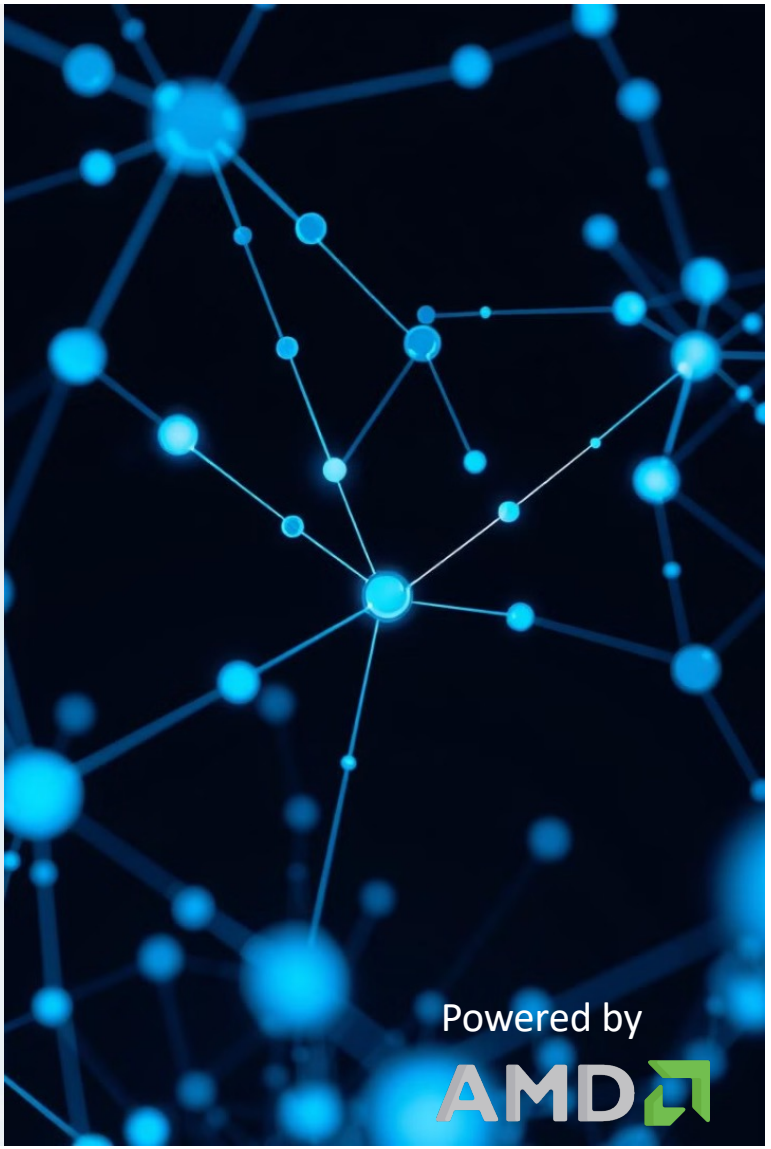
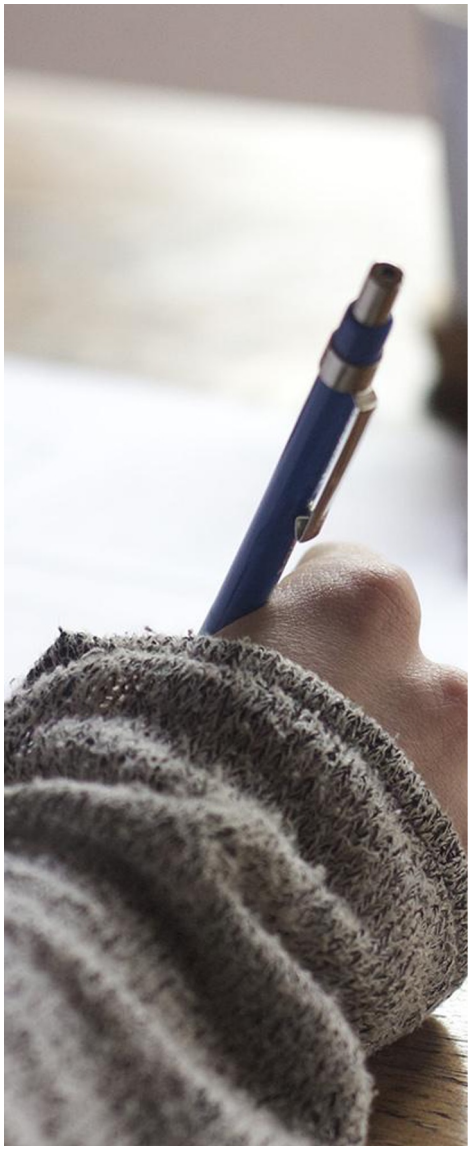
Introduction to AI

Lecture 1: Class Overview and Unsupervised Learning

Welcome to our course on Artificial Intelligence. This introductory lecture will cover the foundations, history, and key concepts of AI. We'll explore its evolution from early concepts to modern deep learning techniques.

by Hong-Han Shuai
National Yang Ming Chiao Tung University





HELLO!

I am Hong-Han Shuai.

Deep Learning/Computer Vision/Natural Language Processing/Multimedia

You can find me at
hhshuai@nycu.edu.tw

ED-807

slido

Please download and install the Slido app on all computers you use

Join at slido.com
#3984303

ⓘ Start presenting to display the joining instructions on this slide.

ID+NAME+EMOJI

0800888帥宏翰😂

slido

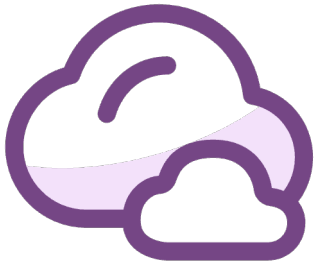
Please download and install the Slido app on all computers you use

Student ID+Name+Emoji

ⓘ Start presenting to display the poll results on this slide.

slido

Please download and install the Slido app on all computers you use



What are the first words that come to your mind when you hear 'Artificial Intelligence'?

ⓘ Start presenting to display the poll results on this slide.

What is Artificial Intelligence?

Definition

AI is the simulation of human intelligence in machines programmed to think and learn like humans.

Scope

AI encompasses various fields including machine learning, natural language processing, and robotics.

Goal

The ultimate aim is to create systems that can perform tasks requiring human-like intelligence.

Early Concepts and History of AI

1

1950s

Alan Turing proposes the Turing Test, a measure of machine intelligence.

2

1956

The term "Artificial Intelligence" is coined at the Dartmouth Conference.

3

1960s

Early AI programs like ELIZA and SHRDLU demonstrate natural language processing capabilities.

AI Winter and Revival

AI Winter

Period of reduced funding and interest in AI research due to unmet expectations.

Challenges

Limited computing power and unrealistic expectations led to setbacks in AI development.

Revival

Advancements in computing power and data availability sparked renewed interest in AI.

What is Machine Learning?

1 **Definition**

ML is a subset of AI that focuses on algorithms improving through experience.

2 **Supervised Learning**

Algorithms learn from labeled data to make predictions or classifications.

3 **Unsupervised Learning**

Algorithms discover hidden patterns in unlabeled data.

The Rise of Machine Learning

1

Classical AI

Rule-based systems and expert systems dominated early AI approaches.

2

Transition

Shift towards data-driven approaches and statistical learning methods.

3

Modern ML

Focus on algorithms that can learn and improve from experience.

Introduction to Neural Networks

Brain-Inspired

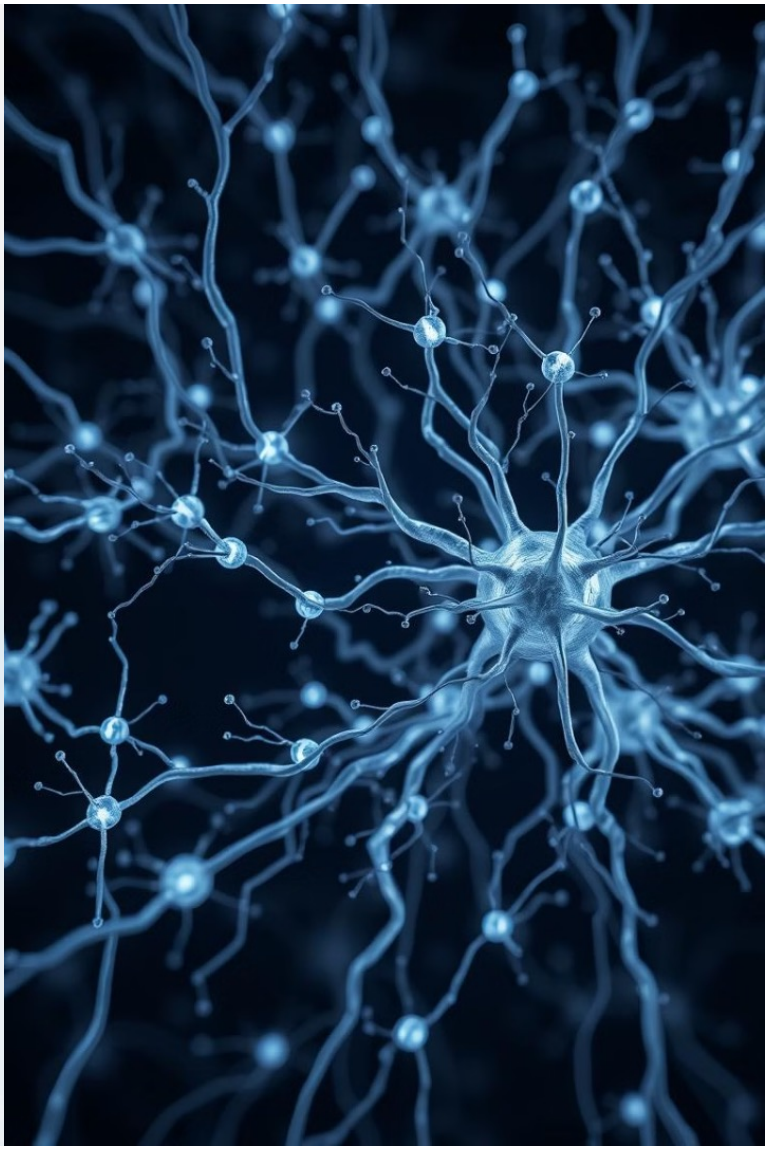
Neural networks are inspired by the structure and function of biological brains.

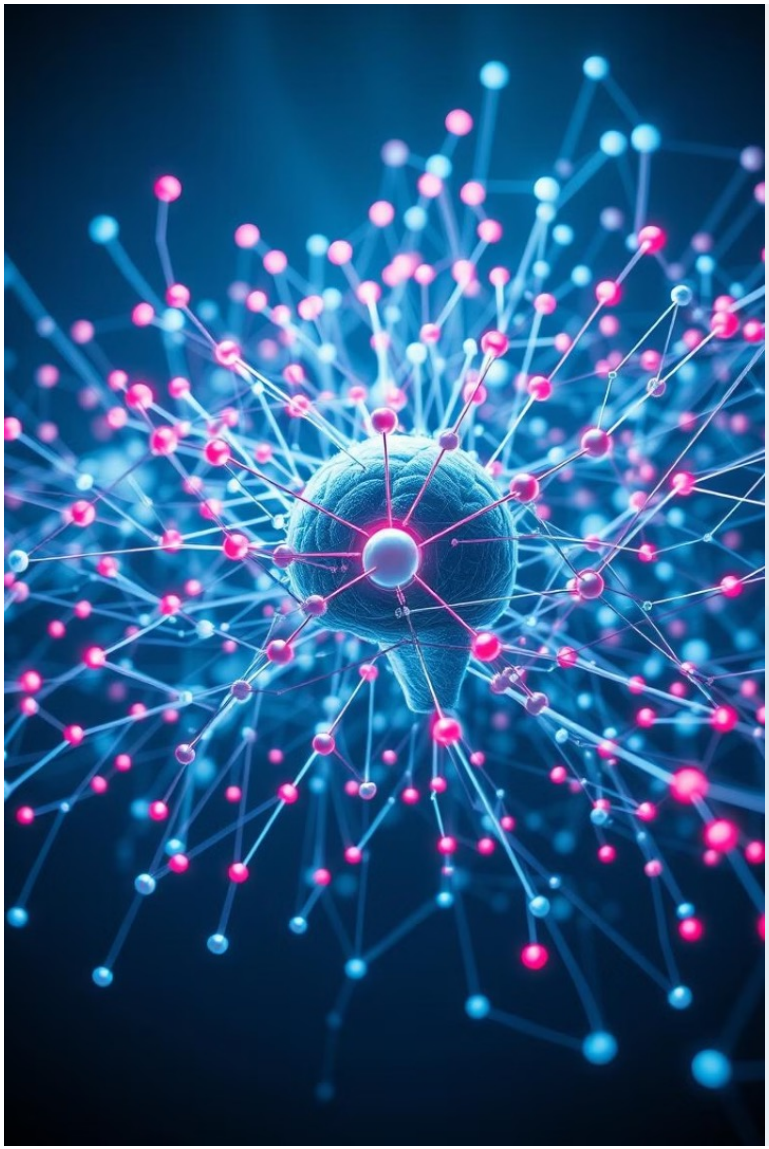
Layered Architecture

Composed of interconnected nodes organized in layers for information processing.

Learning Capability

Can adapt and improve performance through training on large datasets.





Rise of Deep Learning

1

2012

AlexNet wins ImageNet competition, sparking renewed interest in deep learning.

2

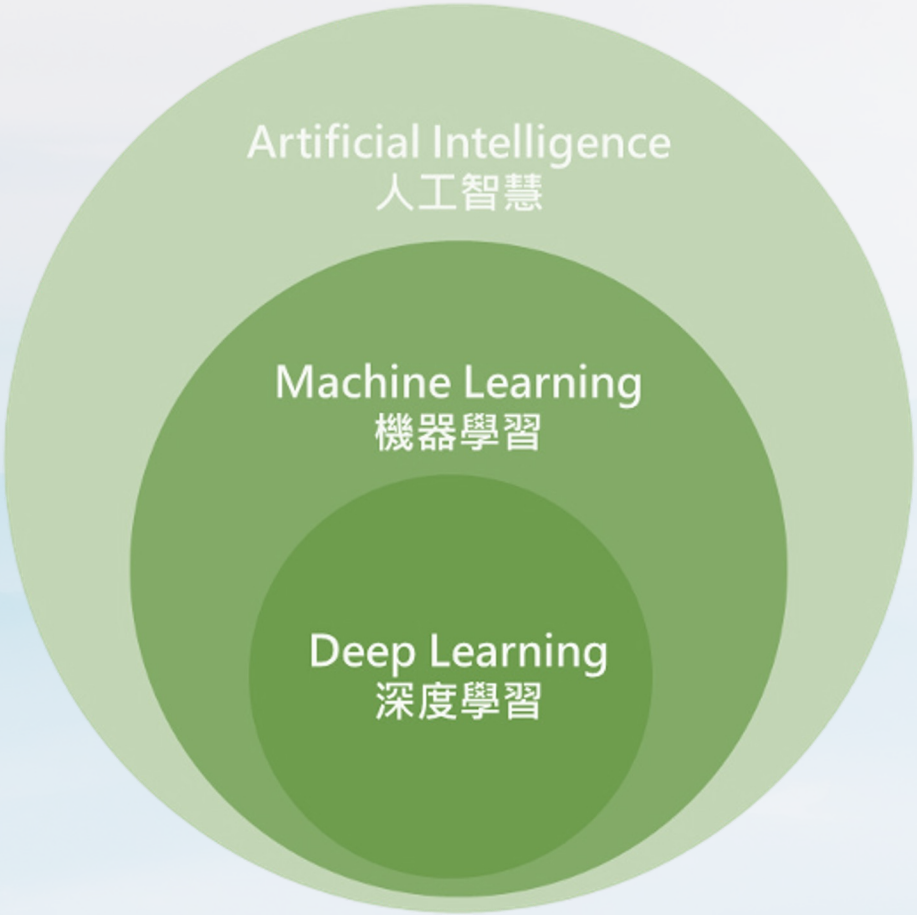
2016

AlphaGo defeats world champion in Go, demonstrating deep learning's potential.

3

2020

GPT-3 showcases impressive natural language generation capabilities.


Artificial Intelligence
人工智慧
Machine Learning
機器學習
Deep Learning
深度學習


Machine Learning不就是個黑盒子，
資料丟進去就跑出結果有什麼好做的？
丟進模型就改善10%，這不是誰都能做？
打死我也不要做Machine Learning
Deep Learning真香

Ethical Considerations in AI

Bias	Privacy	Accountability
Addressing biases in AI algorithms and training data	Protecting personal information in AI systems	Ensuring responsible development and use of AI

Future of AI

Autonomous Systems

Self-driving cars and advanced robotics will become more prevalent.

Healthcare Revolution

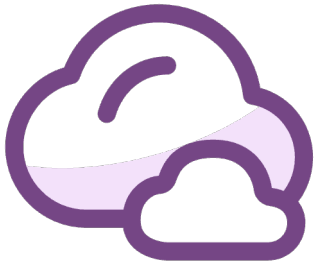
AI will enhance medical diagnosis and treatment planning.

Human-AI Collaboration

AI will augment human capabilities in various fields, including creative industries.

slido

Please download and install the Slido app on all computers you use



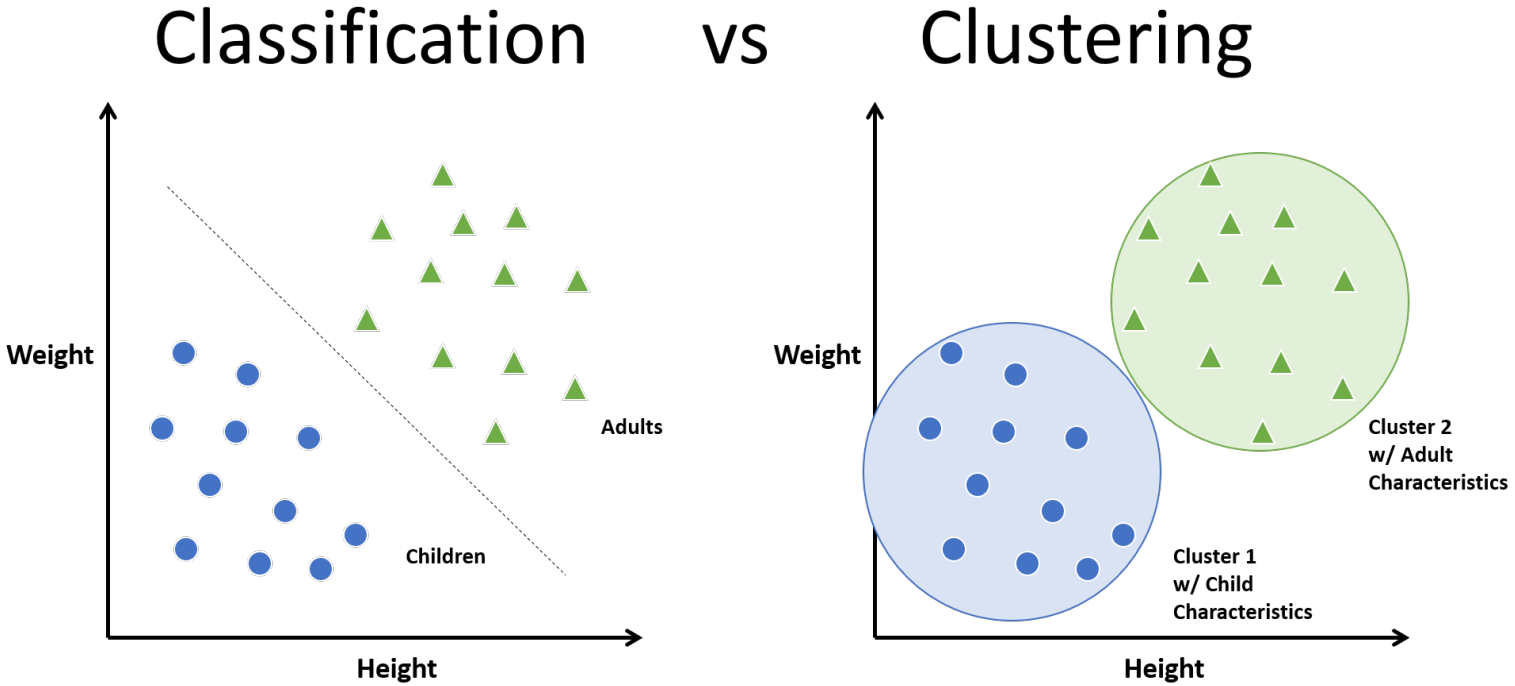
What specific knowledge or skills do you hope to gain from this course on Artificial Intelligence?

ⓘ Start presenting to display the poll results on this slide.

SYLLABUS

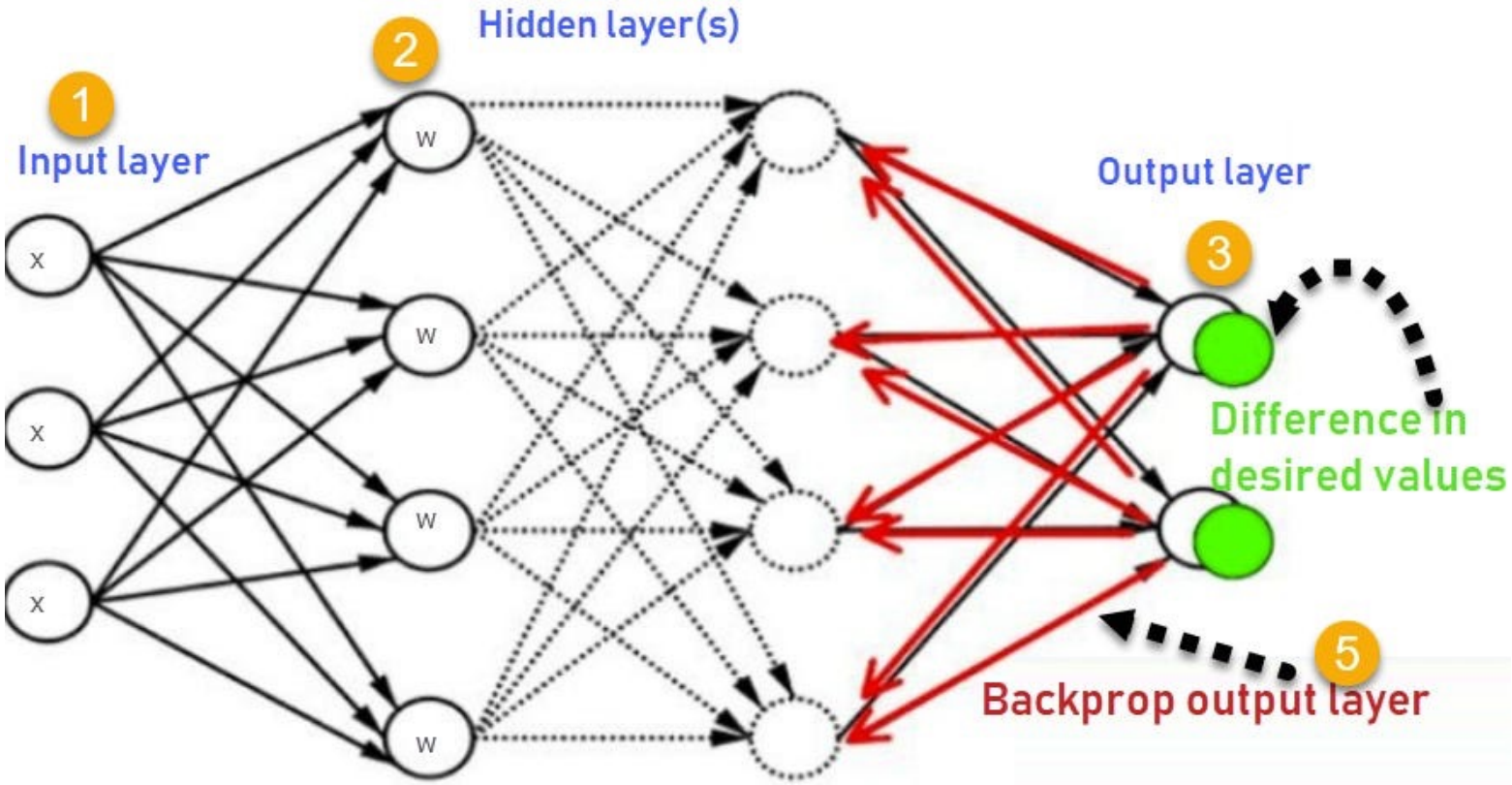
Week	Date	Contents
1	9/2	Lecture 1: Class Overview and Unsupervised Learning (HW#1)
2	9/9	Lecture 2: Traditional Classification-Part 1
3	9/16	Lecture 3: Traditional Classification-Part 2
4	9/23	Lecture 4: Neural Networks Basics (HW#2)
5	9/30	Hands-on Tutorials on PyTorch
6	10/7	Lecture 5: Deep Learning in Practice
7	10/14	Lecture 6: Introduction to Natural Language Processing
8	10/21	Midterm

HOMEWORK 1-CLUSTERING



PC: https://www.analyticsvidhya.com/blog/2021/05/what-why-and-how-of-spectral-clustering/

HOMEWORK 2



PC: https://ai.plainenglish.io/understanding-backpropagation-in-neural-networks-3634aad3a3c4

Homework 2

本次作業要練習推導反向傳播公式，並且使用 `python` 的 `numpy` 套件進行實作。為了有效測試實作結果是否正確，實作時必須嚴格遵照助教指定的實作規範，並使用於作業內提供的 `python` 測試腳本進行測試。

框架

神經網路可不失一般性的表示為：

$$\mathbf{y} = \sigma_N \circ f_{\mathbf{W}_N} \circ \sigma_{N-1} \circ f_{\mathbf{W}_{N-1}} \circ \cdots \circ \sigma_1 \circ f_{\mathbf{W}_1}(\mathbf{x})$$

其中輸入為 $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$，輸出為 $\mathbf{y} \in \mathbb{R}^{d_{\text{out}}}$，$\sigma$ 為激勵函數，$f_{\mathbf{W}}$ 是由$\mathbf{W}$參數控制的計算單元。

損失函數則是用於計算輸出和標準答案的差異：

$$l = \mathcal{L}(\mathbf{y}, \mathbf{y}_{\text{true}})$$

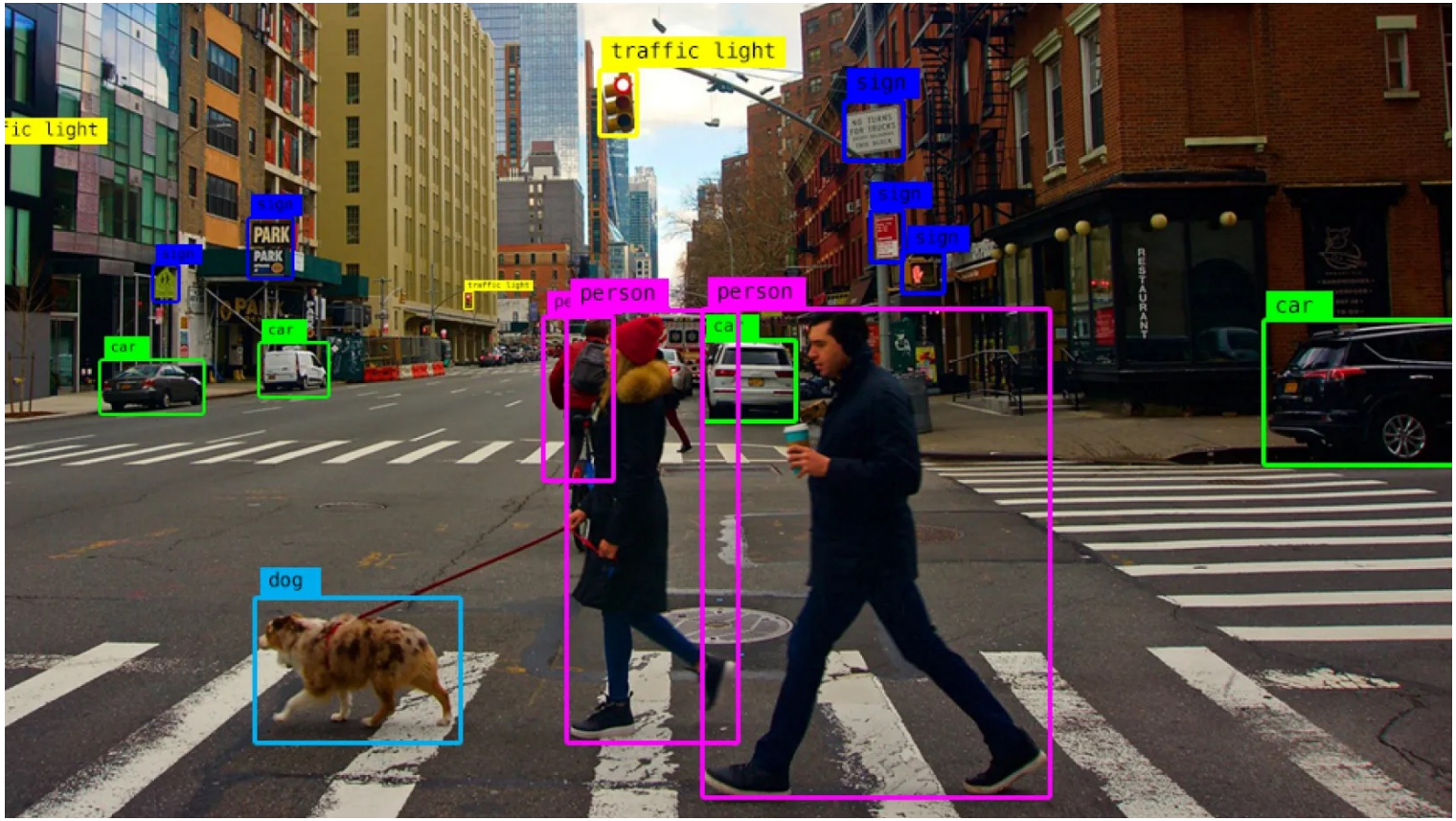
SYLLABUS

Week	Date	Contents
9	10/28	Lecture 7: Deep Learning for NLP+Hands-on Tutorials on NLP (HW#3)
10	11/4	Lecture 8: Introduction to Computer Vision
11	11/11	Lecture 9: Advanced Computer Vision
12	11/18	Lecture 10: Object Detection and Image Segmentation] Hands-on Tutorials on Object Detection (HW#4/Final Project proposal)
13	11/25	Self-Supervised Learning
14	12/2	Graph Neural Networks Processing (HW#5)
15	12/9	Reinforcement Learning
16	12/16	Ethics and Threats of AI

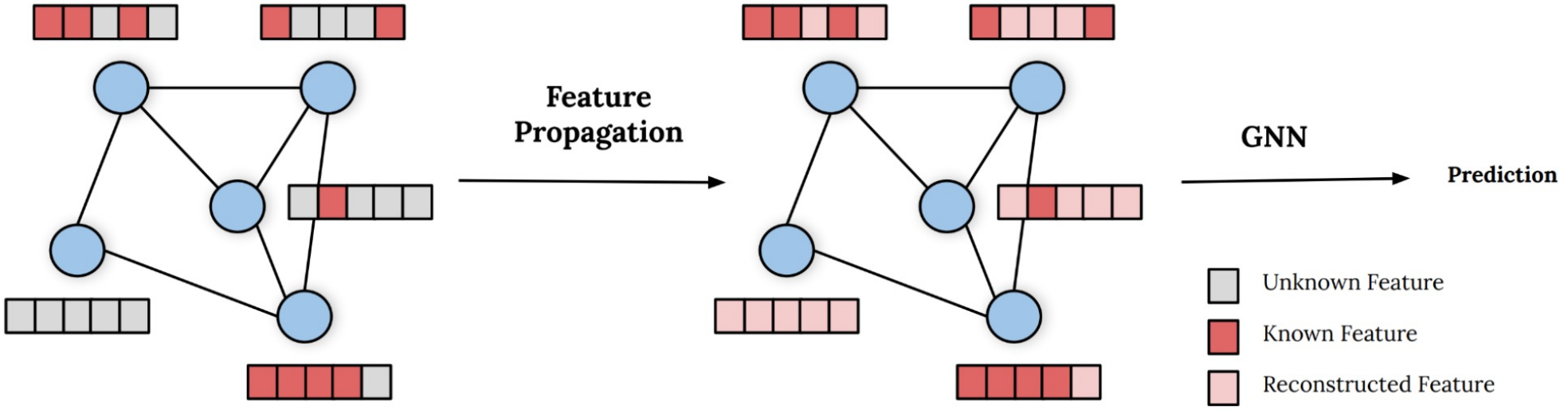
HOMEWORK 3-SENTIMENT ANALYSIS

PC: https://www.linkedin.com/pulse/decoding-emotions-using-text-data-natural-language-roy-rachman-sedik/

HOMEWORK 4-OBJECT DETECTION



HOMEWORK 5



PC: https://blog.x.com/engineering/en_us/topics/insights/2022/graph-machine-learning-with-missing-node-features

GRADING POLICY

- Midterm: 22%
- First Three Homework: 36% (12 pts each)

GRADING POLICY

- Midterm: 22%
- First Three Homework: 36% (12 pts each)
- Class Participation with Slido and Kahoot: 12%

CLASS PARTICIPATION WITH SLIDO AND KAHOOT

- Group-based evaluation for in-class problems
  - Act together we go far!
- Each team initially has
  - 2 cards
  - 5000 coins (1000 coins for 1 pt)

CARDS?

- Help
- Second Chance

COINS?

- Bid in-class problem (500-1000 coins)
- Buy cards (?)

New Cards?

GRADING POLICY

- Midterm: 22%
- First Three Homework: 36% (12 pts each)
- Class Participation with Slido and Kahoot: 12%

- Last Two Homework (15 pts each) or Final Project (30 pts)


AMD Ryzen™
Threadripper™
9000 Series
Processors
Designed to Deliver.
Built for Breakthroughs.
AMD
THREADRIPPER
PRO

AMD Radeon™ AI PRO R9700

Graphics

AMD RDNA4™ Architecture

Dual Slot Design

32GB GDDR6

256-bit Memory Interface

640 GB/s Memory Bandwidth

64 Compute Units

128 AI Accelerators

Up to **191 TFLOPS**
FP16 Dense

Up to **1531 TOPS**
INT4 Sparse

300W TDP

QUESTIONS?

- GPU? Yes. To complete the homework, you don't need hi-end GPU. You can try Colab/Colab Pro.
- Midterm? Hand-written with some explanation, justification and calculation.

Audience Q&A

GROUPING (分組)

- Only for Class Participation
- **At most 4** members in a group

https://tinyurl.com/NYCUAI-2025

AGENDA

- Philosophy
- Machine Learning Concept
- Clustering

WHY STUDY PHILOSOPHY?

- *It is not enough to have a good mind. The main thing is to use it well.* -- Rene Descartes
- Philosophy seeks not simply knowledge, but deep understanding and wisdom.
  - Those who study philosophy are engaged in asking, answering, evaluating, and reasoning about some of life's most basic, meaningful, and difficult questions, such as:
    - What is it to be a human?
    - What is to be human mind?
    - Are we responsible for what we do, or are we just helpless victims of our genes, environment, and upbringing?
    - Is there a God?
    - What is the best sort of life to live?


我在哪？我是誰？

WHAT IS PHILOSOPHY

- Definition:
  - Philosophy is just the activity that philosophers get up to.
  - Philosophy is the activity of working out the best way to think about things.
- Take Physics and Medicine as examples

Think in a particular way

vs.

Step back a think whether the way of thinking is correct or not

PHYSICS

- Conduct experiments and try and build theories on that basis.
  - When you're doing that, you're using way of thinking that's characteristic of physics.
- By stepping back from the activity of doing physics and thinking in that way, we can ask questions like:
  - What is it for data to confirm or refute a theory in physics?
  - What are we doing?
    - We are trying to measure reality.
  - And what does it even mean to try and understand reality in terms of its basic physical constituents.

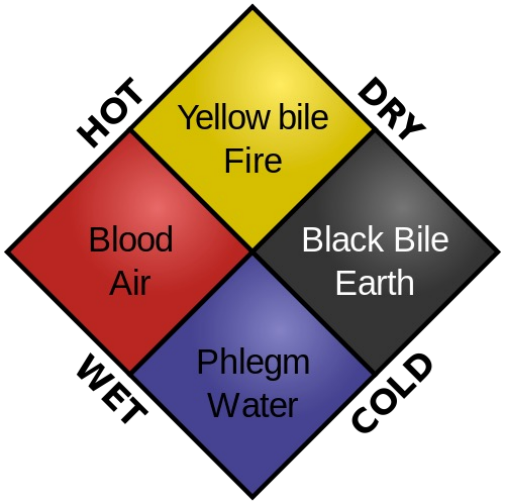
$D = 1/2\ gt^2$ (Calculates Dropping Distance)

It is essential that the student acquires an understanding of and a lively feeling for values. He must acquire a vivid sense of the beautiful and of the morally good. Otherwise he----with his specialized knowledge----more closely resembles a well-trained dog than a harmoniously developed person.

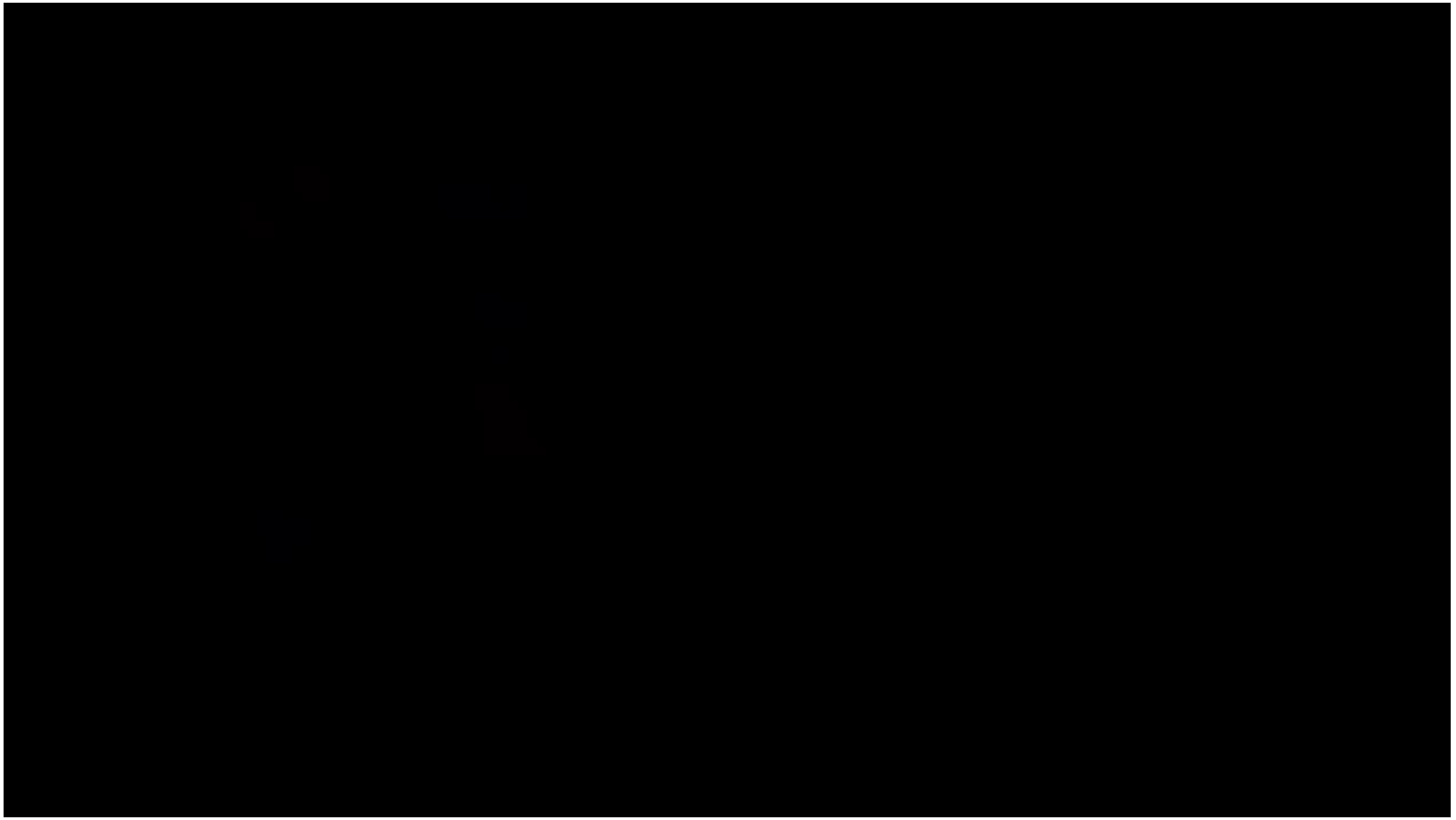
--Albert Einstein

MEDICINE

- In medieval times, Greek physician Hippocrates developed the theory of the four **humors**—blood, yellow bile, black bile, and phlegm—and their influence on the body and its emotions.



- http://tedxtaipei.com/articles/why-do-some-people-go-bald/



MEDICINE (2)

- They tried to understand that disease in terms of some kind of imbalance of those four humors and treat it accordingly.
- By stepping back from the activity of doing physics and thinking in that way, we can ask questions like:
  - What it really means for a disease to be an imbalance of 4 humors?
  - Are there other things in the body that seem to also be important to our physical health?
  - Are the results of treatments according to the theory really good?

BACK TO PHYSICS

- Quantum mechanics which suggest that one thing can instantaneously affect another thing that's very far away from it (doesn't seem to have any connection to it).
- Also, a thing can be like a wave in some respects but like a particle in some other respects.
  - It seems that according to our common sense conception of reality a thing can be either a wave or a particle but not both.
- Do NOT be naïve and just accept everything other people give you.
  - Otherwise, you'll be good at using tools instead of changing the world.

KEY

- The key is to act like a child.
  - Ask why first!
  - And continuously ask why, why, why in response to something.
  - Try to explain things in your way and have it tested.

BUT…

- If you're a **brain surgeon** or a **bomb disposal technician**, you don't ever have to step back and think philosophical questions about what it is to dispose of a bomb or what a brain is in order to be really, really great at your job.
- If you did spend your time stepping back and asking yourself those type of questions, then that would make you the worst brain surgeon or bomb disposal technician.

HOW TO DO (HOT DOGS EXAMPLE)

- Take the decision of whether we should go to a cinema tonight as an example.
- First, I may think about the pros and cons with arguments.
  - They have good hot dogs at the cinema.
  - I like hot dogs.

  Premises
  - Therefore, I should go to the cinema.

  Argument
- The conclusion follows from the premises
- When the conclusion of an argument follows from its premises, that means, when the conclusion has to be true if the premises are true, then the argument is valid.

HOW TO DO (AI EXAMPLE)

- Take the decision of whether we should use an AI system for text summarization as an example.
- First, I may think about the pros and cons with arguments.
  - AI can summarize long articles in seconds.
  - I often need to read many research papers quickly.

  Premises
  - Therefore, I should use AI for text summary.

  Argument
- The conclusion follows from the premises.

Valid and sound?

HOW TO DO?

- Take the decision of whether we should use deep learning as an example.
- First, I may think about the pros and cons with arguments.
  - Deep learning is good at data fusion.
  - The data contains multiple sources.
  - Therefore, I should use deep learning.

Premises

Argument

TRY ONE MORE

- Philosophers have tried to construct arguments that put pressure on that idea that we're free to choose what to do from one moment to the next.
- **First premise:** the way the world was in the past controls exactly how it is in the present, and how it will be in the future.
- **Second premise:** we're part of the world just like everything around us.
- **Third premise:** we can't control how things were in the past, or the way the past controls the present and future.
- **Conclusion:** therefore, we don't control anything that happens in the world, including all the things that we think and say, and do.

QUESTION YOURSELF

- **First premise:** the way the world was in the past controls exactly how it is in the present, and how it will be in the future.

怎麼反駁？

QUESTION YOURSELF

- **Second premise:** we're part of the world just like everything around us.

怎麼反駁？

QUESTION YOURSELF

- **Third premise:** we can't control how things were in the past, or the way the past controls the present and future.

PHILOSOPHER HILARY PUTNAM SAID

- By going through all the different arguments and positions that you're going to encounter there and really trying to critically engage with them, and work out what you think about them.
- Two little qualifications:
  - Vision
  - Argument

AGENDA

- Philosophy
- Machine Learning Concept
- Clustering

From Learning to Machine Learning

- What's learning?
  - knowledge or skill acquired by instruction or study

- What's machine learning?

http://www.csie.ntu.edu.tw/~htlin/mooc/

From Learning to Machine Learning

- What's learning?
  - knowledge or skill acquired by instruction or study

- What's machine learning?

http://www.csie.ntu.edu.tw/~htlin/mooc/

From Learning to Machine Learning

- What's learning?
  - knowledge or skill acquired by instruction or study

- What's machine learning?

  - acquiring skill with experience accumulated/computed from data

Types of Data Sets

- Record
  - Relational records
  - Data matrix, e.g., numerical matrix, crosstabs
  - Document data: text documents: term-frequency vector
  - Transaction data
- Graph and network
  - World Wide Web
  - Social or information networks
  - Molecular Structures
- Ordered
  - Video data: sequence of images
  - Temporal data: time-series
  - Sequential Data: transaction sequences
  - Genetic sequence data
- Spatial, image and multimedia:
  - Spatial data: maps
  - Image data:
  - Video data:

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

		OrgName1	OrgName2	OrgName3
ItemName1	DayValue	10	30	20
	WeekValue	20	10	10
ItemName2	DayValue	10	20	30
	WeekValue	30	10	90
ItemName3	DayValue	40	30	50
	WeekValue	50	90	30

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Data Objects

- Data sets are made up of data objects.
- A **data object** represents an entity.
- Examples:
  - sales database: customers, store items, sales
  - medical database: patients, treatments
  - university database: students, professors, courses
- Also called *samples, examples, instances, data points, objects, tuples*.
- Data objects are described by **attributes**.
- Database rows -> data objects; columns ->attributes.

Attribute (feature)

Data row

Name	Thread pitch (mm)	Minor diameter tolerance	Nominal diameter (mm)	Head shape	Price for 50 screws	Available at factory outlet?	Number in stock	Flat or Phillips head?
M4	0.7	4g	4	Pan	$10.08	Yes	276	Flat
M5	0.8	4g	5	Round	$13.89	Yes	183	Both
M6	1	5g	6	Button	$10.42	Yes	1043	Flat
M8	1.25	5g	8	Pan	$11.98	No	298	Phillips
M10	1.5	6g	10	Round	$16.74	Yes	488	Phillips
M12	1.75	7g	12	Pan	$18.26	No	998	Flat
M14	2	7g	14	Round	$21.19	No	235	Phillips
M16	2	8g	16	Button	$23.57	Yes	292	Both
M18	2.1	8g	18	Button	$25.87	No	664	Both
M20	2.4	8g	20	Pan	$29.09	Yes	486	Both
M24	2.55	9g	24	Round	$33.01	Yes	982	Phillips
M28	2.7	10g	28	Button	$35.66	No	1067	Phillips
M36	3.2	12g	36	Pan	$41.32	No	434	Both
M50	4.5	15g	50	Pan	$44.72	No	740	Flat

Attributes

- **Attribute (**or **dimensions, features, variables**): a data field, representing a characteristic or feature of a data object.
  - *E.g., customer _ID, name, address*
- Types:
  - Nominal (類別，彼此間無順序, categorical)
  - Binary
  - Ordinal
  - Numeric: quantitative
    - Interval-scaled (no true zero point)
    - Ratio-scaled (Inherent zero point)

Attribute Types

Nominal: categories, states, or "names of things"

- *Hair_color* = {*auburn, black, blond, brown, grey, red, white*}
- marital status, occupation, ID numbers, zip codes

Binary

- Nominal attribute with only 2 states (0 and 1)
- Symmetric binary: both outcomes equally important
  - e.g., gender
- Asymmetric binary: outcomes not equally important
  - e.g., medical test (positive vs. negative)
  - Convention: assign 1 to most important outcome (e.g., HIV positive)

Ordinal

- Values have a meaningful order (ranking) but magnitude between successive values is not known.
- *Size* = {*small, medium, large*}, grades, army rankings

Numeric Attribute Types

Quantity (integer or real-valued)

Interval

- Measured on a scale of **equal-sized units**
- Values have order
  - E.g., *temperature in C˚or F˚, calendar dates*
- No true zero-point

Ratio

- Inherent **zero-point**
- We can speak of values as being an order of magnitude larger than the unit of measurement (10 K˚ is twice as high as 5 K˚).
  - e.g., *temperature in Kelvin, length, counts, monetary quantities*

Discrete vs. Continuous Attributes

Discrete Attribute

- Has only a finite or countably infinite set of values
  - E.g., zip codes, profession, or the set of words in a collection of documents
- Sometimes, represented as integer variables
- Note: Binary attributes are a special case of discrete attributes

Continuous Attribute

- Has real numbers as attribute values
  - E.g., temperature, height, or weight
- Practically, real values can only be measured and represented using a finite number of digits
- Continuous attributes are typically represented as floating-point variables

Skill?

- Skill: improve the performance measurements (e.g., prediction, 3pts shooting percentage)
- Therefore, machine learning is defined to be the improvements on some performance measurements by computing from data.
- For example,
  - Image data -> **ML** -> Better image generation quality
  - Signal data -> **ML** -> Faster antenna direction detection
  - Sequential web log data -> **ML** -> Higher accuracy of anomaly detection/efficiency of caching algorithm
  - Stock data -> **ML** -> More money

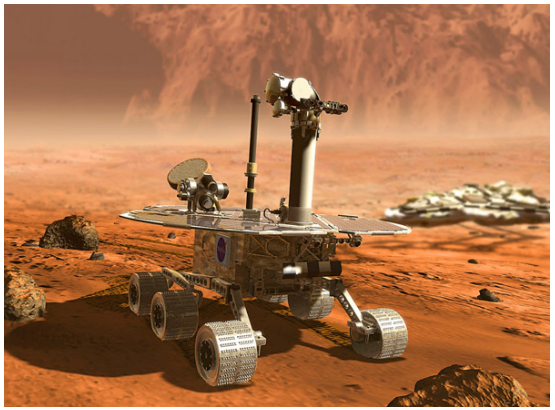
The Machine Learning Route

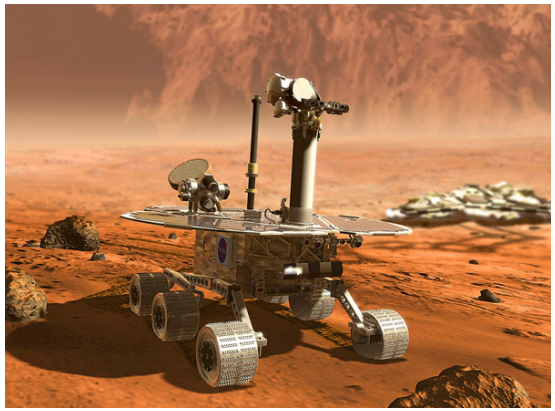
- ML: an **alternative route** to build complicated systems
- Some Use Scenarios
  - when human cannot program the system manually
    - navigating on Mars
  - when human cannot 'define the solution' easily
    - speech/visual recognition
  - when needing rapid decisions that humans cannot do
    - high-frequency trading
  - when needing to be user-oriented in a massive scale
    - consumer-targeted marketing

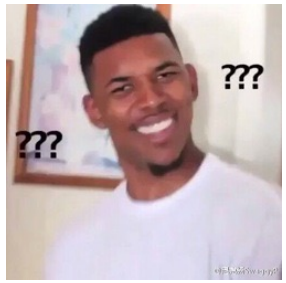
The Machine Learning Route

- ML: an **alternative route** to build complicated systems
- Some Use Scenarios
  - when human cannot program the system manually
    - navigating on Mars
  - when human cannot 'define the solution' easily
    - speech/visual recognition
  - when needing rapid decisions that humans cannot do
    - high-frequency trading
  - when needing to be user-oriented in a massive scale
    - consumer-targeted marketing

Navigating on Mars







http://www.memebucket.com/mb/2015/02/I-got-it-423.png
http://www.rutgersprep.org/kendall/7thgrade/cycleA_2008_09/ds/Mars-Rover.jpg
http://res.pokemon.name/common/pokemon/pgl/201.00.pn
https://cdn2.ettoday.net/images/2471/2471912.jp

The Machine Learning Route

- ML: an **alternative route** to build complicated systems
- Some Use Scenarios
  - when human cannot program the system manually
    - navigating on Mars
  - when human cannot 'define the solution' easily
    - speech/visual recognition
  - when needing rapid decisions that humans cannot do
    - high-frequency trading
  - when needing to be user-oriented in a massive scale
    - consumer-targeted marketing

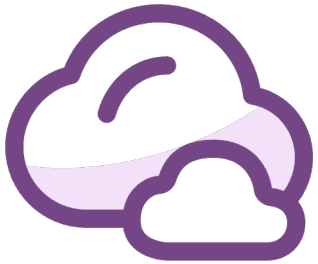
Game Time!

- Describe this thing using the appearance

http://www.cartier.hk/content/dam/rcq/car/92/55/67/925567.png.scale.120.high.png

slido

Please download and install the Slido app on all computers you use



Describe the ring using the appearance

ⓘ Start presenting to display the poll results on this slide.

The Machine Learning Route

- ML: an **alternative route** to build complicated systems
- Some Use Scenarios
  - when human cannot program the system manually
    - navigating on Mars
  - when human cannot 'define the solution' easily
    - speech/visual recognition
  - when needing rapid decisions that humans cannot do
    - high-frequency trading
  - when needing to be user-oriented in a massive scale
    - consumer-targeted marketing

http://www.cartier.hk/content/dam/rcq/car/92/55/67/925567.png.scale.120.high.png

Key Essence of Machine Learning

- Exists some ‘underlying pattern’ to be learned
  - So performance measure can be improved
- But no programmable (easy) definition
  - So machine learning is required
- Somehow there is data about the pattern
  - So machine learning has inputs to learn from

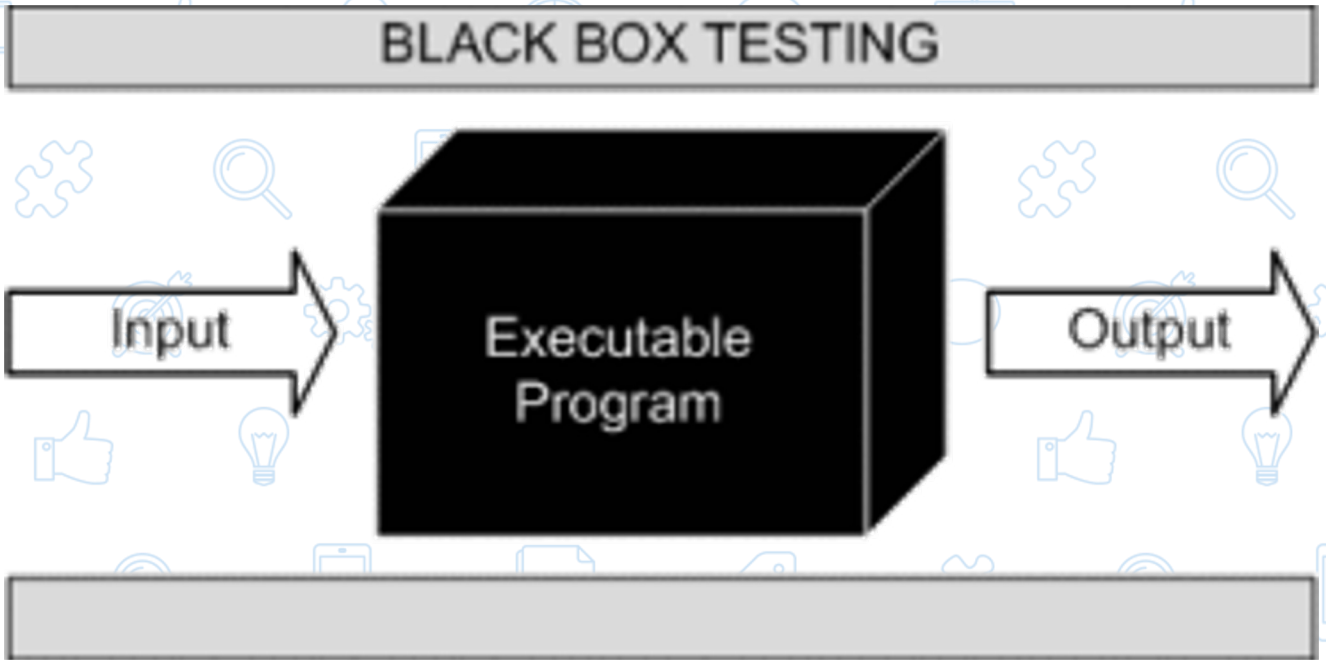
Which of the following is best suited for machine learning?

CREDIT CARD APPROVAL

- Applicant Information
  - 年齡：<30
  - 工作資歷： 同公司約3年
  - 公司規模： 資本額約10M
  - 公司年齡： >10
  - 貸款： 沒有
  - 學貸： 沒有不良還款紀錄： 無
  - 財務狀況： 薪轉銀行(約770K活存)、 Richart(約800K活存)
- Approve or not? What's the level of the card?

ZIP CODE EXAMPLE

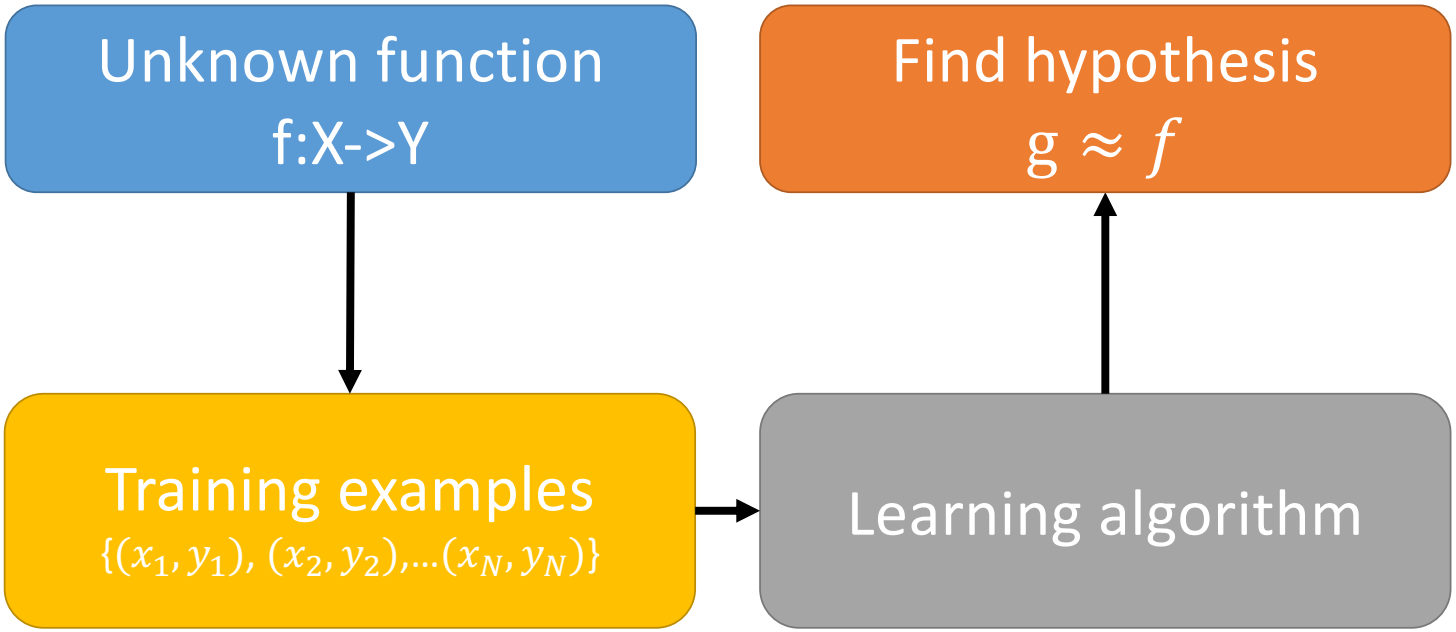
Name	Zip Code	Personality	Loan	Work Exp.	Company	Deposit	Results
Austin	30010	N	N	5	L	1M	Y
Bob	30010	P	P	3	M	1M	Y
Charles	59487	P	N	3	L	2M	N
David	59487	P	P	4	L	500K	N
Eve	59487	N	P	7	M	200K	N
Frank	59487	N	N	1	S	100K	N
Gary	59487	P	N	5	L	1M	N


BLACK BOX TESTING
Input
Executable
Program
Output

Learning Problem Formulation

- Notation
  - **Input:** $x \in X$(application)
  - **Output:** $y \in Y$(good/bad after approving)
  - **Unknown pattern to be learned can be formulated as a function**
    - f: X->Y (ideal function)
  - **Data:** D=$\{(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)\}$
  - **Hypothesis:**
    - g:X->Y (hopefully can be as close to f as possible)

Learning Flow



Learning is to find a function

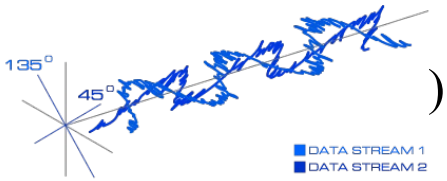
- Speech Recognition

f() = “我不知道你說什麼”

- Image recognition

f() = “Seafood”

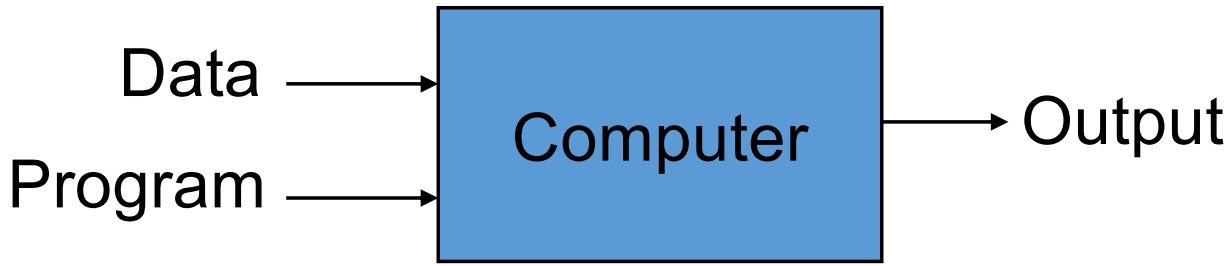
- Channel estimation

f(

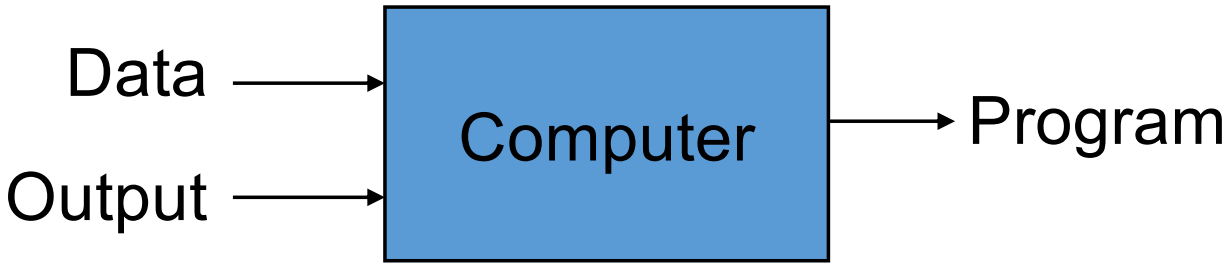
) = Channel parameters

http://cdn1.itpro.co.uk/sites/itpro/files/images/dir_176/it_photo_88225.jpg
http://www.berkeleywellness.com/sites/default/files/field/image/ThinkstockPhotos-520490716_field_img_hero_988_380.jpg
https://telcoantennas.com.au/site/sites/default/files/images/4G-cross-polarisation-low-signal-areas.png

Traditional Programming



Machine Learning



Magic?

No, more like gardening

- **Seeds** = Algorithms
- **Nutrients** = Data
- **Gardener** = You
- **Plants** = Programs

Agenda

- Philosophy
- Machine Learning Concept
- Clustering

Supervised vs. Unsupervised Learning

- Unsupervised learning (clustering)
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data
- Supervised learning (classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the class of the observations
  - New data is classified based on the training set

What is Cluster Analysis?

- **Cluster**: a collection of data objects

Similar to one another within the same cluster

Dissimilar to the objects in other clusters

- **Cluster Analysis**

Grouping a set of data objects into clusters

- Typical applications:

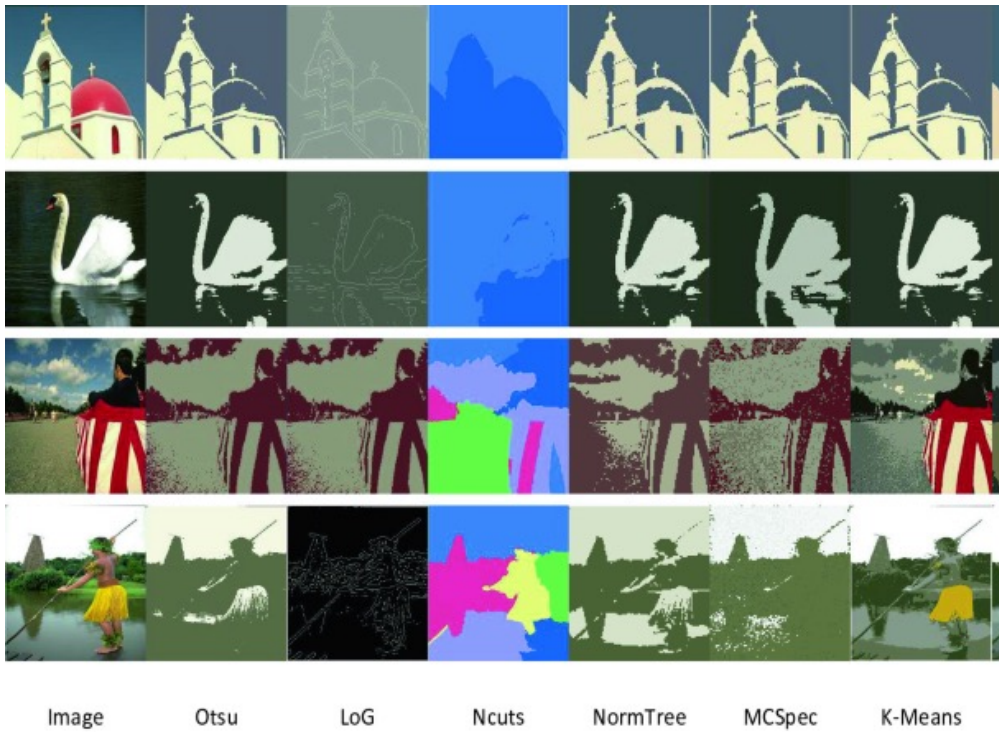
As a stand-alone tool to get insight into data distribution

As a preprocessing step for other algorithms

Unsupervised learning

General Applications of Clustering

- Spatial Data Analysis
  - Detect spatial clusters and explain them in spatial data mining.
- Image Processing
- Pattern Recognition
- Economic Science (especially market research)
- WWW
  - Document classification
  - Cluster Web-log data to discover groups of similar access patterns



https://link.springer.com/chapter/10.1007/978-3-030-55180-3_50

Examples of Clustering Applications

Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs.

Land use: Identification of areas of similar land use in an earth observation database.

Insurance: Identifying groups of motor insurance policy holders with a high average claim cost.

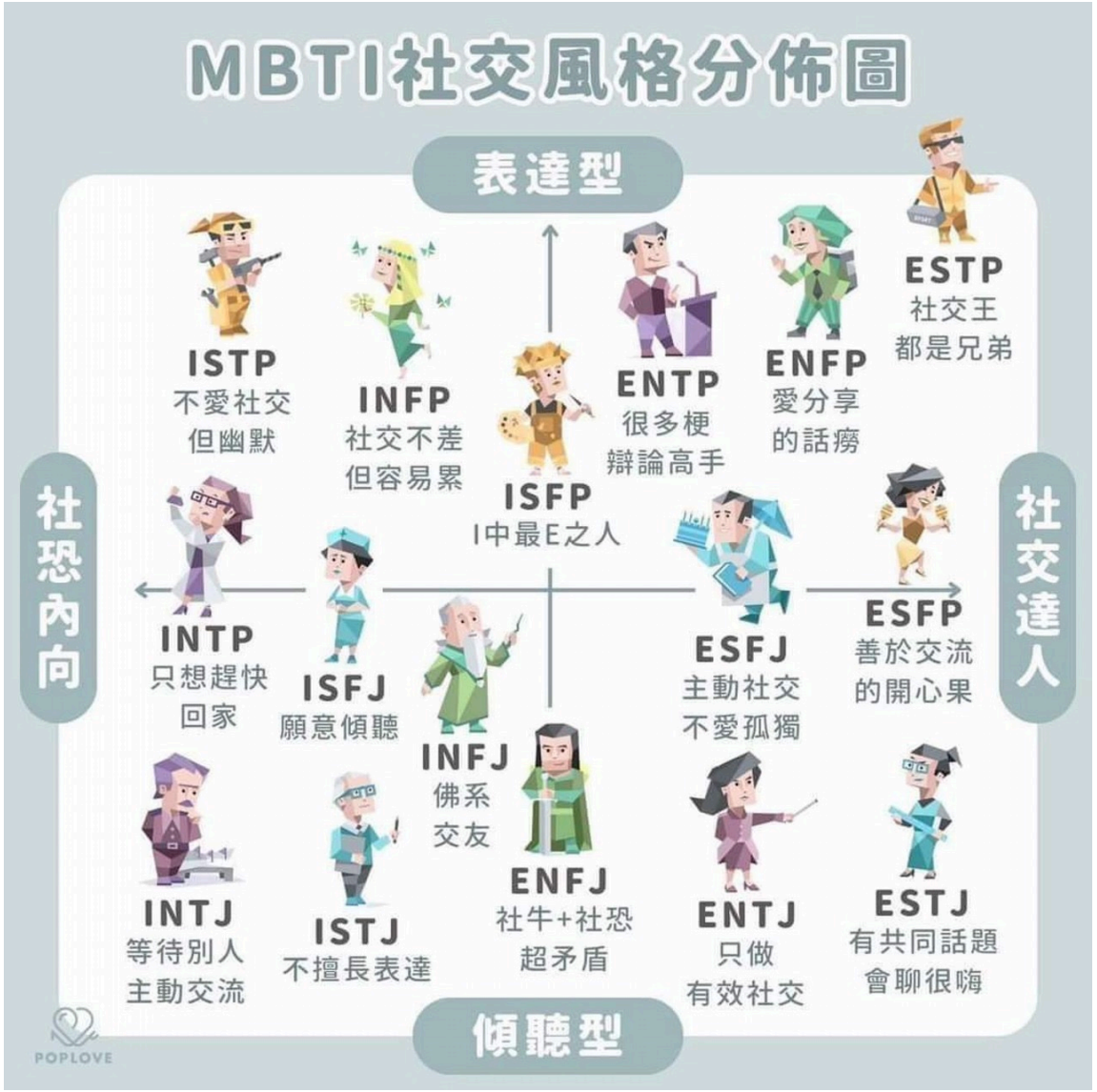
City-planning: Identifying groups of houses according to their house type, value, and geographical location.

[MBTI 人格測試]

MBTI

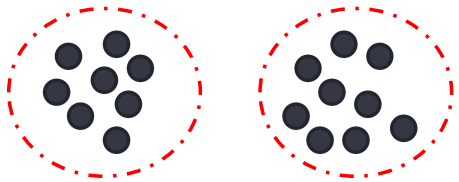
Break

MBTI


MBTI社交風格分佈圖
表達型
傾聽型
社恐內向
社交達人
ISTP
不愛社交
但幽默
INFP
社交不差
但容易累
ISFP
I中最E之人
ENTP
很多梗
辯論高手
ENFP
愛分享
的話癆
ESTP
社交王
都是兄弟
INTP
只想趕快
回家
ISFJ
願意傾聽
INFJ
佛系
交友
ESFJ
主動社交
不愛孤獨
ESFP
善於交流
的開心果
INTJ
等待別人
主動交流
ISTJ
不擅長表達
ENFJ
社牛+社恐
超矛盾
ENTJ
只做
有效社交
ESTJ
有共同話題
會聊很嗨
POPLOVE

What is Good Clustering?

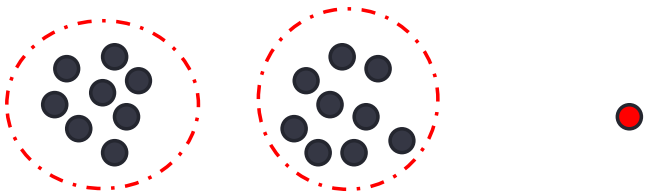
- A good clustering method will produce high quality clusters with
  - High intra-class similarity
  - Low inter-class similarity



- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover hidden patterns.

Requirements of Clustering

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements of domain knowledge for input
- Able to deal with outliers



Requirements of Clustering

- Insensitive to order of input records
- High dimensionality
- Curse of dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Clustering Methods

Partitioning Method

Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors

K-means, k-medoids, CLARANS

Hierarchical Method

Create a hierarchical decomposition of the set of data (or objects) using some criterion

Diana, Agnes, BIRCH, ROCK, CHAMELEON

Density-based Method

Based on connectivity and density functions

Typical methods: DBSACN, OPTICS, DenClue

Calculate the Distance between **Clusters**

- **Single link**: smallest distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = min(t_{ip}, t_{jq})$
- **Complete link**: largest distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = max(t_{ip}, t_{jq})$
- **Average**: average distance between an element in one cluster and an element in the other, i.e., $dis(K_i, K_j) = avg(t_{ip}, t_{jq})$
- **Centroid**: distance between the centroids of two clusters, i.e., $dis(K_i, K_j) = dis(C_i, C_j)$
- **Medoid**: distance between the medoids of two clusters, i.e., $dis(K_i, K_j) = dis(M_i, M_j)$
- Medoid: one chosen, centrally located object in the cluster

Centroid, Radius and Diameter of a Cluster (for numerical data sets)

Centroid: the "middle" of a cluste

$$C_m = \frac{\sum_{i=1}^{N} (t_{ip})}{N}$$

Radius: square root of average mean squared distance from any point of the cluster to its centroid

$$R_m = \sqrt{\frac{\sum_{i=1}^{N} (t_{ip} - c_m)^2}{N}}$$

Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D_m = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} (t_{ip} - t_{jq})^2}{N(N-1)}}$$

diameter != 2 * radius

Partitioning Algorithms: Basic Concept

- Partitioning method: construct a partition of a database D of **n** objects into a set of **k** clusters.
  Given a number k, find a partition of k clusters that optimizes the chosen partitioning criterion.
- Global optimal: exhaustively enumerate all partitions.
- Heuristic methods: k-means, k-medoids

  k-means (MacQueen'67)
  k-medoids or PAM, partition around medoids (Kaufman & Rousseeuw'87)

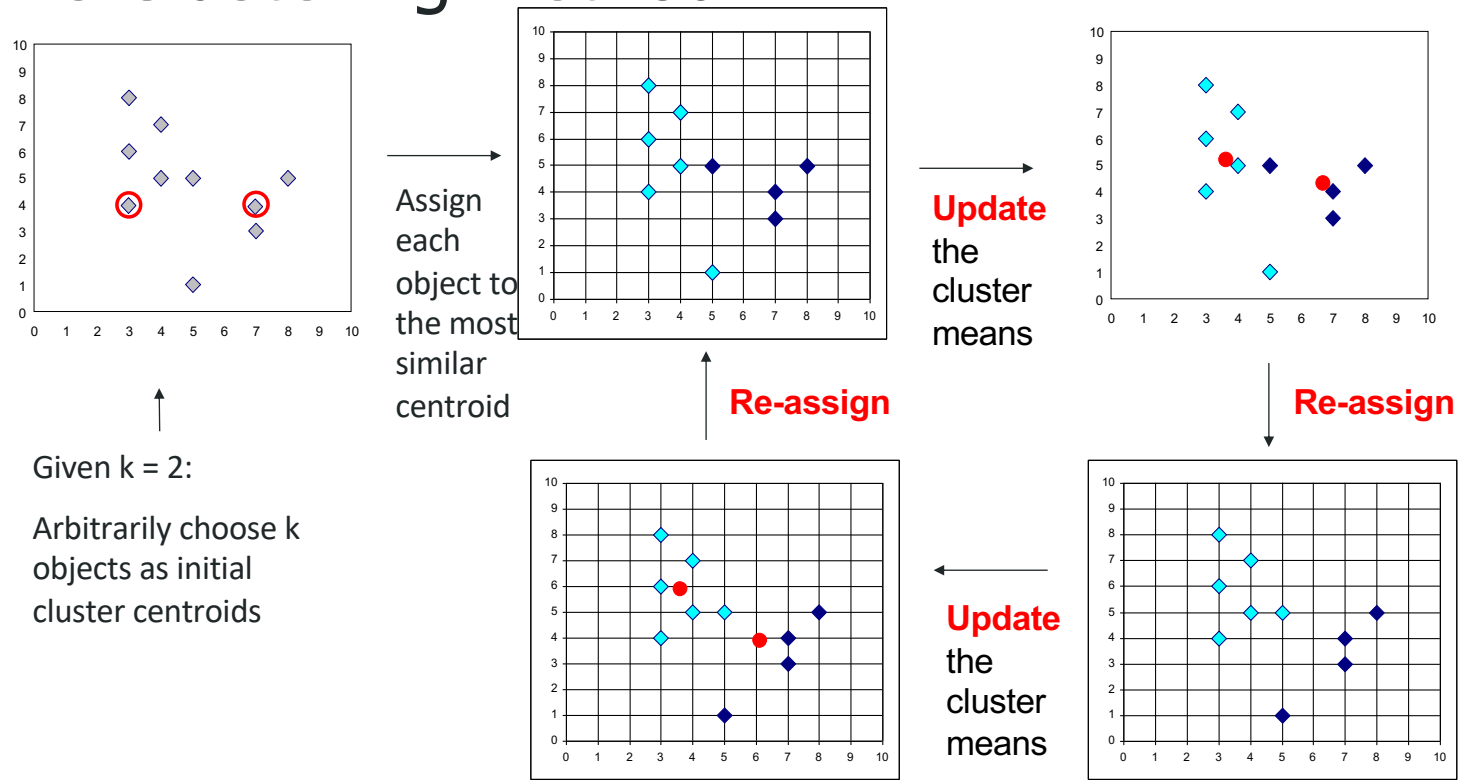
The K-Means Clustering Method

Given k, the k-means algorithm is implemented in four steps:

1. Arbitrarily choose k points as initial cluster centroids.
2. **Update Means (Centroids)**: Compute seed points as the center of the clusters of the current partition. (center: mean point of the cluster)
3. **Re-assign Points**: Assign each object to the cluster with the nearest seed point.
4. Go back to Step 2, stop when no more new assignment.

loop (steps 2–3)

Example of the K-Means Clustering Method



Comments on the K-Means Clustering

Time Complexity: O(tkn), where n is # of objects, k is # of clusters, and t is # of iterations. Normally, k,t<<n.

Often terminates at a local optimum.

(The global optimum may be found using techniques such as: deterministic annealing and genetic algorithms)

Weakness:

Applicable only when mean is defined, how about categorical data?

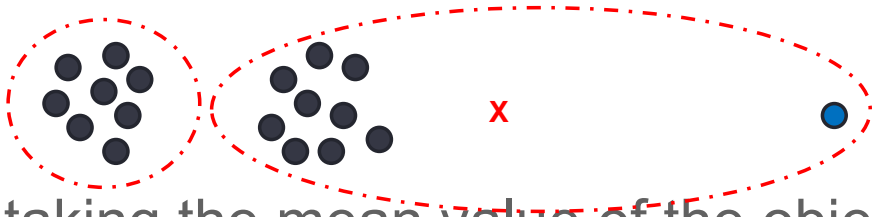
Need to specify k, the number of clusters, in advance

Unable to handle noisy data and outliers

Why is K-Means Unable to Handle Outliers?

The k-means algorithm is sensitive to outliers

Since an object with an extremely large value may substantially distort the distribution of the data.



K-Medoids: Instead of taking the mean value of the object in a cluster as a reference point, medoids can be used, which is the most centrally located object in a cluster.

PAM: The K-Medoids Method

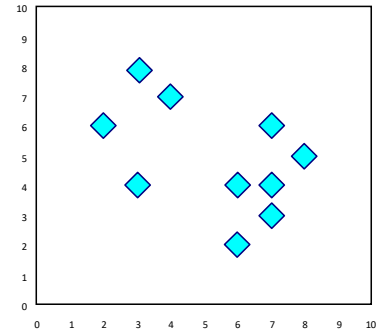
PAM: Partition Around Medoids

Use real object to represent the cluster

1. Randomly select ***k*** representative objects as medoids.
2. Assign each data point to the closest medoid.
3. For each medoid *m*,
   a. For each non-medoid data point o
   b. Swap *m* and o, and compute the total cost of the configuration.
4. Select the configuration with the lowest cost.
5. Repeat steps 2 to 5 until there is no change in the medoid.

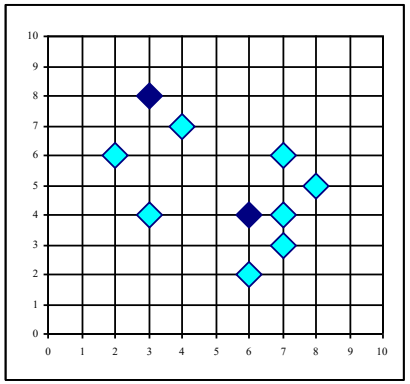
loop

A Typical K-Medoids Algorithm (PAM)

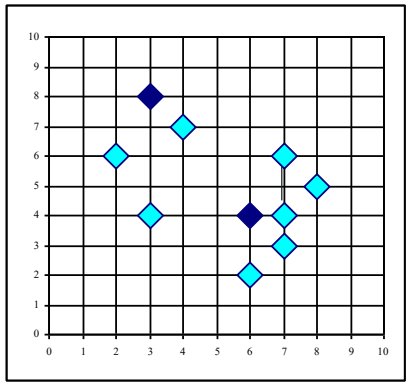


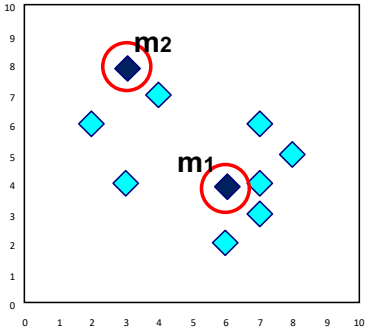
k = 2

Arbitrary choose k object as initial medoids



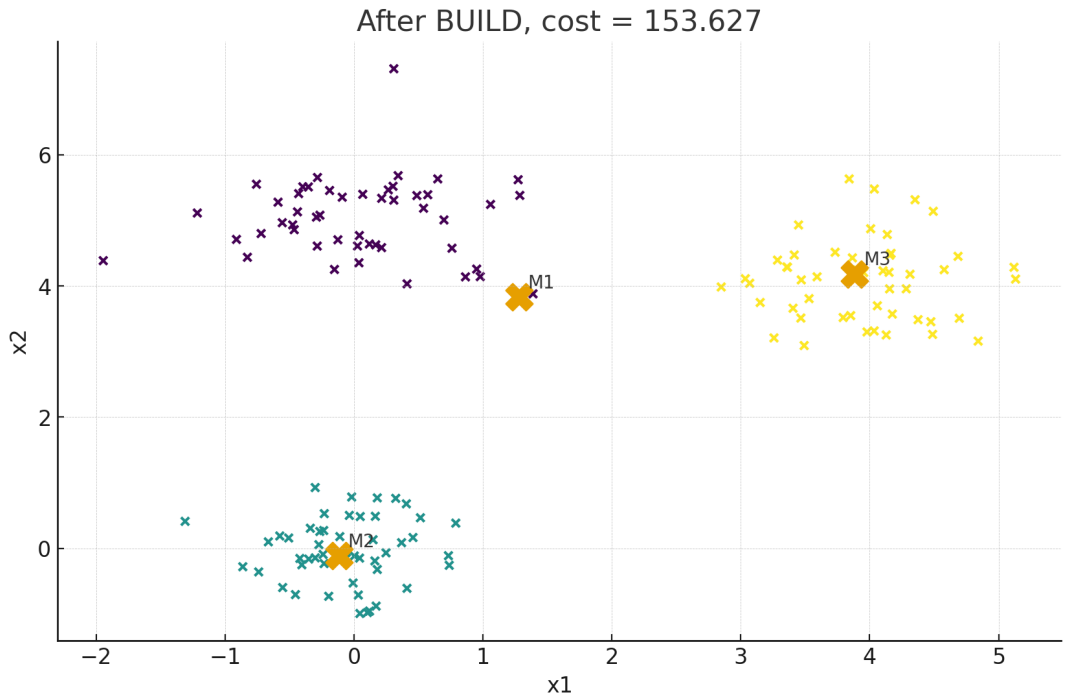
Assign each remaining object to the nearest medoid





Swap each medoid and each data point, and compute the total cost of the configuration


Dataset
x2
x1
6
4
2
0
−2
−1
0
1
2
3
4
5


After BUILD, cost = 153.627
x2
x1
M1
M2
M3
6
4
2
0
−2
−1
0
1
2
3
4
5


After first SWAP, cost = 105.601
M1
M2
M3
x1
x2
−2
−1
0
1
2
3
4
5
0
2
4
6


Final clustering, cost = 105.601
M1
M2
M3
x1
x2
−2
−1
0
1
2
3
4
5
0
2
4
6

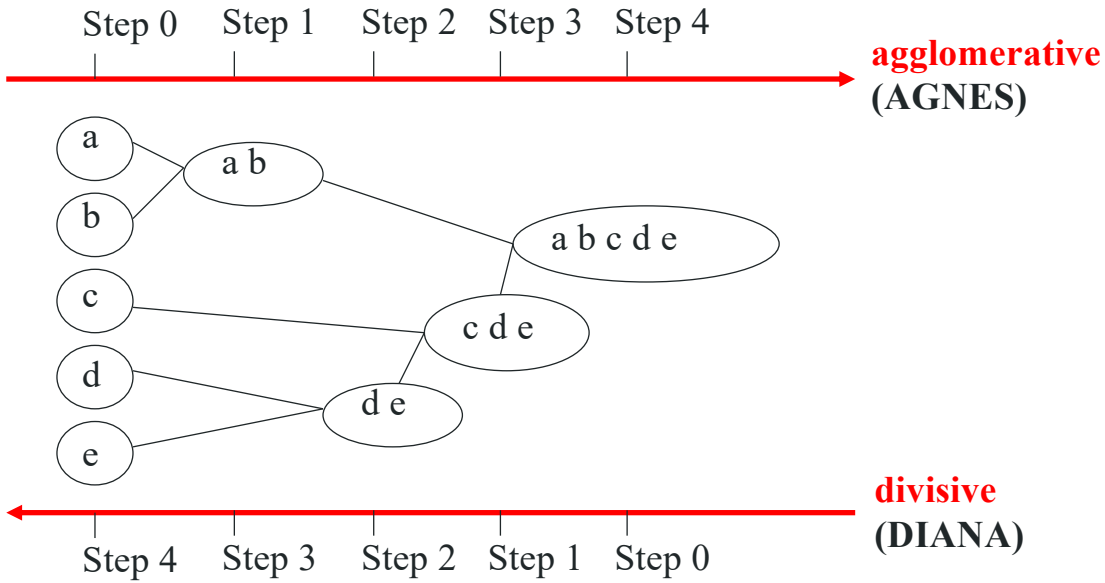
What is the Problem with PAM?

- PAM is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean.
- PAM works efficiently for small data sets but does not scale well for large data sets.
- **O(k(n-k)(n-k))** for each iteration, where n is # of data, k is # of clusters
- Improvements: CLARA (uses a sampled set to determine medoids), CLARANS

Hierarchical Clustering

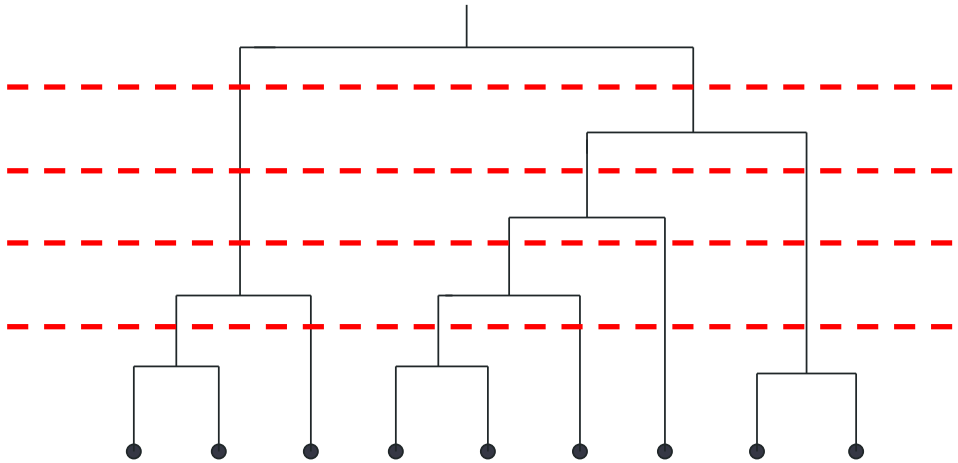
Use distance matrix as clustering criteria.

This method does not require the number of clusters ***k*** as an input, but needs a termination condition.



Dendrogram:Shows How the Clusters are Merged

- Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.
- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.



More on Hierarchical Clustering

Major weakness:

Do not scale well: time complexity is at least $O(n^2)$, where n is the number of total objects.

Can never undo what was done previously.

Integration of hierarchical with distance-based clustering

BIRCH(1996): uses CF-tree data structure and incrementally adjusts the quality of sub-clusters.

CURE(1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction.

Density-Based Clustering Methods

Clustering based on **density** (**local cluster criterion**), such as density-connected points

Major features:

Discover clusters of arbitrary shape

Handle noise

One scan

Need density parameters as termination condition

Several interesting studies:

DBSCAN: Ester, et al. (KDD'96)

OPTICS: Ankerst, et al (SIGMOD'99).

DENCLUE: Hinneburg & D. Keim (KDD'98)

CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

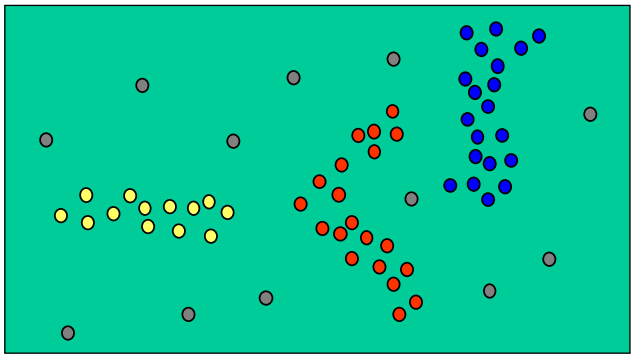
DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Proposed by Ester, Kriegel, Sander, and Xu (KDD96)
- Relies on a density-based notion of cluster: A cluster is defined as a maximal set of density-connected points.
- Discovers clusters of arbitrary shape in spatial databases with noise

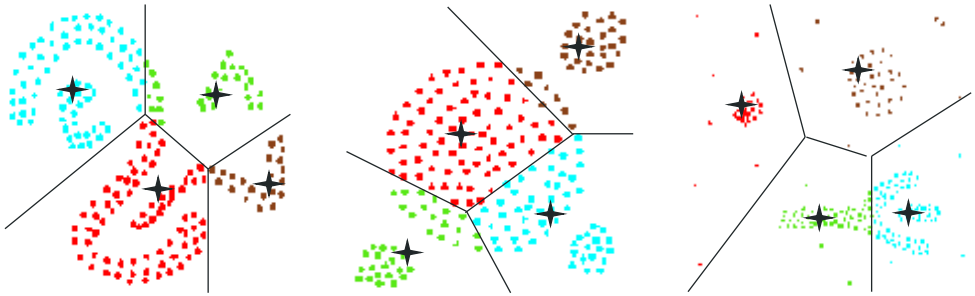
Density-Based Clustering

Basic Idea:

Clusters are dense regions in the data space, separated by regions of lower object density



Why Density-Based Clustering?



Results of a *k*-medoid algorithm for *k*=4

Different density-based approaches exist (see Textbook & Papers)
Here we discuss the ideas underlying the DBSCAN algorithm

Density Based Clustering: Basic Concept

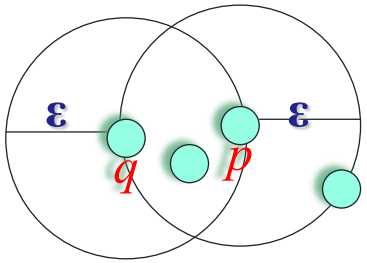
- Intuition for the formalization of the basic idea
- For any point in a cluster, the local point density around that point has to exceed some threshold
- The set of points from one cluster is spatially connected
- Local point density at a point *p* defined by two parameters
- ε – radius for the neighborhood of point p:
  $N_{\varepsilon}(p) := \{q \text{ in data set } D \mid dist(p, q) \leq \varepsilon\}$
- *MinPts* – minimum number of points in the given neighbourhood $N(p)$

ε-Neighborhood

■ ε-Neighborhood – Objects within a radius of ε from an object.

$$N_\varepsilon(p) : \{q \mid d(p,q) \le \varepsilon\}$$

■"High density" - ε-Neighborhood of an object contains at least *MinPts* of objects.



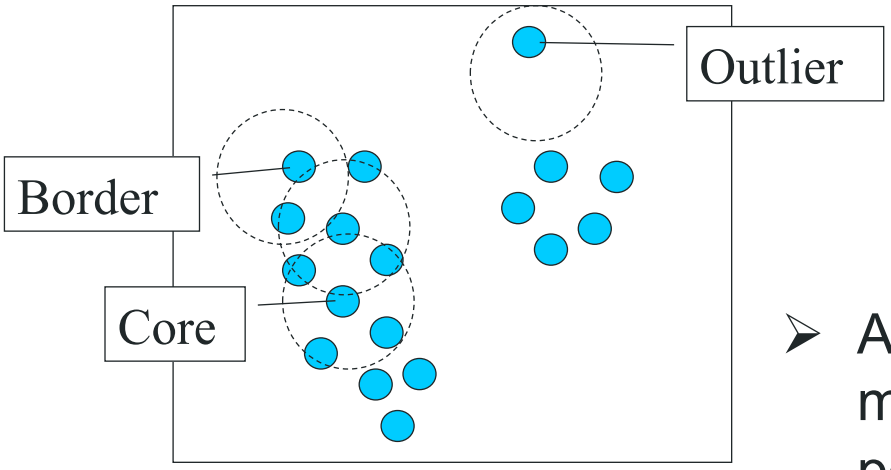
ε-Neighborhood of *p*

ε-Neighborhood of *q*

Density of p is "high" (MinPts = 4)

Density of q is "low" (MinPts = 4)

Core, Border & Outlier



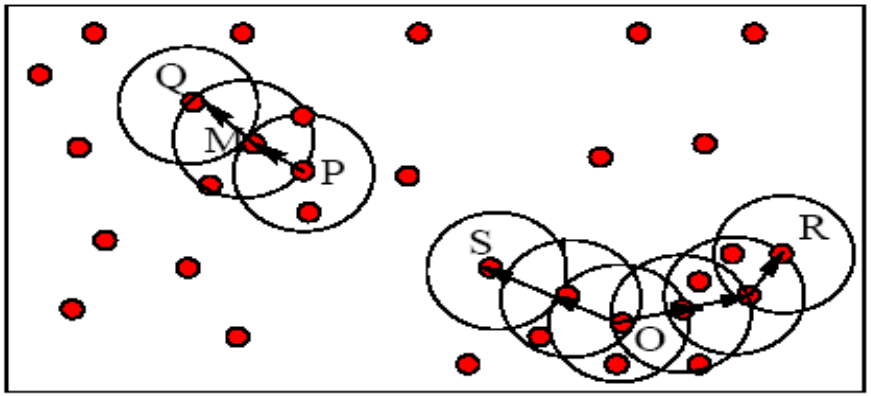
ε = 1unit, MinPts = 5

Given ε and *MinPts*, categorize the objects into three exclusive groups.

- A point is a core point if it has more than a specified number of points (MinPts) within Eps These are points that are at the interior of a cluster.
- A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.
- A noise point is any point that is not a core point nor a border point.

Example

M, P, O, and R are core objects since each is in an Eps neighborhood containing at least 3 points

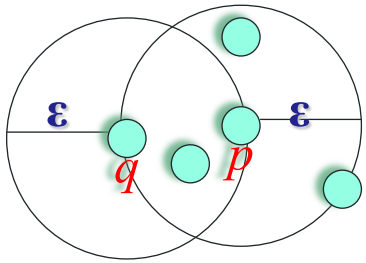


Minpts = 3

Eps=radius of the circles

Density-Reachability

- **Directly density-reachable**
  - **An object q is directly density-reachable from object p if p is a core object and q is in p's ε-neighborhood.**

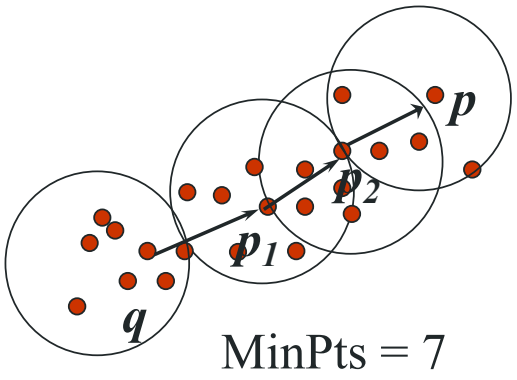


MinPts = 4

- **q is directly density-reachable from p**
- **p is not directly density- reachable from q?**
- **Density-reachability is asymmetric.**

Density-reachability

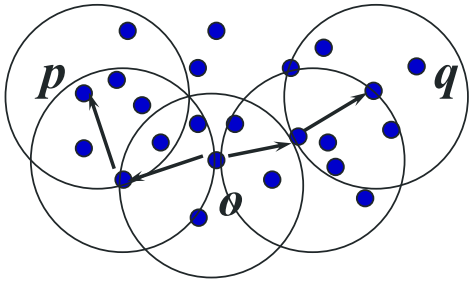
- Density-Reachable (directly and indirectly):
- A point p is directly density-reachable from p2;
- p2 is directly density-reachable from p1;
- p1 is directly density-reachable from q;
- p←p2←p1←q form a chain.



- **p is (indirectly) density-reachable from q**
- **q is not density- reachable from p?**

Density-Connectivity

- **Density-reachable is not symmetric**
  - **not good enough to describe clusters**
- **Density-Connected**
  - **A pair of points p and q are density-connected if they are commonly density-reachable from a point o.**



- **Density-connectivity is symmetric**

Formal Description of Cluster

Given a data set D, parameter ε and threshold MinPts.

A cluster C is a subset of objects satisfying two criteria:

- *Connected:* for all p,q in C: p and q are density-connected.
- *Maximal:* for p,q: if p in C and q is density-reachable from p, then q inC. (avoid redundancy)

↓

P is a core object.

DBSCAN Algorithm


```
Input: The data set D

Parameter: ε, MinPts

For each object p in D
    if p is a core object and not processed then
        C = retrieve all objects density-reachable from p
        mark all objects in C as processed
        report C as a cluster
    else mark p as outlier
    end if

End For
```

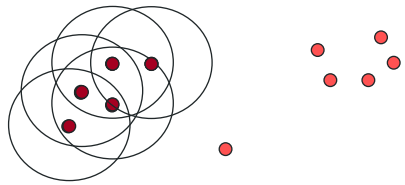
DBScan Algorithm

DBSCAN Algorithm: Example

Parameter

ε = 2 cm

MinPts = 3



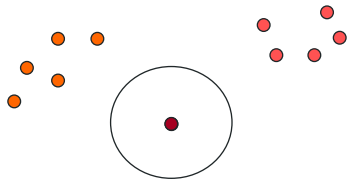
```
for each o ∈ D do
    if o is not yet classified then
        if o is a core-object then
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE
```

DBSCAN Algorithm: Example

Parameter

ε = 2 cm

MinPts = 3



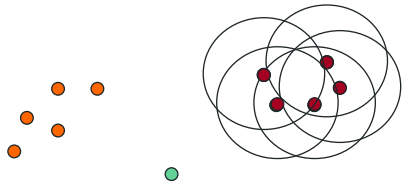
```
for each o ∈ D do
    if o is not yet classified then
        if o is a core-object then
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE
```

DBSCAN Algorithm: Example

Parameter

ε = 2 cm

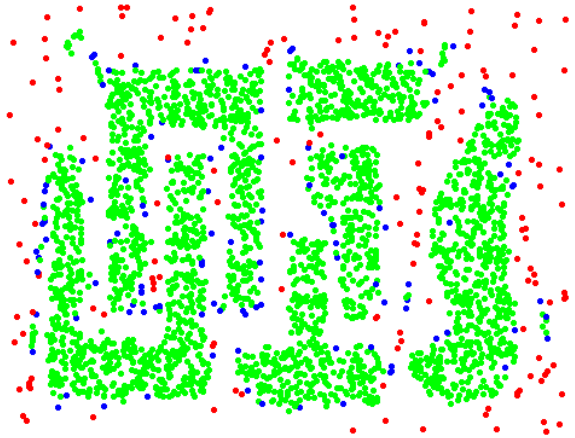
MinPts = 3



for each $o \in D$ **do**
    if o is not yet classified **then**
        if o is a core-object **then**
            collect all objects density-reachable from o
            and assign them to a new cluster.
        else
            assign o to NOISE

Example

Original Points

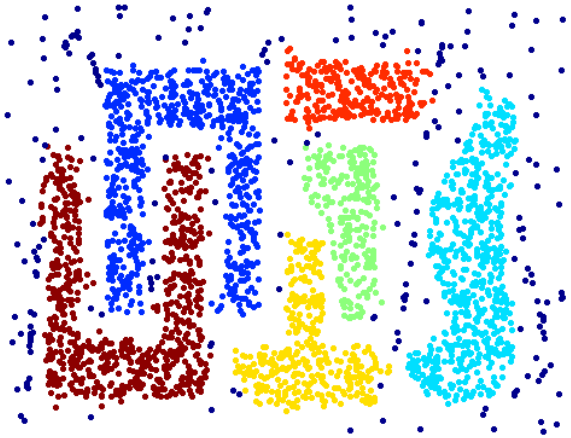


Point types: core, border and outliers

ε = 10, MinPts = 4

When DBSCAN Works Well

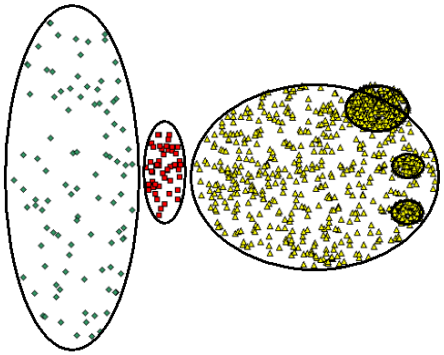
Original Points



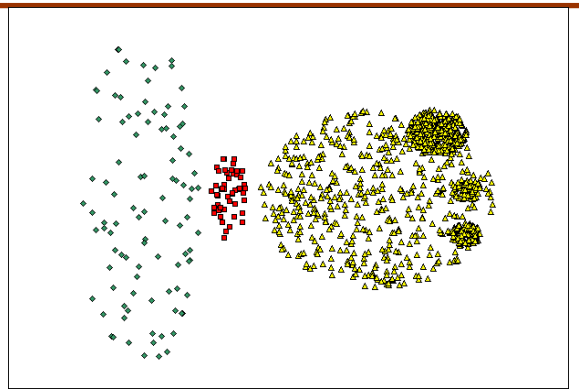
Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

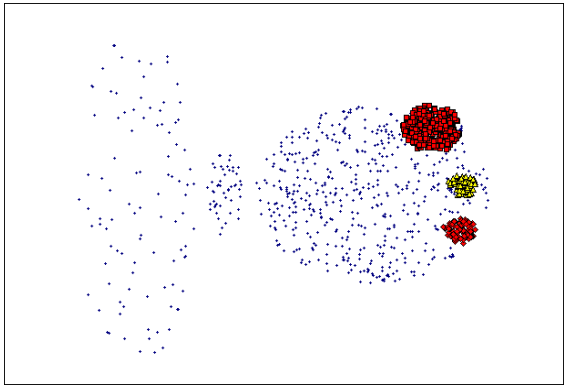
When DBSCAN Does NOT Work Well



Original Points



(MinPts=4, Eps=9.92).



(MinPts=4, Eps=9.75)

- **Cannot handle Varying densities**
- **sensitive to parameters**

K-MEANS AND DBSCAN in Scikit-Learn

首先先引入本教學會用到的套件

In [1]:
```
# Start from importing necessary packages.
import warnings
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm as cm

from IPython.display import display
from sklearn import metrics # for evaluations
from sklearn.datasets import make_blobs, make_circles # for generating experimental data
from sklearn.preprocessing import StandardScaler # for feature scaling
from sklearn.cluster import KMeans
from sklearn.cluster import DBSCAN

# make matplotlib plot inline (Only in Ipython).
warnings.filterwarnings('ignore')
%matplotlib inline
```

產生含有三個 clusters 的二維資料

In [2]:
```
# Generate data.
# `random_state` is the seed used by random number generator for reproducibility (default=None).
X, y = make_blobs(n_samples=5000,
                  n_features=2,
                  centers=3,
                  random_state=170)

# Print data like Ipython's cell output (Only in Ipython, otherwise use `print`).
display(X)
display(y)
```

```
array([[-4.01009423, -1.01473496],
       [ 1.00550526,  0.13163222],
       [ 2.06563121, -0.24527689],
       ...,
       [-5.09493013,  1.47160372],
       [-9.61459714, -4.91848716],
       [-7.72675795, -5.86656563]])
```

← 這是我們的訓練資料 X，共 5000 筆，每筆資料是二維，因此 X 是 (5000 x 2) 的矩陣。

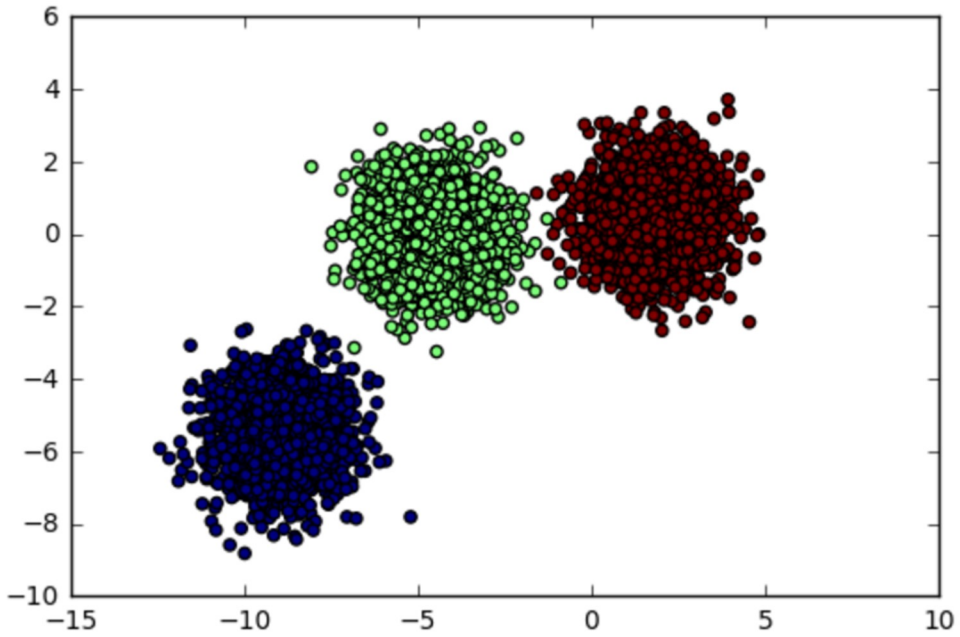
```
array([1, 2, 2, ..., 1, 0, 0])
```

← 這是我們每筆資料對應的 cluster label y (ground truth)，在此例有三個 clusters，因此 label = {0, 1, 2}，共 5000 筆，因此 y 是 5000 維的向量。

讓我們把資料視覺化一下，看看長什麼樣子。

```
In [3]: # Plot the data distribution (ground truth) using matplotlib `scatter(axis-x, axis-y, color)`.
        plt.scatter(X[:,0], X[:,1], c=y)
```

```
Out[3]: <matplotlib.collections.PathCollection at 0x111d4c310>
```



In [4]:
```
# Perform K-means on our data (Train centroids)
kmeans = KMeans(n_clusters=3,
                n_init=3,
                init='random',
                tol=1e-4,
                random_state=170,
                verbose=True).fit(X)
```

```
Initialization complete
Iteration  0, inertia 93044.046
Iteration  1, inertia 11941.892
Iteration  2, inertia 9733.271
Converged at iteration 2
Initialization complete
Iteration  0, inertia 49491.222
Iteration  1, inertia 43229.847
Iteration  2, inertia 42926.507
Iteration  3, inertia 42574.503
Iteration  4, inertia 41999.156
Iteration  5, inertia 39853.967
Iteration  6, inertia 28163.261
Iteration  7, inertia 11338.225
Iteration  8, inertia 9733.783
Converged at iteration 8
Initialization complete
Iteration  0, inertia 35757.303
Iteration  1, inertia 10003.824
Iteration  2, inertia 9733.154
Converged at iteration 2
```

使用 scikit-learn 所提供的 KMeans 來跑我們的資料。

參數解釋

- n_clusters：你想在這資料上分幾群，也就是「選擇 “k”」啦。
- n_inits：你想讓 scikit-learn 的 k-means 演算法「跑幾次」，這邊我們是設定成 3，意思就是它會跑 3 次的 k-means，並且選擇最低 inertia 的那一次 (這邊的 inertia 目標就是 minimize sum of squared distance，distance 越小當然就越好囉！)，當作 k-means 的訓練完後的結果，也就是留著最好的 centroids。
- init：還記得 k-means 演算法一開始會隨機選擇 centroids 嗎？這邊就是設定「怎麼選擇 centroids 的演算法」，有 “random” 跟 “k-means++” (後面會解釋) 可以選擇。
- tol：k-means 收斂時距離門檻值的比例，比例越高會越早視為收斂結束。
- random_state：給一固定值可以讓我們一直跑出同樣的結果，通常 debug 才會這麼用，一般來說不需要。
- verbose：設定 True 則會 print 出訓練 k-means 時的過程，反之則 False。

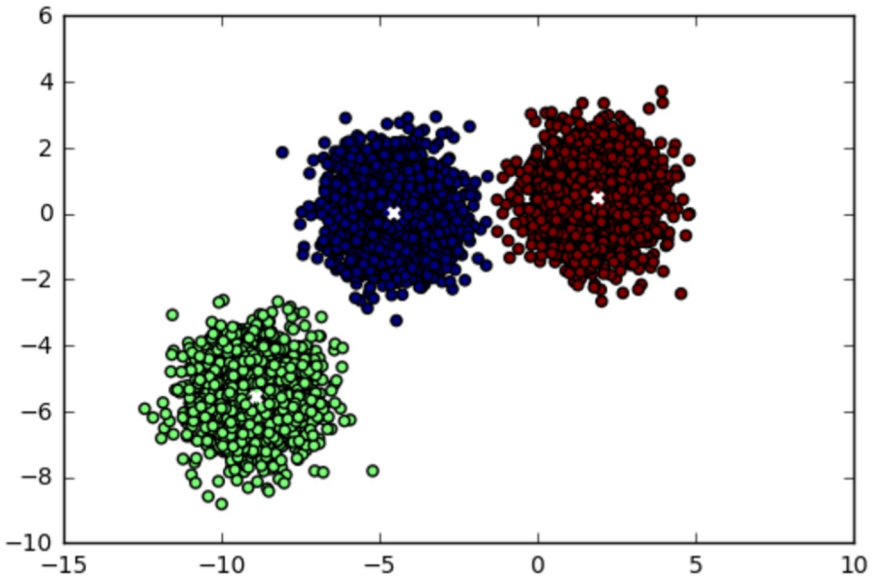
```
In [5]: # Retrieve predictions and cluster centers (centroids).
        display(kmeans.labels_)
        display(kmeans.cluster_centers_)
```

```
array([0, 2, 2, ..., 0, 1, 1], dtype=int32)

array([[-4.55676387,  0.04603707],
       [-8.94710203, -5.51613184],
       [ 1.89450492,  0.5009336 ]])
```

← 使用 “kmeans.labels_” 來取得我們預測的結果，同樣是 5000 維的向量。
← 使用 “kmeans.cluster_centers_” 來取得訓練出來的 centroids，有三個 clusters，資料為二維，因此是 (3x2) 的矩陣。

```
In [6]: # Plot the predictions.
        plt.scatter(X[:,0], X[:,1], c=kmeans.labels_)
        plt.scatter(kmeans.cluster_centers_[:,0],
                    kmeans.cluster_centers_[:,1],
                    c='w', marker='x', linewidths=2)
```

```
Out[6]: <matplotlib.collections.PathCollection at 0x1125648d0>
```



← 把預測的結果視覺化出來，並且也順便把 centroids 給畫上去。

注意：每次預測出來的 cluster 雖然跟 ground truth 畫的 cluster 不同，但這只是因為 k-means 隨機選擇 centroids 的關係，所以有可能每次跑出來的結果中，同樣的一群資料被分群到的 cluster label 會不同，但是重點是看整體的分群結果喔！顏色不同其實也沒關係。

In [7]:
```
# We can make new predictions without re-run kmeans (simpily find nearest centroids).
X_new = np.array([[10,10], [-10, -10], [-5, 10]])
y_pred = kmeans.predict(X_new)

""" The below code is equivalent to:
y_pred = KMeans(...).fit_predict(X), but this needs to fit kmeans again.
"""

display(y_pred)
```

```
array([2, 1, 0], dtype=int32)
```

← 訓練完 k-means 之後，使用 “kmeans.predict()” 來預測新資料 X_new。

In [8]:
```
# We can get distances from data point to every centroid

""" The below code is equivalent to:
from sklearn.metrics.pairwise import euclidean_distances
euclidean_distances(X_new, kmeans.cluster_centers_)
"""

kmeans.transform(X_new)
```

Out[8]:
```
array([[ 17.63464636,  24.48965134,  12.48724601],
       [ 11.42592142,   4.60582976,  15.86659553],
       [  9.96382639,  16.01030799,  11.73739582]])
```

← 使用 “kmeans.transform()” 取得 X_new 中每個資料點到 每個 centroid 的距離。如果你有 M 個資料點，N 個 clusters，則你會有 (MxN) 的 矩陣，每個 row 對應至一個資料點到 N 個 clusters 的距離。

K-means++

K-means++ 演算法，更聰明初始化 centroid 的方式

A Smarter Way to Initialize Centroids: K-means++

Since *K-means* highly depends on the initialization of the centroids, the clustering results may be converged to a local minimum. We can address this by setting `init='kmeans++'` instead of `'random'`. *K-means*++ initializes centroids in a smarter way to speed up convergence. The algorithm is as follows:

1. Randomly choose one centroid from the data points.
2. For each data point x_i, compute the distance $D(x_i, c_j)$ where c_j is nearest to x_i.
3. Randomly choose one new data point as a new centroid using *weighted probability distribution* proportional to $D(x_i, c_j)^2$.
4. Repeat steps 2 and 3 until k centroids have been chosen.
5. Now we have initialized centroids, run *K-means* algorithm.

簡單來說，kmeans++ 跟 kmeans 的差別是：
主要希望選到的 centroids 它們各自的距離越大越好，避免選到離彼此之間很近的 centroids。

In [9]:
```
# Perform K-means++ on our data.
kmeans_plus_plus = KMeans(n_clusters=3,
                          n_init=3,
                          init='k-means++',
                          tol=1e-4,
                          random_state=170,
                          verbose=True).fit(X)
```

```
Initialization complete
Iteration  0, inertia 14504.164
Iteration  1, inertia 9735.745
Iteration  2, inertia 9733.168
Converged at iteration 2
Initialization complete
Iteration  0, inertia 10751.002
Iteration  1, inertia 9733.462
Converged at iteration 1
Initialization complete
Iteration  0, inertia 12316.398
Iteration  1, inertia 9733.520
Converged at iteration 1
```

You can see that *K-means++* converges much faster than *K-means*!

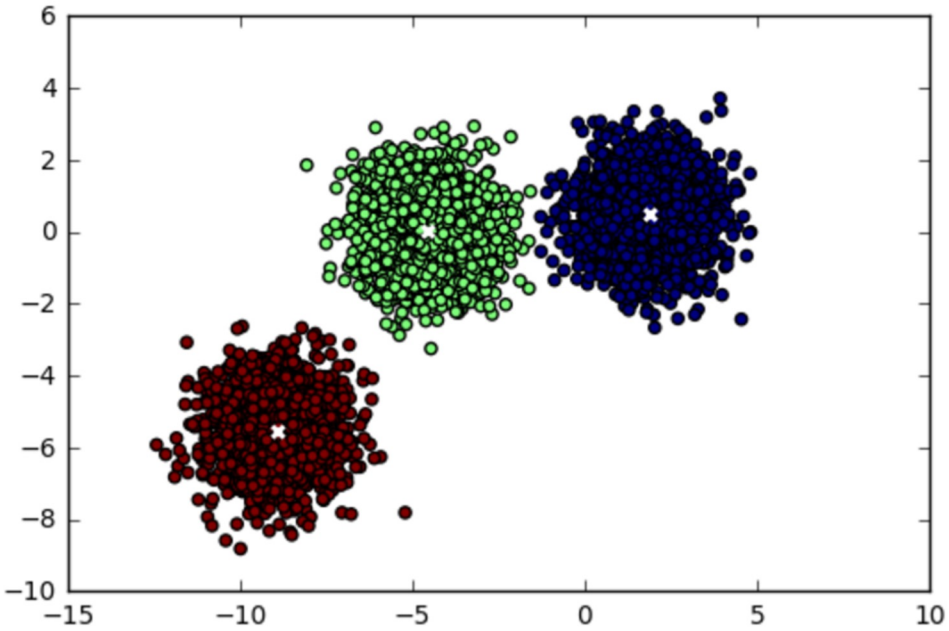
想使用 k-means++，你只要設定 init='k-means++' 就可以囉！很簡單吧？而且有沒有發現 k-means 每次在訓練的時候，收斂速度變快了呢？這要歸功於 k-means++ 一開始選了很好的 centroids，導致後面在 clustering 的時候就會很快！

在 scikit-learn 中，你其實也不需要手動設定 init='k-means++'，因為它默認值就是如此囉！

K-means++ 視覺化的預測結果

In [10]:
```
# Plot the predictions.
plt.scatter(X[:,0], X[:,1], c=kmeans_plus_plus.labels_)
plt.scatter(kmeans_plus_plus.cluster_centers_[:,0],
            kmeans_plus_plus.cluster_centers_[:,1],
            c='w', marker='x', linewidths=2)
```

Out[10]: <matplotlib.collections.PathCollection at 0x1127a4250>

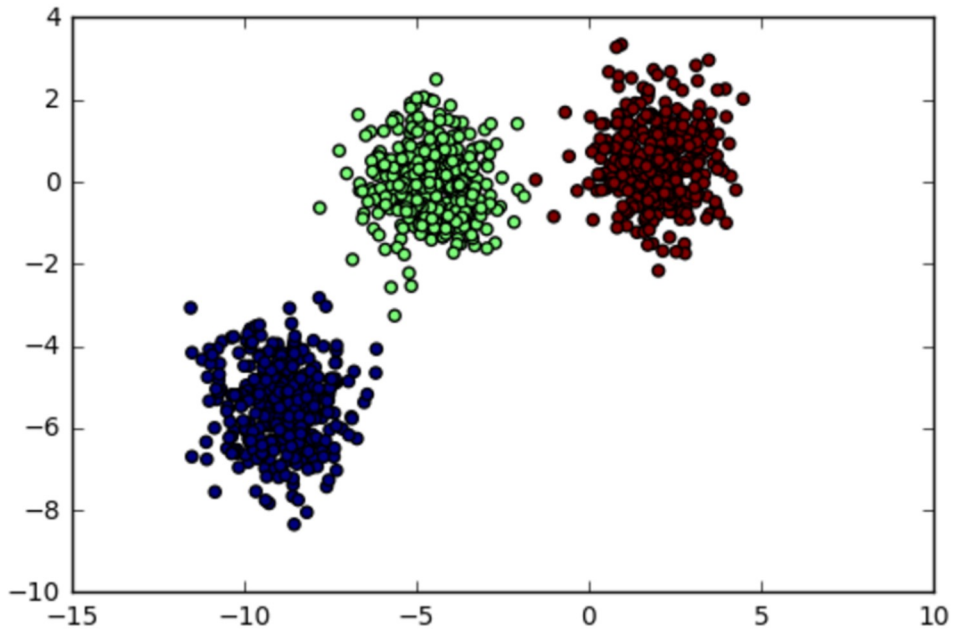


弱點一：需要選擇正確的 “k”

Drawback 1: Need to choose a right number of clusters

In [11]:
```
# Generate data.
X, y = make_blobs(n_samples=1000,
                  n_features=2,
                  centers=3,
                  random_state=170)

# Plot the data distribution.
plt.scatter(X[:,0], X[:,1], c=y)
```

Out[11]: <matplotlib.collections.PathCollection at 0x1129adc90>

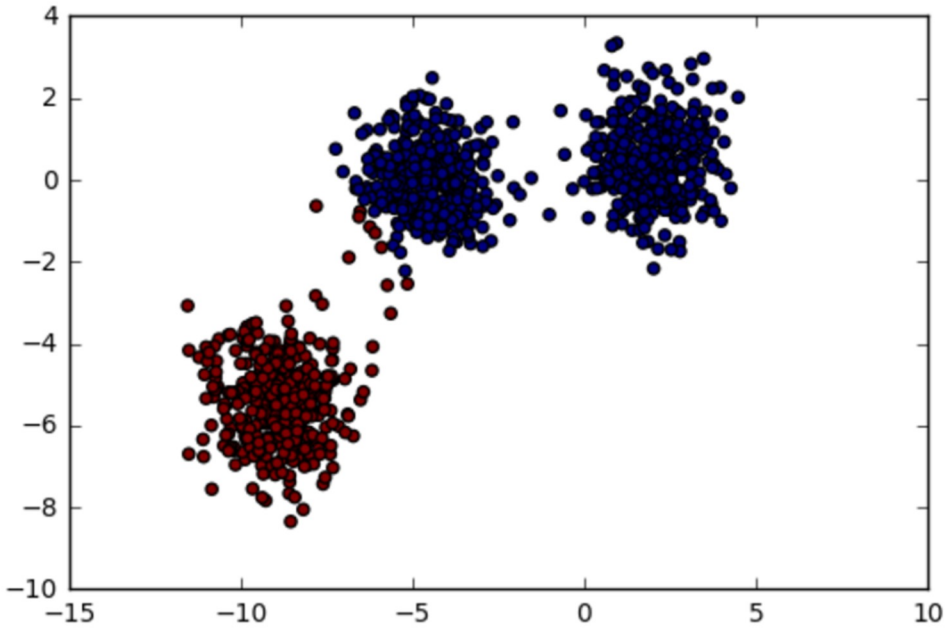


首先，我們一樣先產生三群資料，並且視覺化 ground truth 來看看。

如果我們選擇錯誤的 “k”，會變得怎麼樣呢？

```
In [12]: # Run k-means on non-spherical data.
         y_pred = KMeans(n_clusters=2, random_state=170).fit_predict(X)

         # Plot the predictions.
         plt.scatter(X[:,0], X[:,1], c=y_pred)
```

```
Out[12]: <matplotlib.collections.PathCollection at 0x112b5bc90>
```



答案是：k 選錯，還是可以 work！

如果我們選擇 k=2，結果還是可以分群，你可以看到中間的 cluster 被歸類成藍色，是因為與右邊的 cluster 比較近的關係。

雖然能 work，但與 ground truth 明顯不符，因此未來如果要預測新資料，也許會表現得不好。

那 … 我們要怎麼選擇一個好的 k 呢？

Solution: Measuring Cluster Quality to Determine the Number of Clusters

Supervised method

Homogeneity: Each cluster contains only members of a single class.

Completeness: All members of a given class are assigned to the same cluster.

如果你的訓練資料中已經含有 ground truth，也就是知道哪一筆資料點是屬於哪一個 cluster，那你可以使用 “supervised” 的方式來評估 clusters 的好壞！主要有兩種指標：

- Homogeneity：每個 cluster 中的資料點最好都要是同一個 label。
- Completeness：每個資料點所被預測出來的 cluster label 最好跟 ground truth 一樣（猜對）。

What if we don‘t have ground truth?

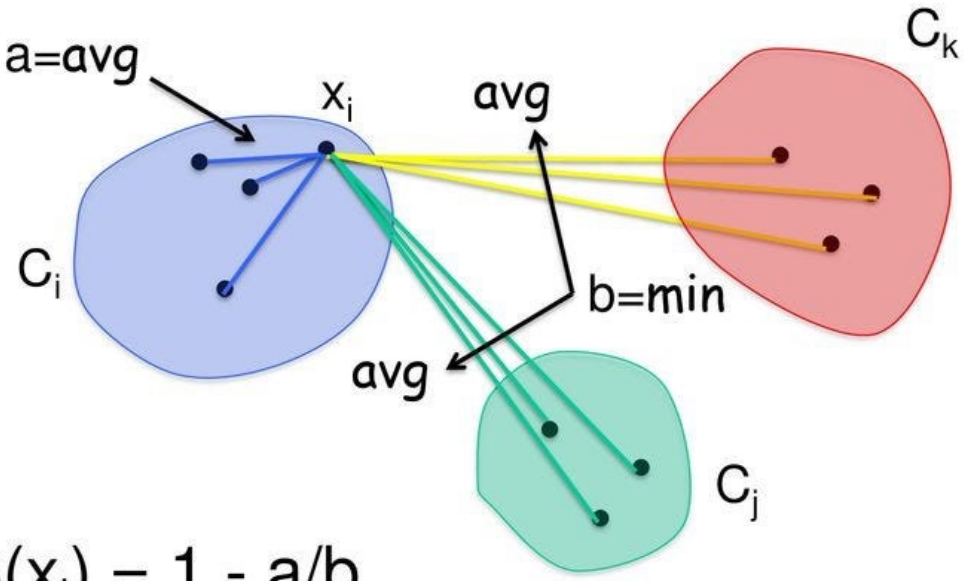
Unsupervised method

Silhouette coefficient: a value for interpretation and validation of consistency within clusters of data, which provides a succinct graphical representation of how well each object lies within its cluster.

The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

Silhouette Coefficient

- The idea...



- Usually, $S(x_i) = 1 - a/b$

Silhouette coefficient

For each datum **i**, let **a(i)** be the average distance between **i** and all other data within the same cluster.

Let **b(i)** be the smallest average distance of **i** to all points in any other cluster, of which **i** is not a member.

We now define a silhouette:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Unsupervised method

Sihouette Coefficient: Evaluate how well the **compactness** and the **separation** of the clusters are. (Note that the notation below is consistent with the above content.) Using *Sihouette Coefficient*, we can choose an optimal value for number of clusters.

$a(x_i)$ denotes the **mean intra-cluster distance**. Evaluate the compactness of the cluster to which x_i belongs. (The smaller the more compact)

$$a(x_i) = \frac{\sum_{x_k \in C_j,\, k \neq i} D(x_i, x_k)}{|C_j| - 1}$$

For the data point x_i, caculate its average distance to all the other data points in its cluster. (Minusing one in denominator part is to leave out the current data point x_i)

$b(x_i)$ denotes the **mean nearest-cluster distance**. Evaluate how x_i is separated from other clusters. (The larger the more separated)

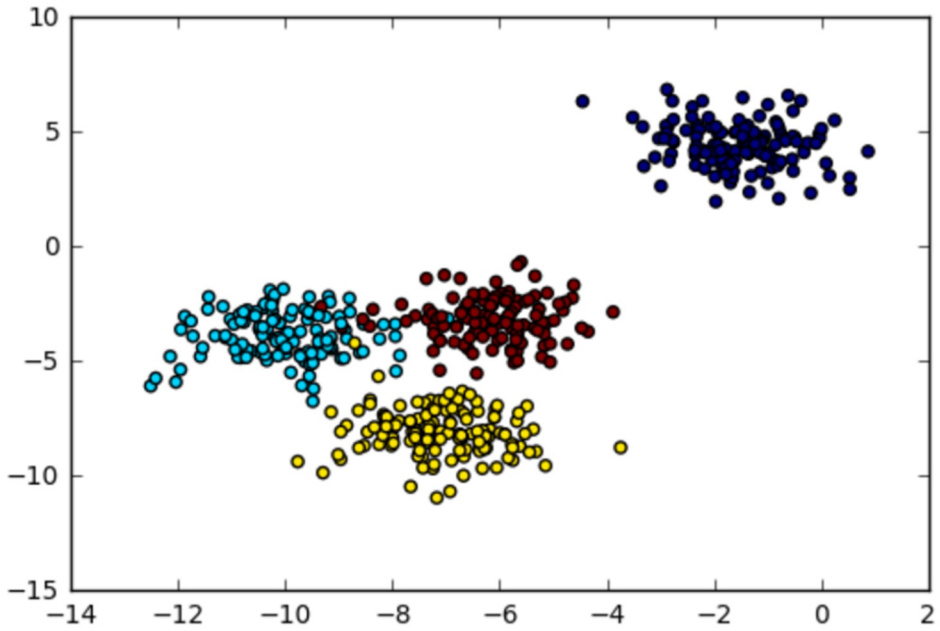
$$b(x_i) = \min_{C_j:\, 1 \leq j \leq k,\, x_i \notin C_j} \left\{ \frac{\sum_{x_k \in C_j} D(x_i, x_k)}{|C_j|} \right\}$$

For the data point x_i and all the other clusters not containing x_i, caculate its average distance to all the other data points in the given clusters. Find the minimum distance value with respect to the given clusters.

Finally, *Silhouette Coefficient*: $s(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}}$, $-1 \leq s(x_i) \leq 1$. Want $a(x_i) < b(x_i)$ and $a(x_i) \to 0$ so as to $s(x_i) \to 1$.

In [13]:
```
# Generate data.
# This particular setting has one distinct cluster and 3 clusters placed close together.
X, y = make_blobs(n_samples=500,
                  n_features=2,
                  centers=4,
                  cluster_std=1,
                  center_box=(-10.0, 10.0),
                  shuffle=True,
                  random_state=1)

# Plot the data distribution.
plt.scatter(X[:,0], X[:,1], c=y)
```

Out[13]:
```
<matplotlib.collections.PathCollection at 0x112ccd950>
```

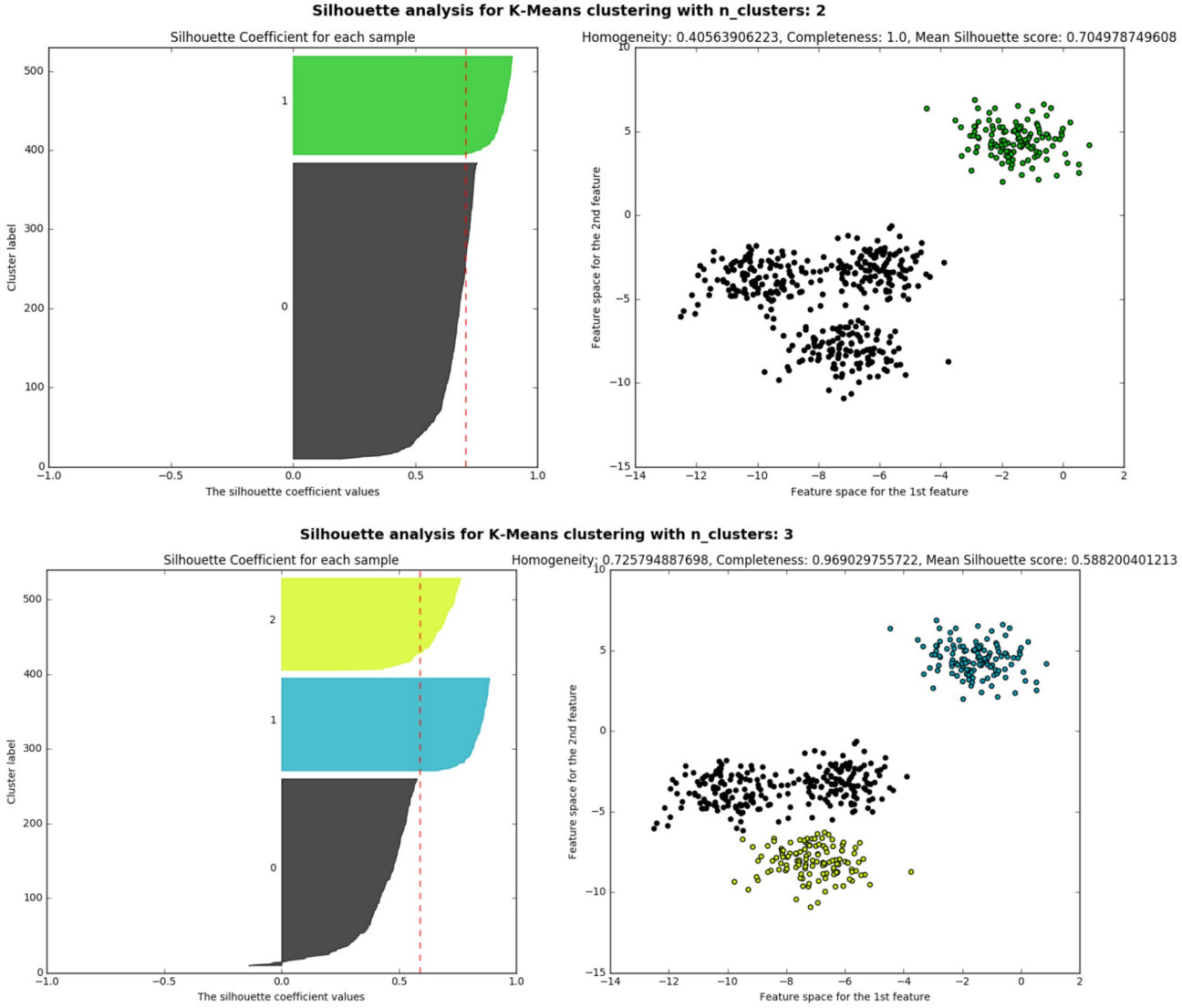


首先，我們先產生四群 clusters 的資料，並且視覺化出來，注意到一點是我們故意產生出三群 clusters 是稍微有重疊到的，方便我們用上述所提到的 cluster quality measurements 來跑看看如果選擇不同的 “k”，系統跑出來的 cluster quality 指數是好還是壞。

In [14]:
```
# List of number of clusters
range_n_clusters = [2, 3, 4, 5, 6]

# For each number of clusters, perform Silhouette analysis and visualize the results.
for n_clusters in range_n_clusters:

    # Perform k-means.
    kmeans = KMeans(n_clusters=n_clusters, random_state=10)
    y_pred = kmeans.fit_predict(X)

    # Compute the cluster homogeneity and completeness.
    homogeneity = metrics.homogeneity_score(y, y_pred)
    completeness = metrics.completeness_score(y, y_pred)

    # Compute the Silhouette Coefficient for each sample.
    s = metrics.silhouette_samples(X, y_pred)

    # Compute the mean Silhouette Coefficient of all data points.
    s_mean = metrics.silhouette_score(X, y_pred)
```

上述程式碼解釋：

- 首先我們先設定要跑的 “k” 的範圍，此例我們嘗試跑 k=2, 3, 4, 5, 6 各個不同的 case，並且使用 for 迴圈，分別計算不同 “k” 之分群結果的 homogeneity, completeness 跟 silhouette coefficient。
- 使用 “homogeneity_score(y, y_pred)” 與 “completeness_score(y, y_pred)” 即可計算對應的兩種指標
- 注意到 scikit-learn 在計算 silhouette coefficient 時有提供兩種選擇：
    - 使用 “silhoette_samples(X, y_pred)” 來計算每個資料點的 silhouette coefficient。
    - 使用 “silhoette_score(X, y_pred)” 來計算每個資料點的 silhouette coefficient 加總之後的平均值。

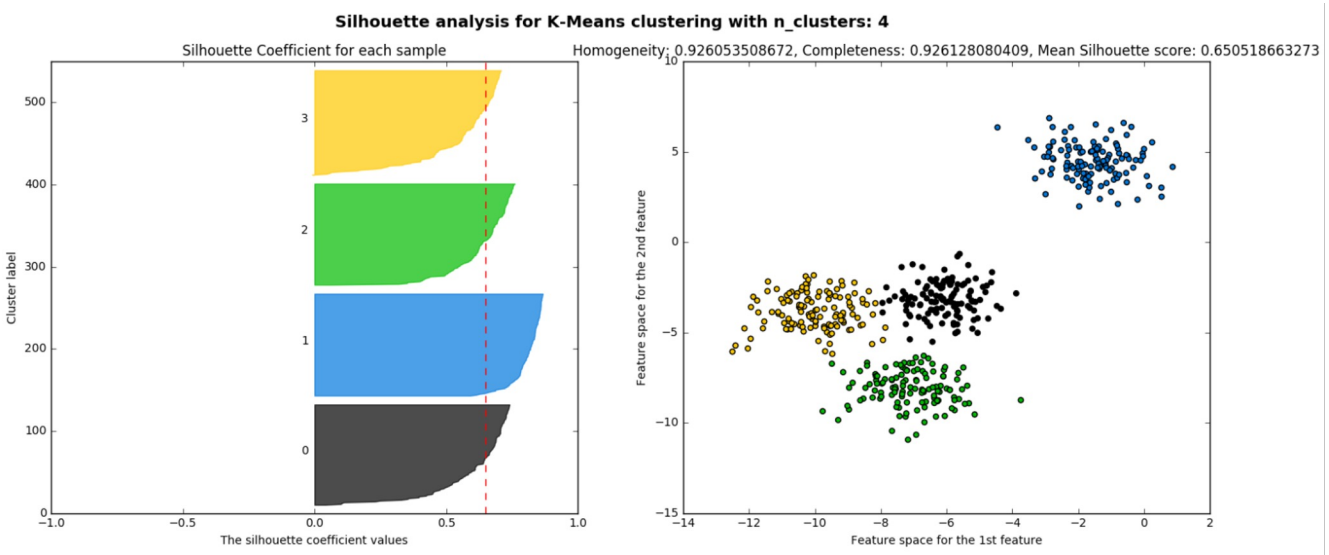


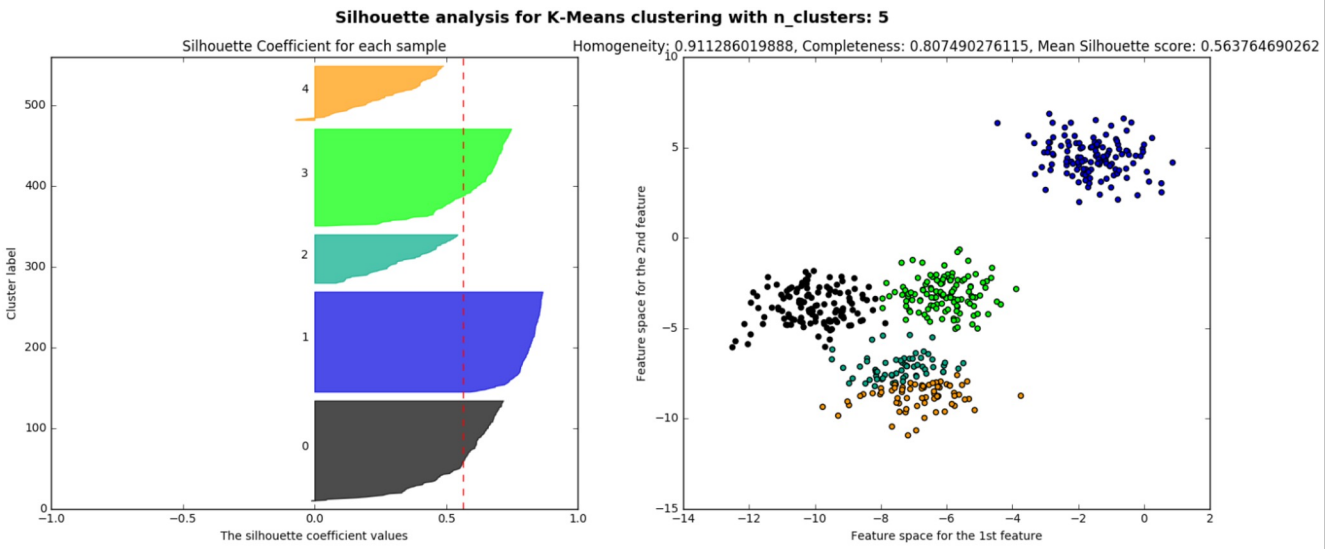
由於視覺化的程式碼稍微複雜，有興趣請查看 Ipython。

我們先來講講怎麼看這張圖吧！

右邊很簡單，就是分群後的結果，
主要是左邊：

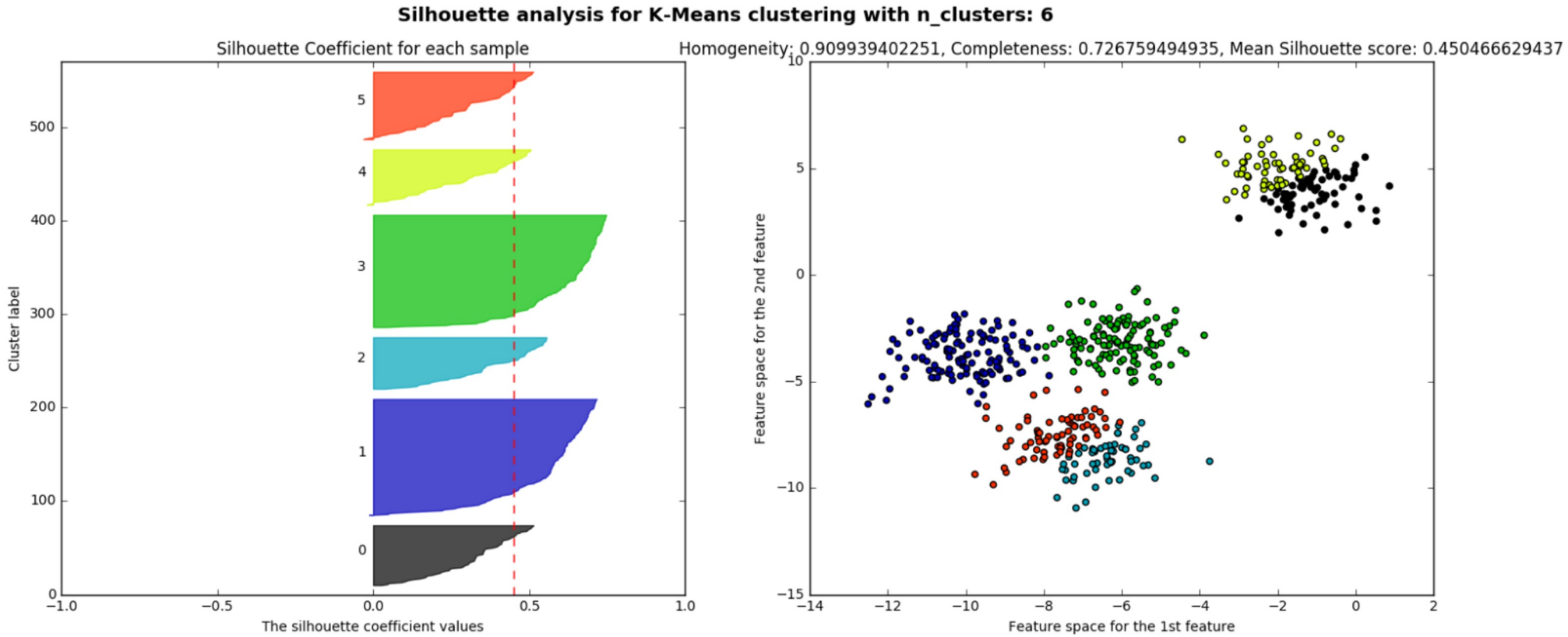
- 橫軸代表 silhouette coefficient。
- 縱軸代表 data points。
- 相同 cluster 的資料點畫成相同顏色，且該 cluster 內的資料點之 silhouette coefficient 皆排序過。
- 注意：紅色的垂直虛線是 “silhouette_score()” 算出來的全部 silhouette coefficient 加總之平均。





← k=4 時，每個 cluster 內的資料點之 silhouette coefficient 普遍高於平均值（超出紅色垂直線），因此以 k=4 來講，分群效果還算不錯。

← k=5 時，可以注意到有兩群（土黃色以及蒂芬尼藍）的平均 silhouette coeffiecient 很低，因為兩群在右圖中重疊很多（影響到 silhouette coeffiecient 中計算 mean nearest cluster distance b(x)的指標），因此 k=5 不是一個好選擇。



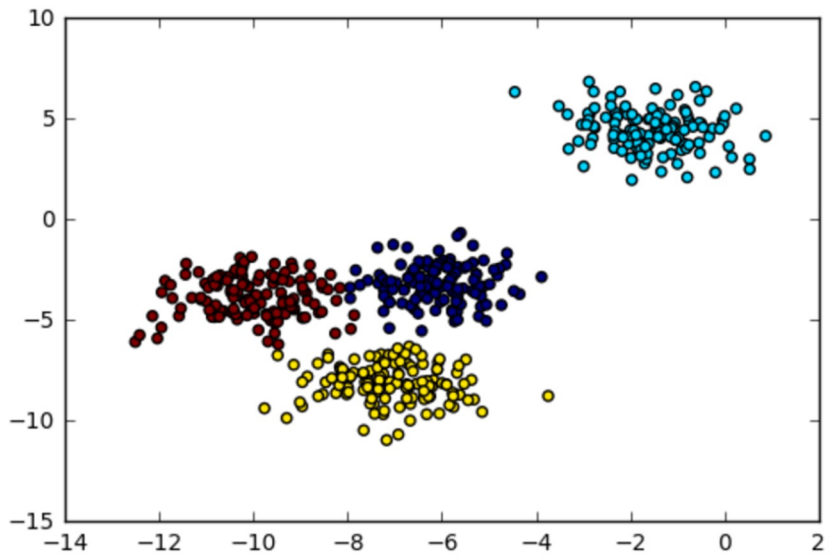
The silhouette plot shows that the `n_clusters` value of 3, 5 and 6 are a bad pick for the given data due to the presence of clusters with above average silhouette scores and also due to wide fluctuations in the size of the silhouette plots. Silhouette analysis is more ambivalent in deciding between 2 and 4.

總體來說，k=3, 5, 6 分群效果不佳，而 k=2, 4 還不錯，至於要選 2 還是 4，就看應用在什麼問題上了。

弱點二：不能處理 Noise data 與 Outliers

In [15]:
```
# Generate data.
# This particular setting has one distinct cluster and 3 clusters placed close together.
# (Same as the above example)
X, y = make_blobs(n_samples=500,
                  n_features=2,
                  centers=4,
                  cluster_std=1,
                  center_box=(-10.0, 10.0),
                  shuffle=True,
                  random_state=1)

# Perform k-means with n_clusters=4
kmeans = KMeans(n_clusters=4, random_state=10)
y_pred = kmeans.fit_predict(X)

# Plot the prediction
plt.scatter(X[:,0], X[:,1], c=y_pred)
```

Out[15]:
```
<matplotlib.collections.PathCollection at 0x1066c8e50>
```



一樣使用上面同樣的資料，並且使用 k=4 去分群，可以發現其實還是有些點離它們各自的 cluster 很遠，但是 k-means 依然將他們分群至 cluster 內。

那 … 我們要怎麼把這些離得比較遠的點，也就是所謂的 outliers 給偵測出來呢？

Solution: Use Distance Threshold to Detect Noise data and Outliers

However, we can detect the noises/outliers conditioning on whether the distance between the data point x_i and the centroid c_j of x_i 's corresponding cluster is larger than the average distance in the cluster. That is to say:

$$x_i = \begin{cases} \text{Outlier}, & \text{if } D(x_i, c_j) > \frac{1}{|Cluster_j|} \sum_{k=0,\, k \neq i}^{|Cluster_j|} D(x_k, c_j) \\ \text{Non-outlier}, & \text{otherwise} \end{cases} \quad \text{where } c_j \in Cluster_j$$

Let's begin to find out the outliers of each cluster.

你可以藉由計算每個 cluster 中的 average distance 當作判斷一個資料點是否是 outlier 的 distance threshold (有時候 distance threshold 還會再乘上一個 ratio 來控制 threshold 大小）。

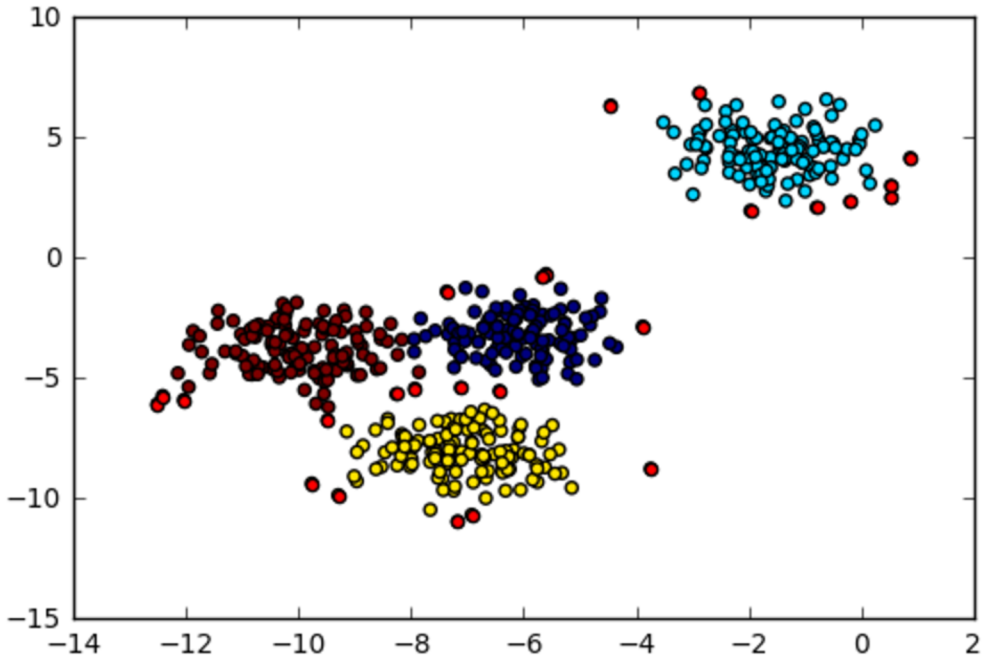
In [16]:

```
# Ratio for our distance threshold, controlling how many outliers we want to detect.
distance_threshold_ratio = 2.0

# Plot the prediction same as the above.
plt.scatter(X[:,0], X[:,1], c=y_pred)

# For each ith cluster, i=0~3 (we have 4 clusters in this example).
for i in [0, 1, 2, 3]:

    # Retrieve the indexs of data points belong to the ith cluster.
    # Note: `np.where()` wraps indexs in a tuple, thus we retrieve indexs using `tuple[0]`
    indexs_of_X_in_ith_cluster = np.where(y_pred == i)[0]

    # Retrieve the data points by the indexs
    X_in_ith_cluster = X[indexs_of_X_in_ith_cluster]

    # Retrieve the centroid.
    centroid = kmeans.cluster_centers_[i]

    # Compute distances between data points and the centroid.
    # Same as: np.sqrt(np.sum(np.square(X_in_ith_cluster - centroid), axis=1))
    # Note: distances.shape = (X_in_ith_cluster.shape[0], 1). A 2-D matrix.
    distances = metrics.pairwise.euclidean_distances(X_in_ith_cluster, centroid)

    # Compute the mean distance for ith cluster as our distance threshold.
    distance_threshold = np.mean(distances)

    # Retrieve the indexs of outliers in ith cluster
    # Note: distances.flatten() flattens 2-D matrix to vector, in order to compare with scalar `distance_threshold`.
    indexs_of_outlier = np.where(distances.flatten() > distance_threshold * distance_threshold_ratio)[0]

    # Retrieve outliers in ith cluster by the indexs
    outliers = X_in_ith_cluster[indexs_of_outlier]

    # Plot the outliers in ith cluster.
    plt.scatter(outliers[:,0], outliers[:,1], c='r')
```

← 1. 取得 cluster 內所有資料點，以及該 cluster 的 centroid。

← 2. 計算 cluster 內所有點到 centroid 的距離，並且平均起來，得出 distance threshold。

← 3. 找出資料點到 centroid 的距離如果大於 (distance threshold * ratio)，則視為 outliers。



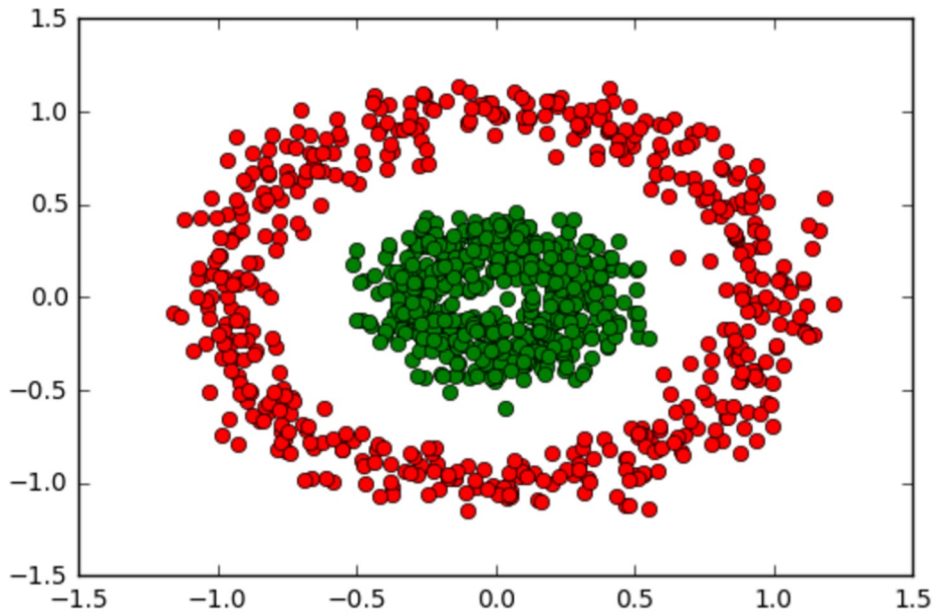
← 藉由上述方法，我們成功將 outliers（紅色）給偵測出來了！不過其實值得提醒的一點是：當你的 cluster 之中佔有太多的 outliers，可想而知，k-means 分群出來得結果還是會不理想喔！因為在計算 distance threshold 時，還是會將這些 outliers 列入計算，因此大大影響了 distance threshold。

As we've mentioned about measuring cluster quality analysis, you can run different settings of `distance_threshold_ratio` to find out the best cluster quality.

弱點三：不能處理非球型分佈的資料集

In [17]:
```
# Generate non-spherical data.
X, y = make_circles(n_samples=1000, factor=0.3, noise=0.1)

# Plot the data distribution. (Here's another way to plot scatter graph)
plt.plot(X[y == 0, 0], X[y == 0, 1], 'ro')
plt.plot(X[y == 1, 0], X[y == 1, 1], 'go')
```

Out[17]:
```
[<matplotlib.lines.Line2D at 0x10f8e1fd0>]
```



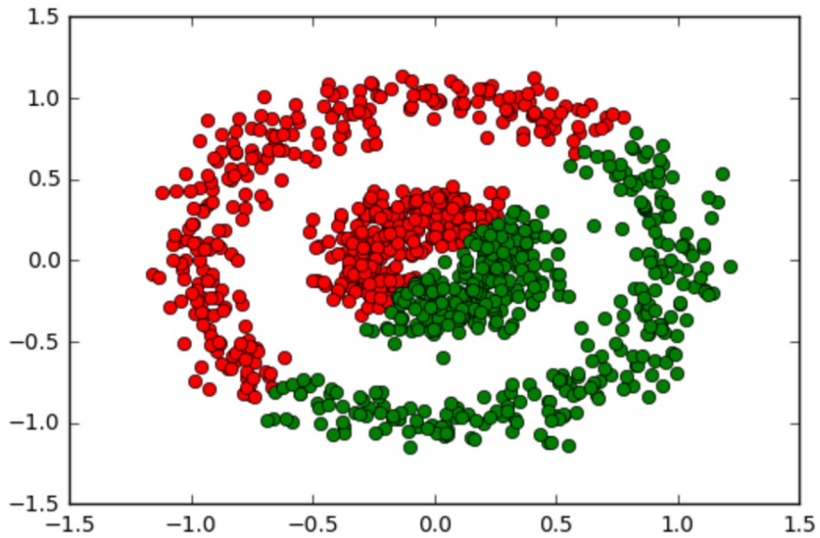
k-means 演算法根據其定義：此演算法目的是將資料點分成離它們最接近的 mean 值 的 cluster 之中。這個假設建立於你的資料分布必須得是球型分佈，因此如果我們的資料長得像左圖這種同心圓資料分佈，就會輕易的使 k-means 分群的效果不佳。

讓我們來看看當 k=2 時，k-means 會將資料分成什麼樣子。

After performing *K-means* on non-spherical data, the following result shows that it fails to cluster non-spherical data, since *K-means* has an assumption that the data distribution is spherical.

In [18]:
```
# Run k-means on non-spherical data.
y_pred = KMeans(n_clusters=2, random_state=170).fit_predict(X)

# Plot the predictions.
plt.plot(X[y_pred == 0, 0], X[y_pred == 0, 1], 'ro')
plt.plot(X[y_pred == 1, 0], X[y_pred == 1, 1], 'go')

# Print the evaluations
print('Homogeneity: {}'.format(metrics.homogeneity_score(y, y_pred)))
print('Completeness: {}'.format(metrics.completeness_score(y, y_pred)))
print('Mean Silhouette score: {}'.format(metrics.silhouette_score(X, y_pred)))
```

```
Homogeneity: 0.000184968858484
Completeness: 0.000185182645182
Mean Silhouette score: 0.295848480827
```



使用 k-means 分群後的效果並不佳，因為外圈與內圈的 mean 是差不多的，而 k-means 是基於 euclidean distance 來做分群，因此當 k=2 時，此同心圓有一半會跟其中一個 centroid 比較近，另一半會跟另一個 centroid 比較近，因此就被硬生生的砍成一半了。

Solution: Using Feature Transformation or Extraction Techiques Makes Data Clusterable

If you know that your clusters will always be concentric circles, you can simply convert your cartesian (x-y) coordinates to polar coordinates, and use only the radius for clustering - as you know that the angle theta doesn't matter.

Or more generally: use a suitable kernel for k-means clustering, e.g. use *Kernel PCA* to find a projection of the data that makes data linearly separable, or use another clustering algorithm, such as *DBSCAN*.

In [19]:

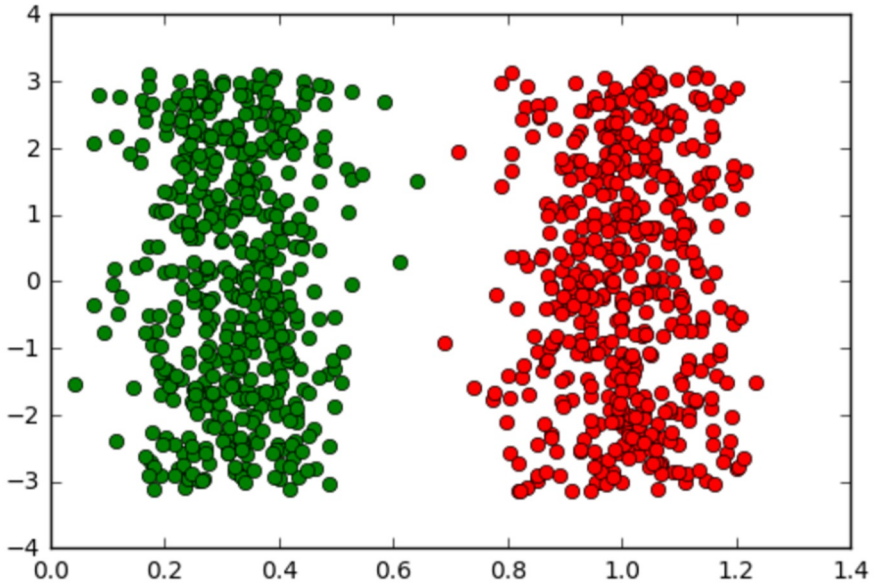
```
def cart2pol(x, y):
    radius = np.sqrt(x**2 + y**2)
    theta = np.arctan2(y, x)
    return radius, theta

X_transformed = np.zeros_like(X)
# Convert cartesian (x-y) to polar coordinates.
X_transformed[:,0], _ = cart2pol(X[:,0], X[:,1])

# Only use `radius` feature to cluster.
y_pred = KMeans(n_clusters=2).fit_predict(X_transformed)

plt.plot(X[y_pred == 0, 0], X[y_pred == 0, 1], 'ro')
plt.plot(X[y_pred == 1, 0], X[y_pred == 1, 1], 'go')
```

← 將 Carteisan 座標轉換成 polar 座標的 function。

← 我們只看第一個 feature－radius，第二個 feature 就全部填 0，最後直接丟到 k-means 做分群。

In [38]:
```
def cart2pol(x, y):
    radius = np.sqrt(x**2 + y**2)
    theta = np.arctan2(y, x)
    return radius, theta

X_transformed = np.zeros_like(X)
X_transformed[:,0], X_transformed[:,1] = cart2pol(X[:,0], X[:,1])

plt.plot(X_transformed[y == 0, 0], X_transformed[y == 0, 1], 'ro')
plt.plot(X_transformed[y == 1, 0], X_transformed[y == 1, 1], 'go')
```

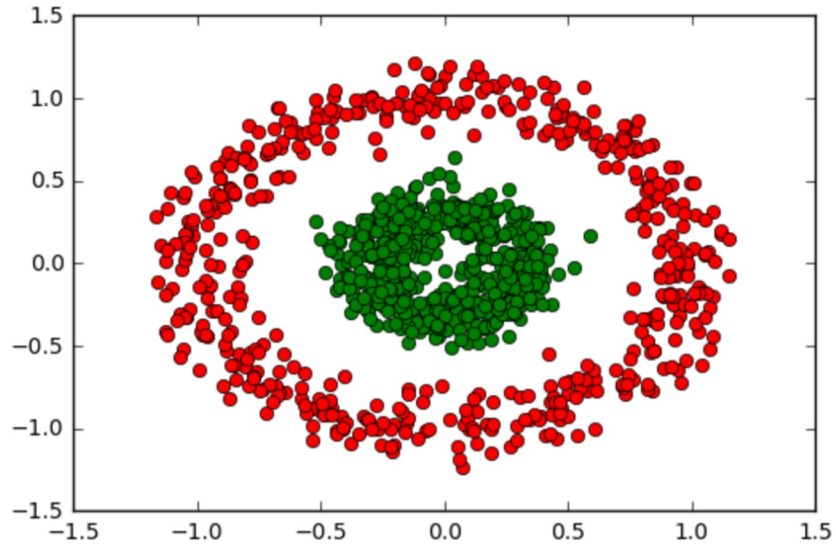
Out[38]:
```
[<matplotlib.lines.Line2D at 0x1163fe890>]
```



We just successfully make data linearly separable by converting features (x-y) to (radius-theta) !

In [39]:
```
def cart2pol(x, y):
    radius = np.sqrt(x**2 + y**2)
    theta = np.arctan2(y, x)
    return radius, theta

X_transformed = np.zeros_like(X)
# Convert cartesian (x-y) to polar coordinates.
X_transformed[:,0], _ = cart2pol(X[:,0], X[:,1])

# Only use `radius` feature to cluster.
y_pred = KMeans(n_clusters=2).fit_predict(X_transformed)

plt.plot(X[y_pred == 0, 0], X[y_pred == 0, 1], 'ro')
plt.plot(X[y_pred == 1, 0], X[y_pred == 1, 1], 'go')
```

Out[39]:
```
[<matplotlib.lines.Line2D at 0x1181a2490>]
```



Now the data is successfully clustered!

DBSCAN: Density-Based Spatial Clustering Algorithm with Noise

Let's begin to perform *DBSCAN* on spherical data

In [20]:

```
# Generate data with 3 centers.
X, y = make_blobs(n_samples=1000,
                  n_features=2,
                  centers=3,
                  random_state=170)

# Standardize features to zero mean and unit variance.
X = StandardScaler().fit_transform(X)

# Perform DBSCAN on the data
y_pred = DBSCAN(eps=0.3, min_samples=30).fit_predict(X)

# Plot the predictions
plt.scatter(X[:,0], X[:,1], c=y_pred)

# Print the evaluations
print('Number of clusters: {}'.format(len(set(y_pred[np.where(y_pred != -1)]))))
print('Homogeneity: {}'.format(metrics.homogeneity_score(y, y_pred)))
print('Completeness: {}'.format(metrics.completeness_score(y, y_pred)))
print('Mean Silhouette score: {}'.format(metrics.silhouette_score(X, y_pred)))
```

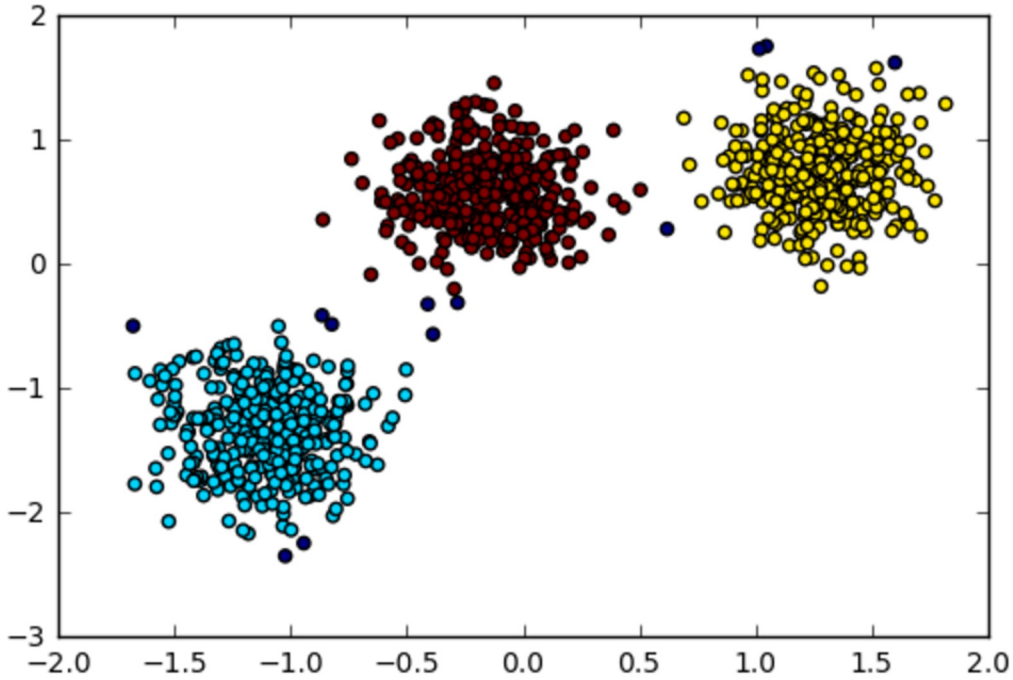
← 1. 產生 (1000x2) 的訓練資料 X 及對應 ground truth y。

← 2. 將資料 standardize 有助於避免某些維度因為 scaling 很大而影響 classification 或 clustering 的結果。

← 3. 使用 DBSCAN 來分群，其參數 ” min_samples” 等同前面所介紹的 “MinPts”。

← 4. 印出 cluster quality 指數

```
Number of clusters: 3
Homogeneity: 0.982039796605
Completeness: 0.93781096194
Mean Silhouette score: 0.679854880541
```



The **black** data points denote the **outliers** in the above result.

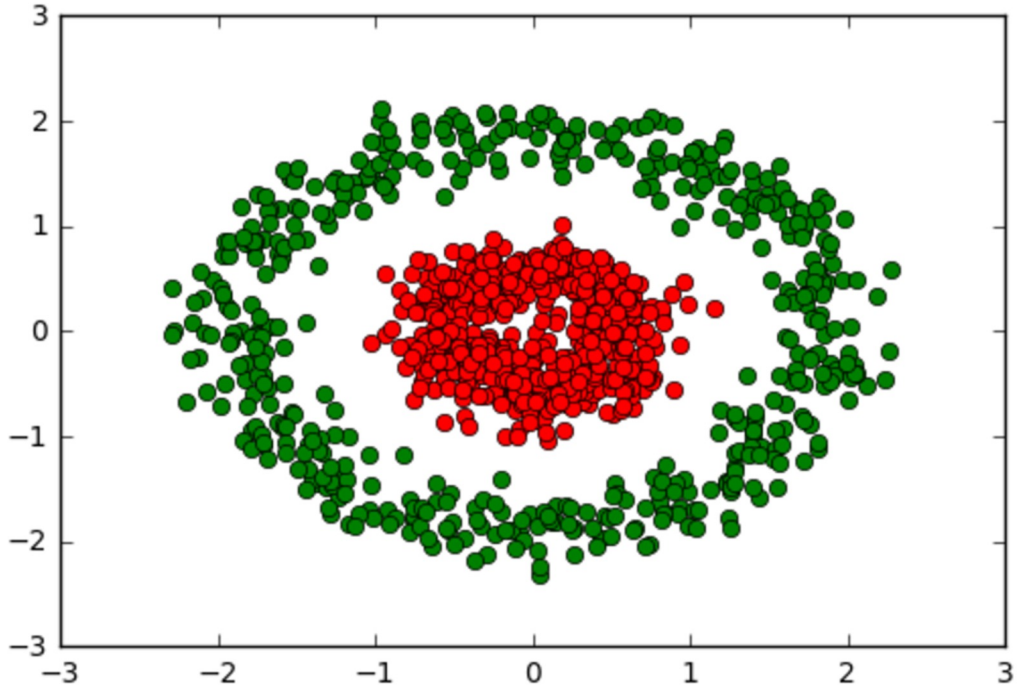
Note that we don't need to specify the number of clusters with *DBSCAN* algorithm. Besides, *DBSCAN* is good at finding out the outliers without requiring some hacks like we did above in *K-means* section.

讓我們來試試 DBSCAN 跑在非球狀資料分布上

Now, let's try *DBSCAN* on non-spherical data.

In [21]:
```
# Generate non-spherical data.
X, y = make_circles(n_samples=1000, factor=0.3, noise=0.1)

# Standardize features to zero mean and unit variance.
X = StandardScaler().fit_transform(X)

# Perform DBSCAN on the data
y_pred = DBSCAN(eps=0.3, min_samples=10).fit_predict(X)

# Plot the data distribution. (Here's another way to plot scatter graph)
plt.plot(X[y_pred == 0, 0], X[y_pred == 0, 1], 'ro')
plt.plot(X[y_pred == 1, 0], X[y_pred == 1, 1], 'go')

# Print the evaluations
print('Number of clusters: {}'.format(len(set(y_pred[np.where(y_pred != -1)]))))
print('Homogeneity: {}'.format(metrics.homogeneity_score(y, y_pred)))
print('Completeness: {}'.format(metrics.completeness_score(y, y_pred)))
print('Mean Silhouette score: {}'.format(metrics.silhouette_score(X, y_pred)))
```

```
Number of clusters: 2
Homogeneity: 0.996390359526
Completeness: 0.957691789224
Mean Silhouette score: 0.174843819318
```



Comparing to *K-means*, we can directly apply *DBSCAN* on this form of data distribution due to the density-based clustering criterion.

Note: It's worth mention that the *Silhouette score* is generally higher for **convex** clusters than other concepts of clusters, such as density based clusters.

https://kahoot.it/

1st-3rd: 1 pt
4th and 5th: 0.5 pts