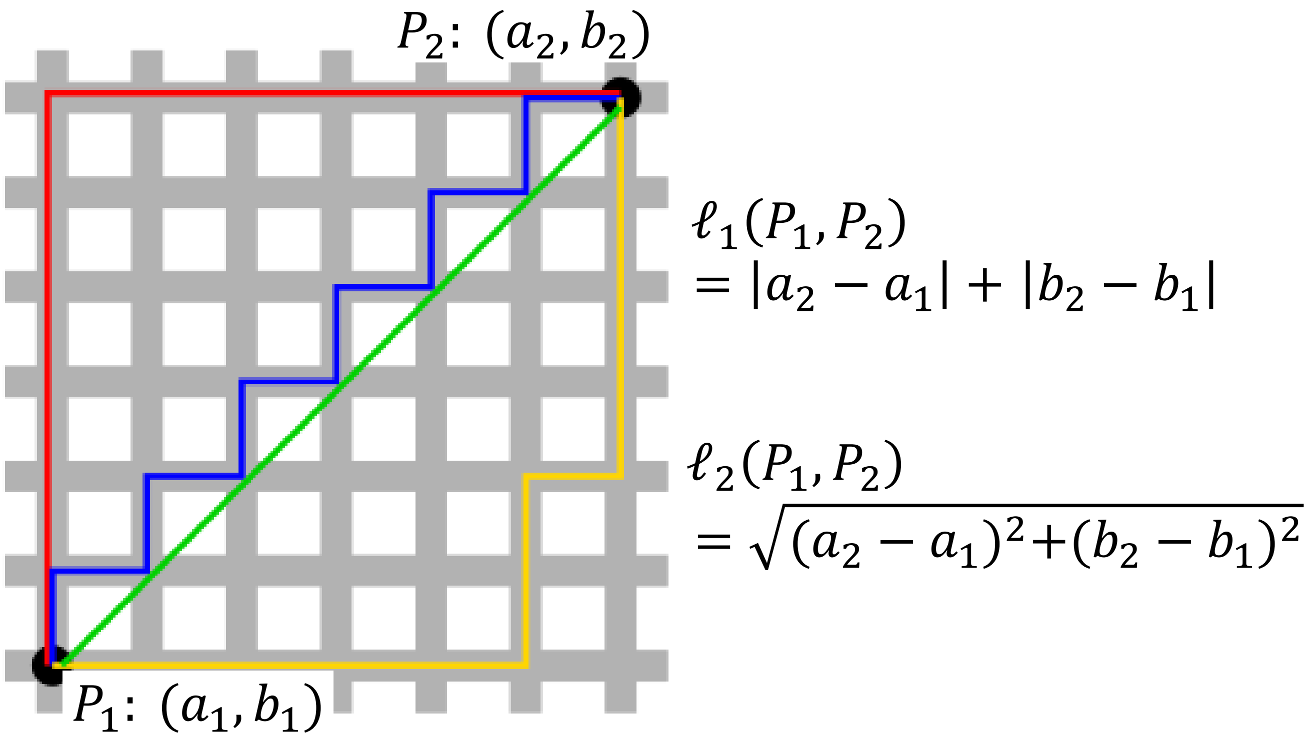


Simple Approach 2: Nearest Neighbors

- Use observations in the training set closest to the given input.

$$\hat{Y}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i .$$

- $N_k(x)$ is the set of the k closest points to x is the training sample
- Average the outcome of the k closest training sample points



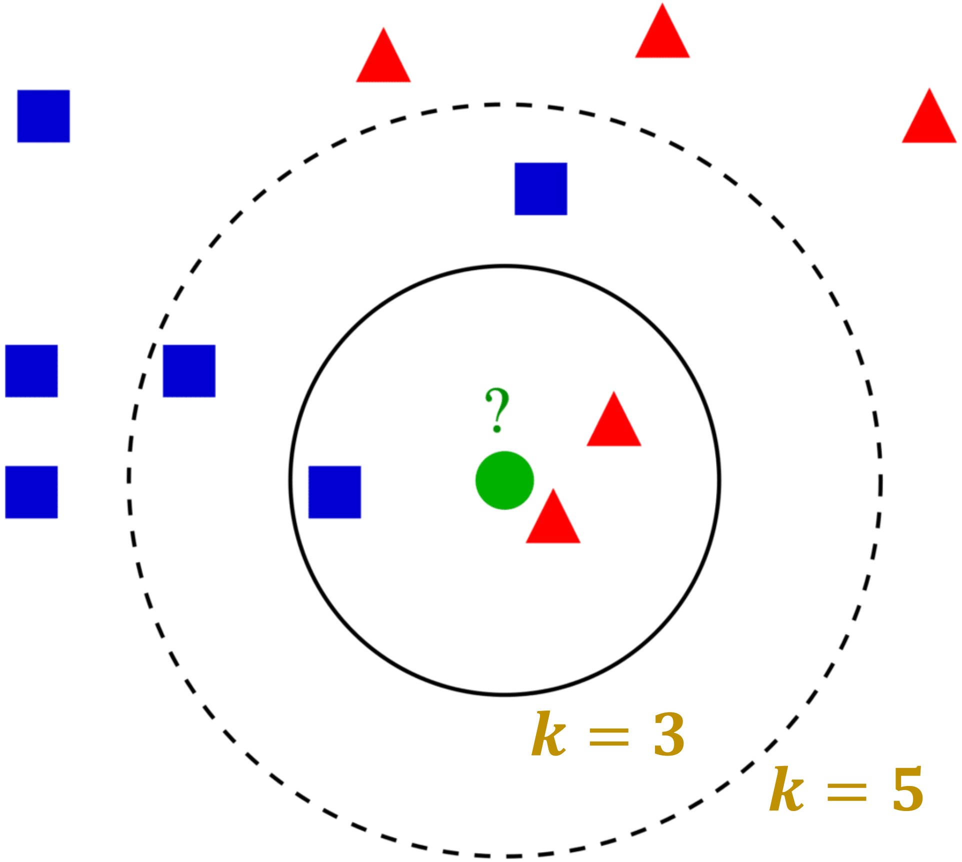
Taxicab geometry $(\boldsymbol{\ell_1})$ versus **Euclidean distance** $(\boldsymbol{\ell_2})$: In taxicab geometry, the red, yellow, and blue paths all have the same shortest path length of 12. In Euclidean geometry, the green line has length $6\sqrt{2} \approx 8.49$ and is the unique shortest path.

Simple Approach 2: Nearest Neighbors

- Use observations in the training set closest to the given input.

$$\hat{Y}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i.$$

- $N_k(x)$ is the set of the k closest points to x is the training sample
- Average the outcome of the k closest training sample points



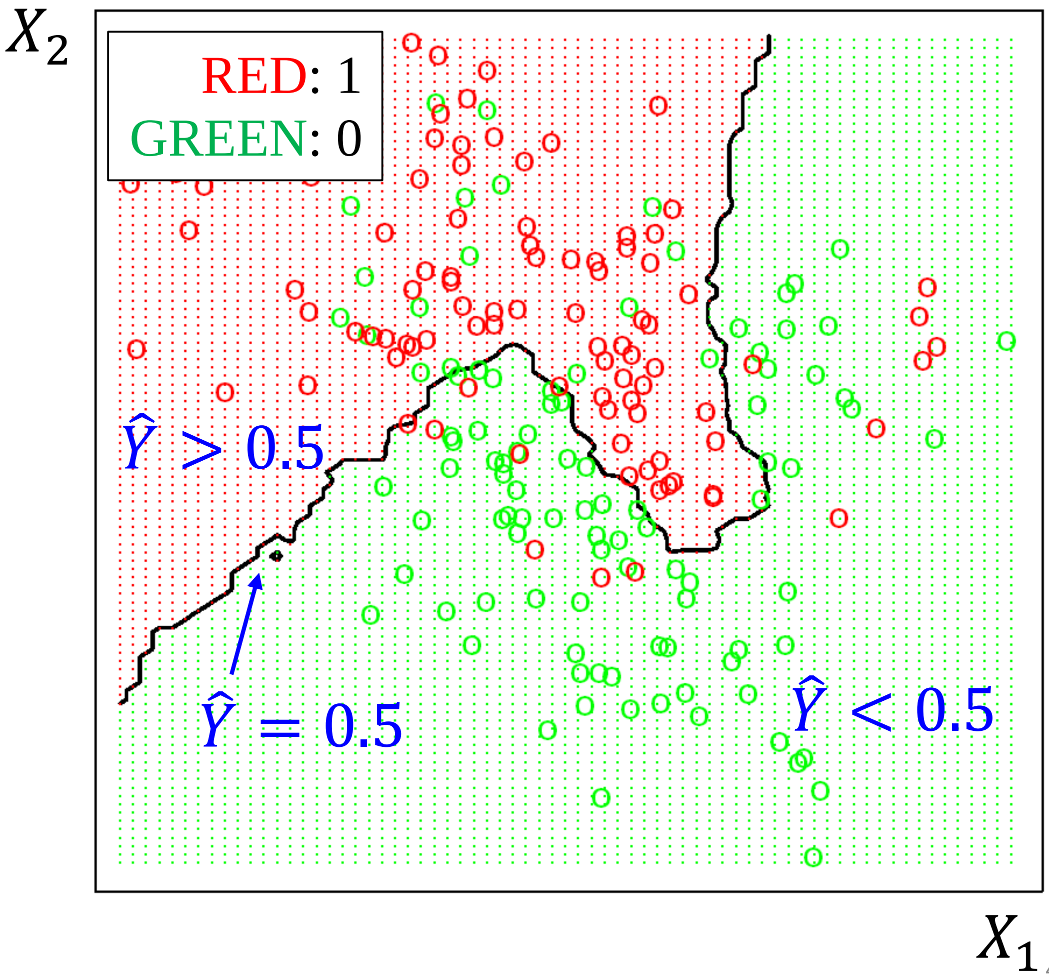
Simple Approach 2: Nearest Neighbors

- Use observations in the training set closest to the given input.

$$\hat{Y}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i.$$

- $N_k(x)$ is the set of the k closest points to x is the training sample
- Average the outcome of the k closest training sample points
- Fewer misclassifications

15-nearest neighbors averaging



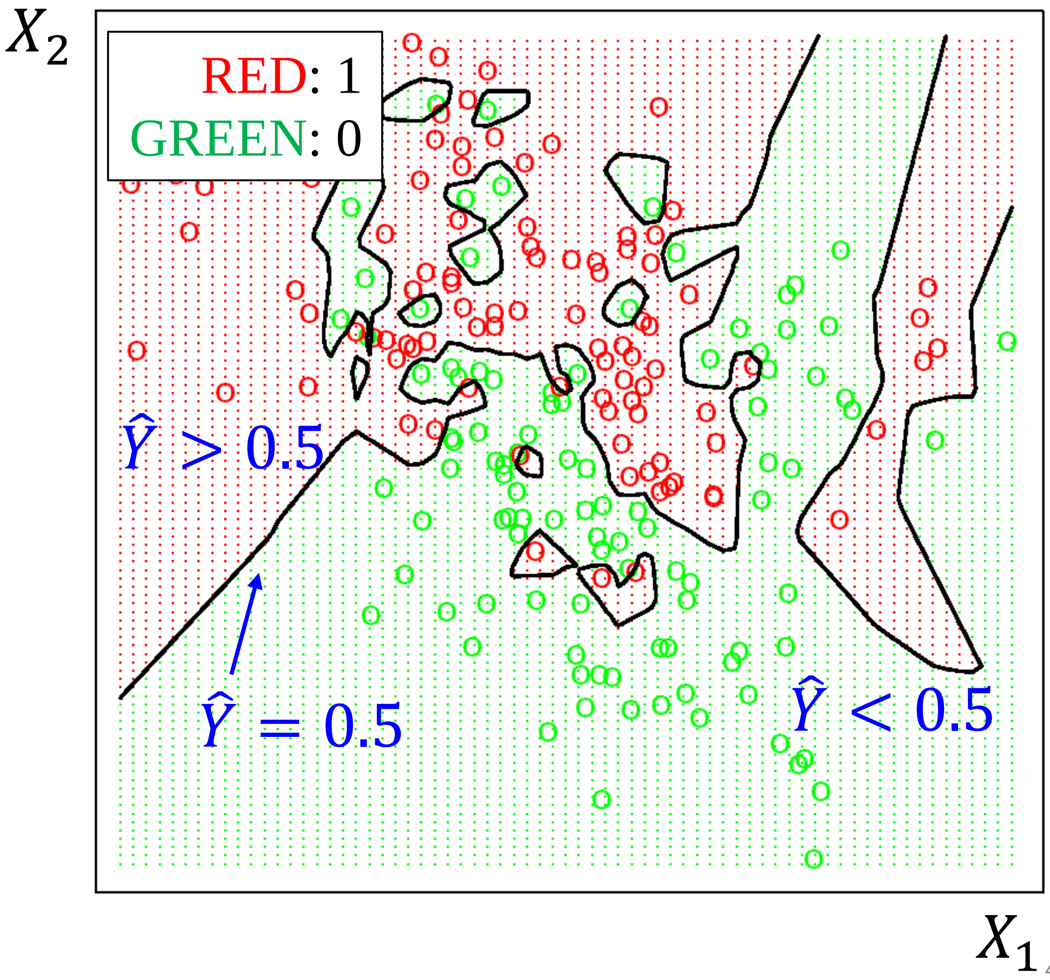
Simple Approach 2: Nearest Neighbors

- Use observations in the training set closest to the given input.

$$\hat{Y}(x) = \frac{1}{k}\sum_{x_i \in N_k(x)} y_i.$$

- $N_k(x)$ is the set of the k closest points to x is the training sample
- Average the outcome of the k closest training sample points
- No misclassifications: overtraining

1-nearest neighbors averaging



Simple Approach 2: Nearest Neighbors

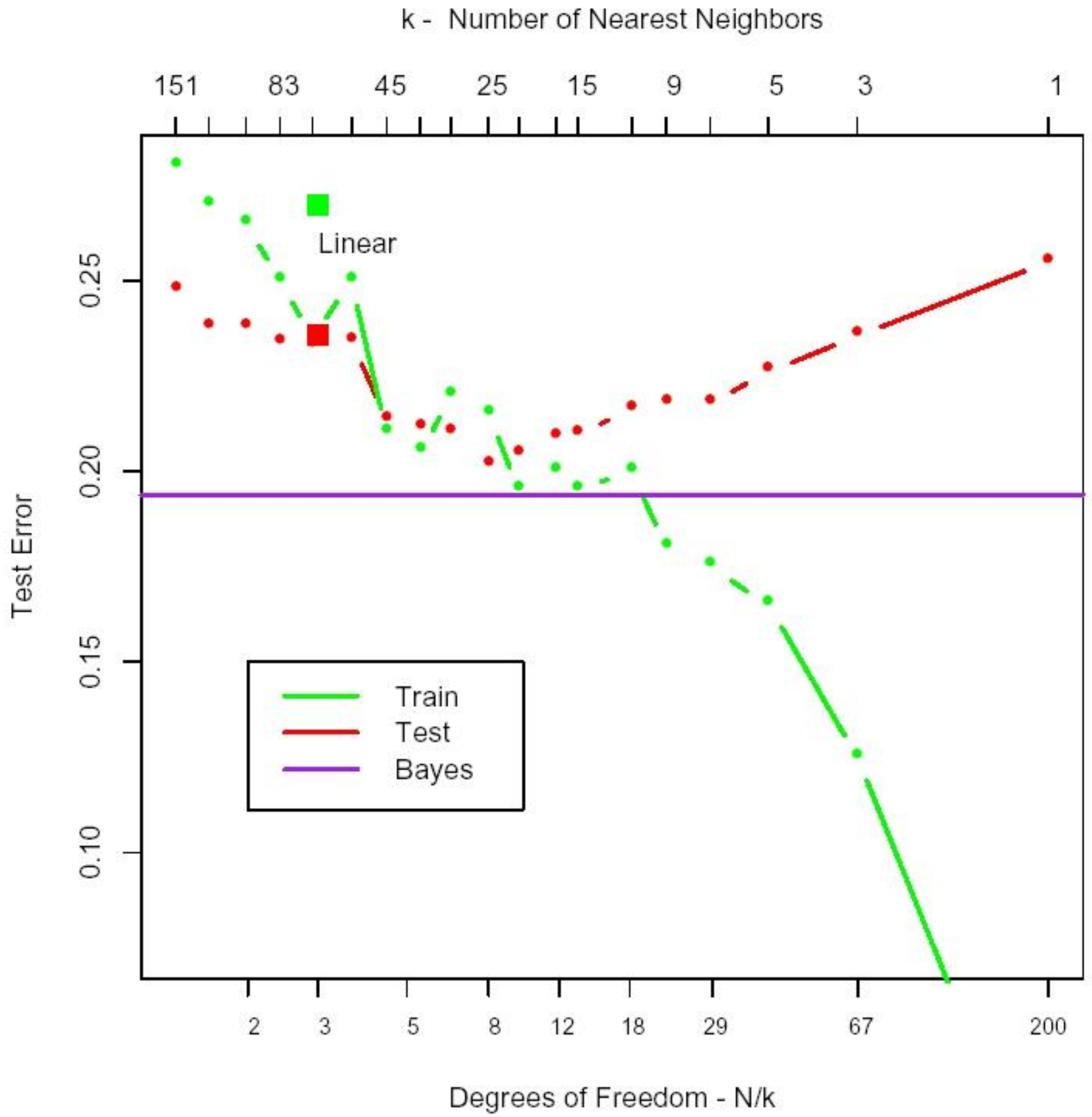
Pros:

- Simple algorithm, easy to implement (good baseline)
- No training time
- Easily scalable to multiple classes
- Works for “unusual” data distributions

Cons:

- Expensive query for test instances (time intensive)
- Memory intensive: stores data instead of parameters
- Not suitable for high-dimensional data (curse of dimensionality)

Comparison of the Two Simple Approaches



Comparison of the Two Approaches

Linear regression	k-nearest neighbors
p parameters (p = #variables)	$\frac{N}{k}$ parameters (k: hyperparameter) (N = #observations)
Low variance (robust)	High variance (not robust)
High bias (strong assumption)	Low bias (mild assumption)

Appendix

Symbol	Statistics	Machine Learning
X	variable, covariable predictor independent variable	feature attribute
Y	response dependent variable	label
x_i	observation data point	example instance
β	weights coefficients	parameters
$f(\cdot)$	model	learner

Difference between Statistics and Machine Learning in Terminology