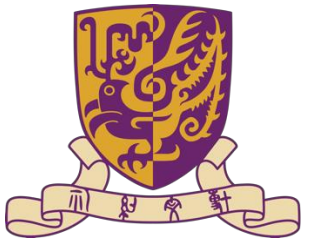


Clustering and classification performance evaluation

Yu LI (李煜)

Wednesday, 9 October 2024

liyu95.com

liyu@cse.cuhk.edu.hk

Department of Computer Science and Engineering (CSE)

The Chinese University of Hong Kong (CUHK)

Comments and question

- Good * 2
- Looking for w part is difficult
- Good to know the reason why we use gradient descent but not directly find min pt

- Do we need to know the python code during the midterm exam?

The problem of KNN

- Need to store all the data

Not memory-efficient

- Need to calculate the distance matrix

Involve square and root

↓

Slow

- Predicting is slow

O(nd) complexity, requires comparisons

↓ ↓

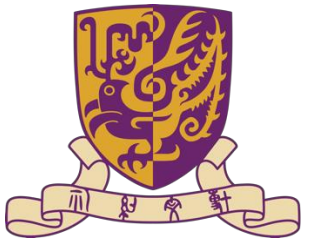
Dimension

Cardinality

Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

Person	P5	Gender
P1	0.267	M
P2	0.809	F
P3	0.358	M
P4	0.636	M
P5	0	??

What if we have a formula?



Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

It seems if H+W is large, the person is very likely to be a Male

No need to calculate the distance matrix
Getting the results with simple arithmetic calculation

$H + W \geq 0.5$ -> Male

P5: 0.4583+0.6667=1.125 ≥ 0.5 -> Male

1. How to get the formula?
2. Different attributes may not be equally important
May not be 0.5

Logistic function

Weight Bias Confidence

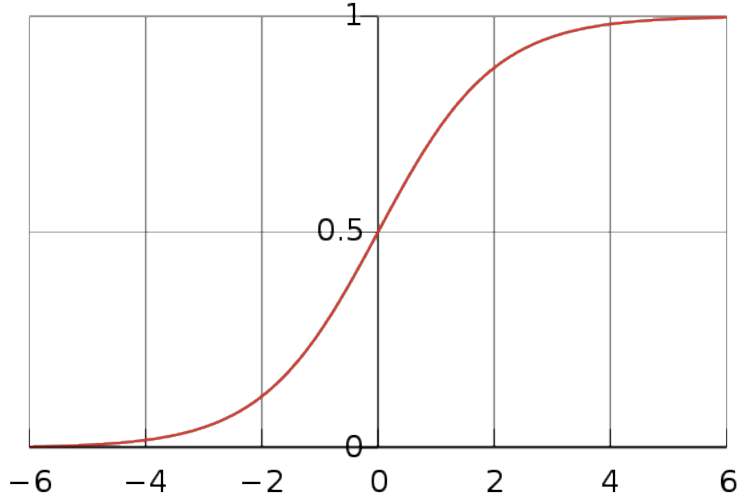
- $w_h H + w_w W + w_0 \geq 0.5$

Independent variables

- What is w_h, w_w, and w_0 are large?

Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

- $\frac{1}{1+e^{-(w_h H + w_w W + w_0)}} \geq 0.5$



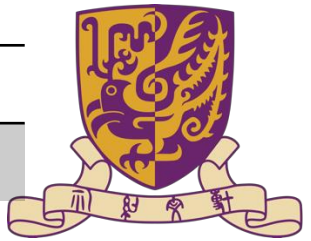
Sigmoid

$$\frac{1}{1+e^{-t}} \geq 0.5$$

- Training: fit the training data
  - To get w_h and w_w, and w_0
- Testing: run the formula

How to train?

Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??



- Training
  - To get w_h and w_w, and w_0
- To make the model fit the training data
- Make $\frac{1}{1+e^{-(w_h H + w_w W + w_0)}} \geq 0.5$ correct for the training data
- $Y^{output} = \frac{1}{1+e^{-(w_h H + w_w W + w_0)}}$
  - 1 for male, 0 for female

```
>>> from sklearn.datasets import load_iris
>>> from sklearn.linear_model import LogisticRegression
>>> X, y = load_iris(return_X_y=True)
>>> clf = LogisticRegression(random_state=0).fit(X, y)
>>> clf.predict(X[:2, :])
array([0, 0])
>>> clf.predict_proba(X[:2, :])
array([[9.8...e-01, 1.8...e-02, 1.4...e-08],
       [9.7...e-01, 2.8...e-02, ...e-08]])
>>> clf.score(X, y)
0.97...
```

Loss function

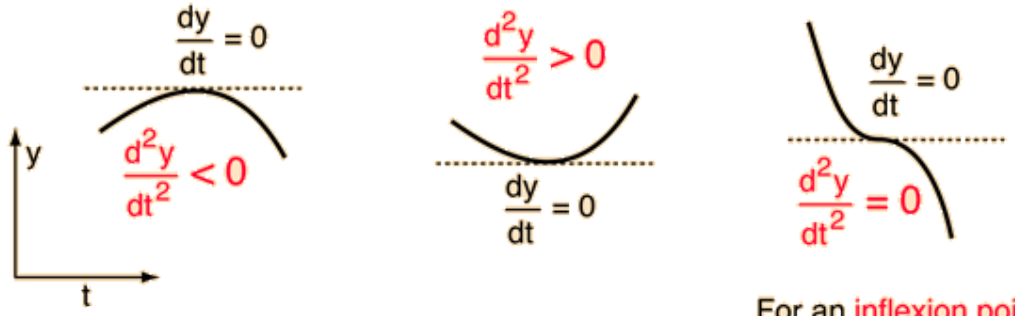
Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

- Training
  - To get w_h and w_w, and w_0
- To make the model fit the training data
- Make $\frac{1}{1+e^{-(w_h H + w_w W + w_0)}} \geq 0.5$ correct for the training data
- $Y^{output} = \frac{1}{1+e^{-(w_h H + w_w W + w_0)}}$
  - 1 for male, 0 for female
- $(Y^{output} - Y)^2$ should be as small as possible
  - Y: the true label we have for training data
  - Loss function that we would like to minimize

Find out the minimum value

❖Calculus??

The second derivative demonstrates whether a point with zero first derivative is a maximum, a minimum, or an inflexion point.

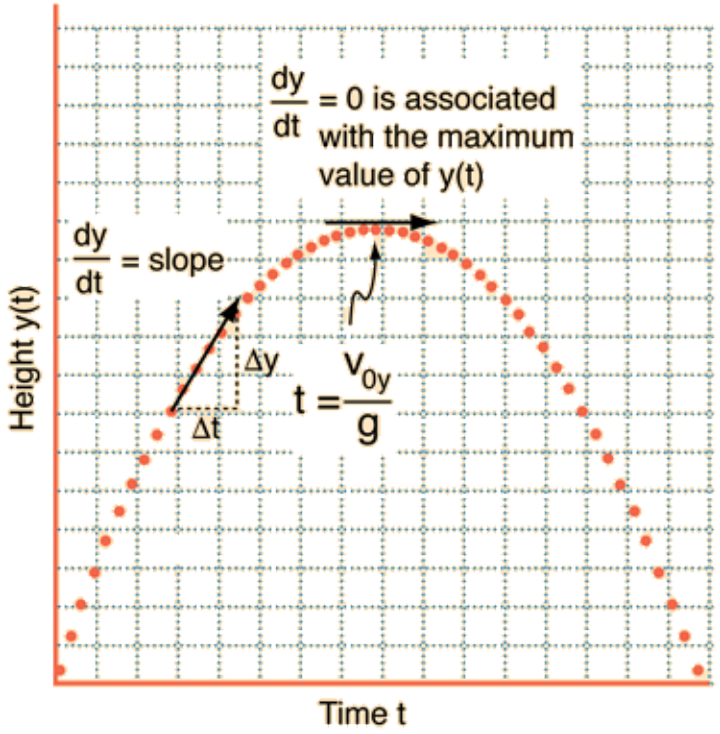


For a maximum, the second derivative is negative. The slope of the curve (first derivative) is at first positive, then goes through zero to become negative.

For a minimum, the second derivative is positive. The slope of the curve = first derivative is at first negative, then goes through zero to become positive.

For an inflexion point, the second derivative is zero at the same time the first derivative is zero. It represents a point where the curvature is changing its sense. Inflexion points are relatively rare in nature.

What if the equation is not easy to resolve?



$$y(t) = v_{0y}t - \frac{1}{2}gt^2$$

$$\frac{dy}{dt} = v_{0y} - gt = 0$$

$$\frac{d^2y}{dt^2} = -g$$

The fact that the second derivative is negative gaurantees that the condition

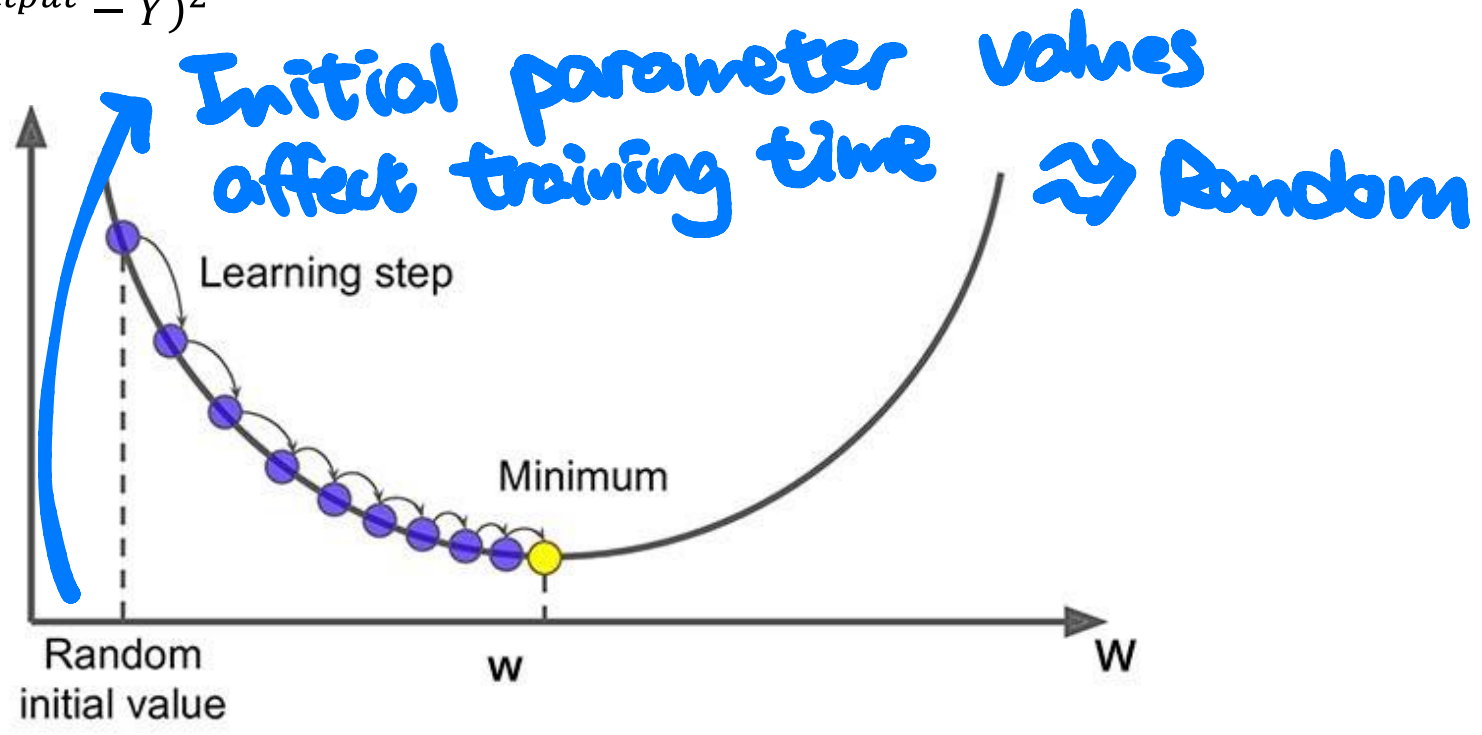
$$\frac{dy}{dt} = 0$$

corresponds to a maximum.

Gradient descent algorithm

- $L = \sum_{P_1}^{P_4}(Y^{output} - Y)^2$ is a function of ws
- For each w, we want to find a value to make the function value smallest

$$\sum_{P_1}^{P_4}(Y^{output} - Y)^2$$



To get the formula

Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

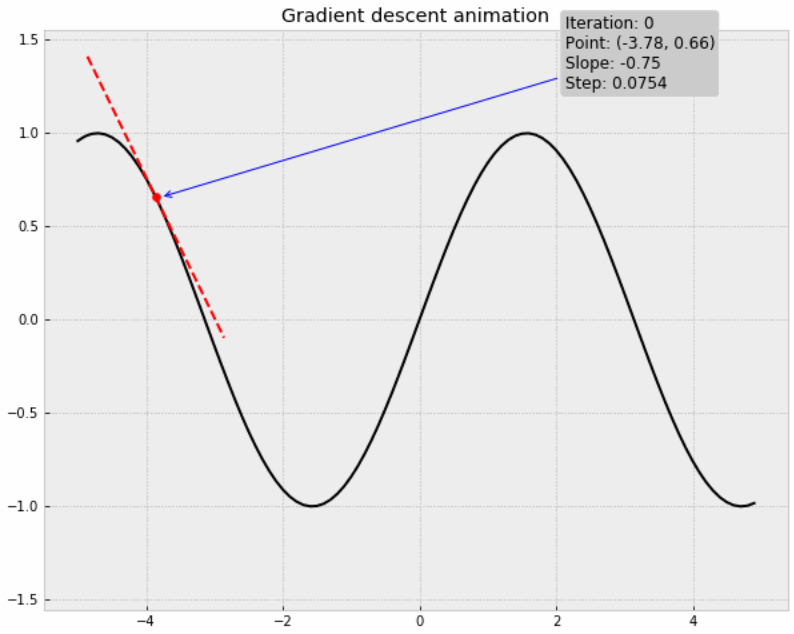
- Initialize w_h and w_w, and w_0
  - Random values
- For P1, P2, P3, P4
  - Calculate the output Y^{output}
  - Update weights
    - $w_i = w_i + \Delta w_i$
    - $\Delta w_i = 2 * \alpha(Y - Y^{output}) \frac{\partial Y^{output}}{\partial w_i}$
    - α is a small constant
- Repeat the above step
  - Until no more to update

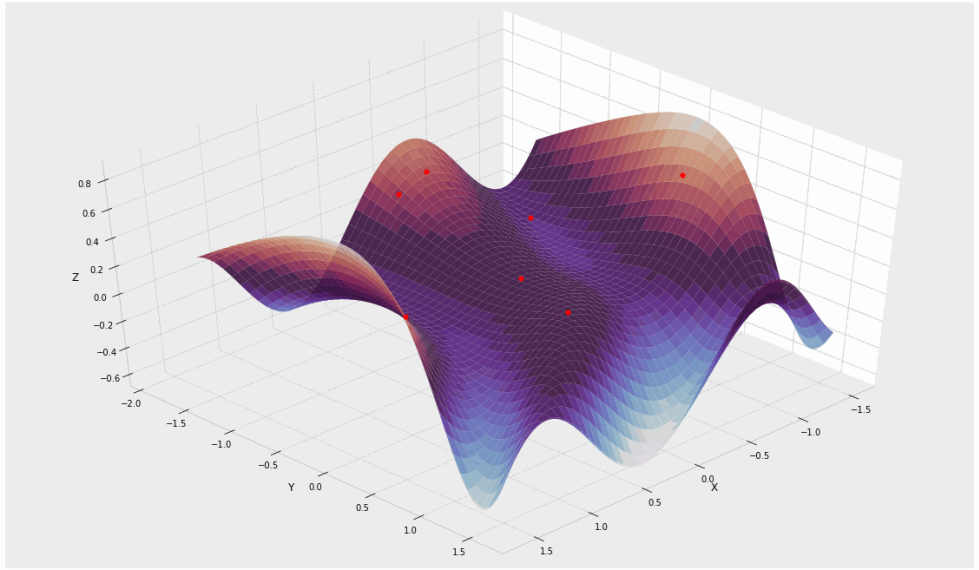
gradient of loss function

$$Y^{output} = \frac{1}{1 + e^{-(w_h H + w_w W + w_0)}}$$

- What are good w_h and w_w, and w_0?
- The ones make $(Y^{output} - Y)^2$ the smallest

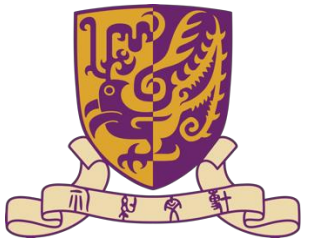
Gradient descent algorithm





https://www.kaggle.com/code/trolukovich/animating-gradien-descent

Gradient descent algorithm



Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

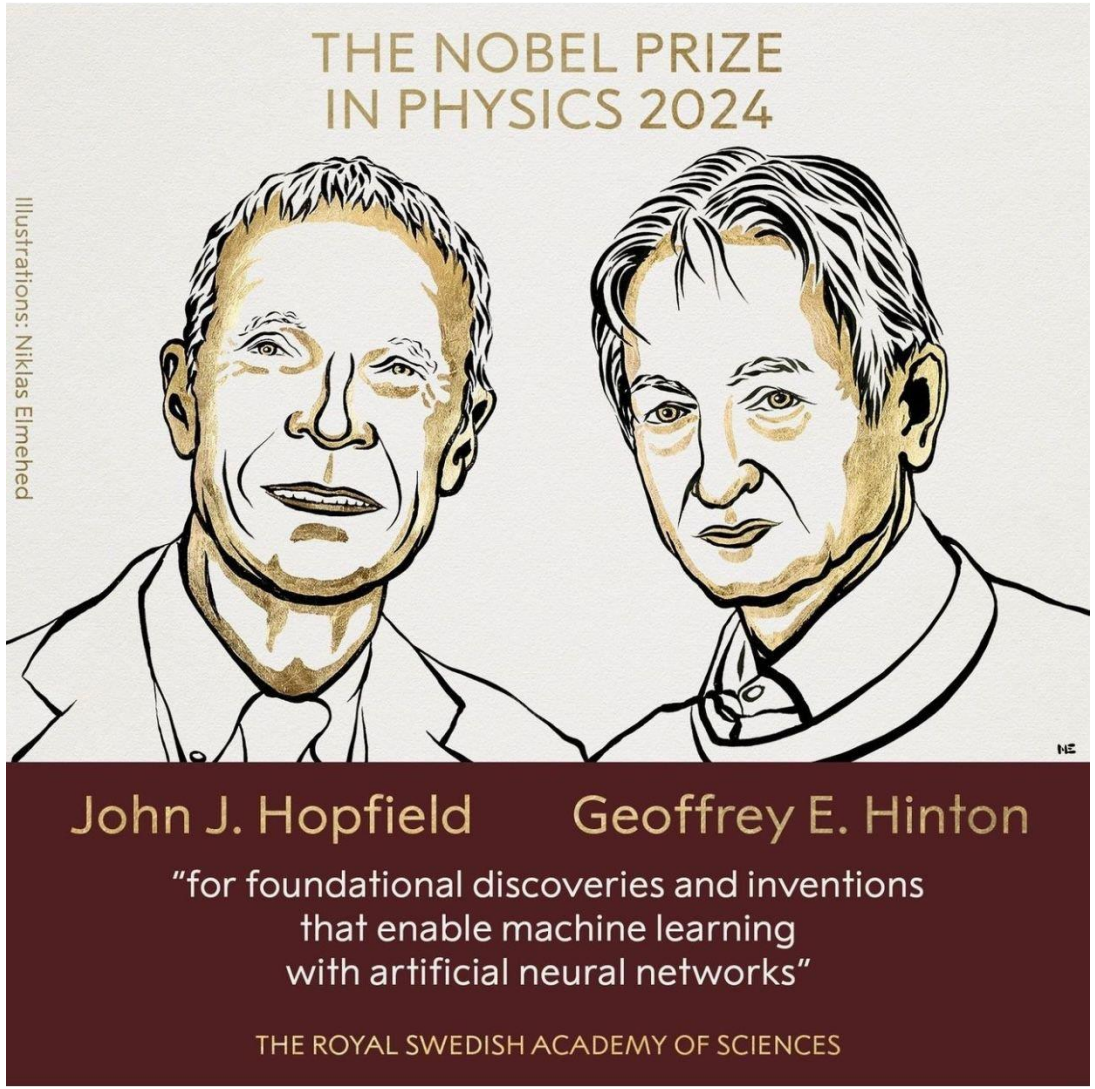
- Initialize w_h and w_w, and w_0
  - Random values
- For P1, P2, P3, P4
  - Calculate the output Y^{output}
  - Update weights
    - $w_i = w_i + \Delta w_i$
    - $\Delta w_i = 2 * \alpha(Y - Y^{output}) \frac{\partial Y^{output}}{\partial w_i}$
    - α is a small constant
- Repeat the above step
  - Until no more to update

```
>>> from sklearn.datasets import load_iris
>>> from sklearn.linear_model import LogisticRegression
>>> X, y = load_iris(return_X_y=True)
>>> clf = LogisticRegression(random_state=0).fit(X, y)
>>> clf.predict(X[:2, :])
array([0, 0])
>>> clf.predict_proba(X[:2, :])
array([[9.8...e-01, 1.8...e-02, 1.4...e-08],
       [9.7...e-01, 2.8...e-02, ...e-08]])
>>> clf.score(X, y)
0.97...
```

To make you awake

https://ureply.mobi/teacher

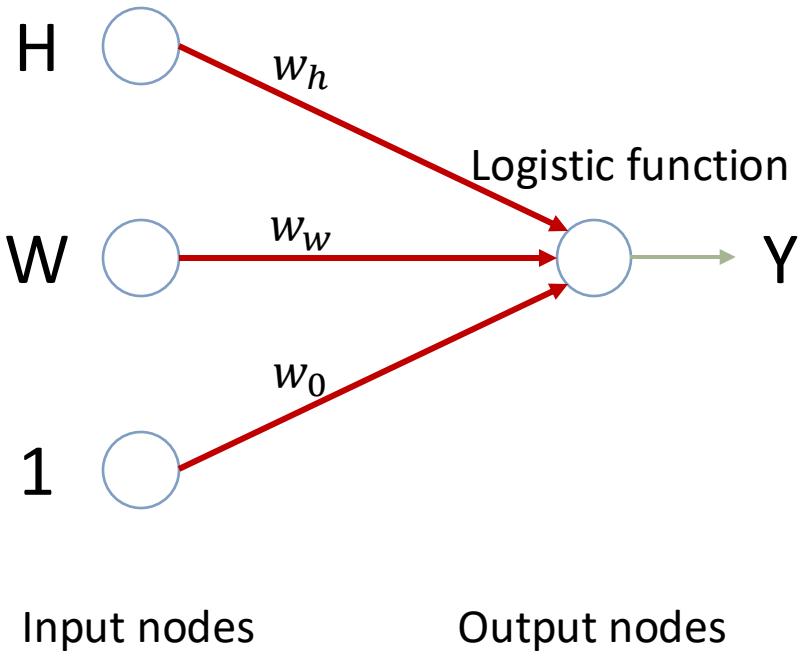
How is logistic regression related to Nobel prize?



The simplest neural network

- $\frac{1}{1+e^{-(w_h H+w_w W+w_0)}} \geq 0.5$

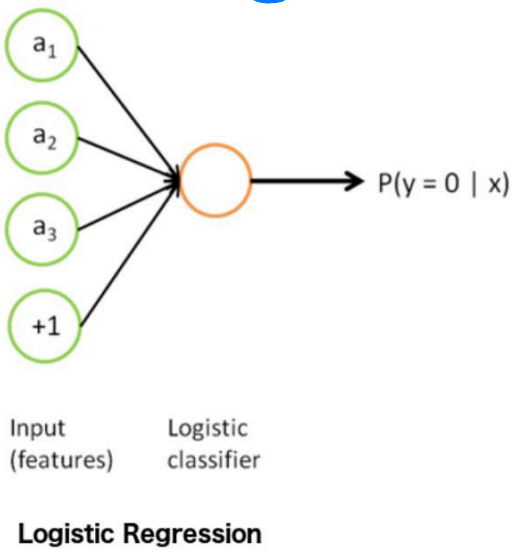
Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??



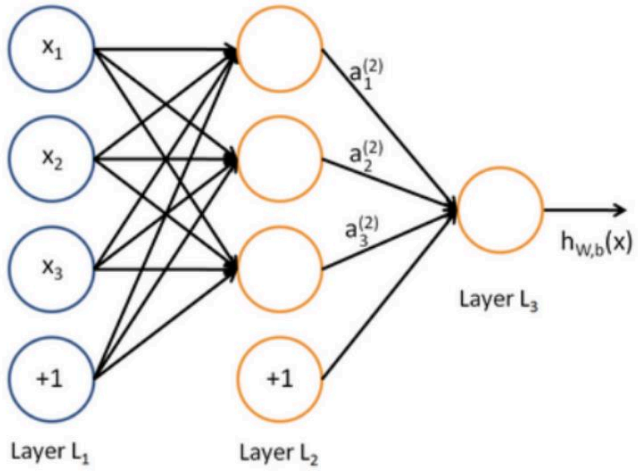
$$Y^{output} = \frac{1}{1+e^{-(w_h H+w_w W+w_0)}}$$

From LR to NN

- Fast prediction
- Successful in real-life problems
- High tolerance to noisy data → Depending on training parameters
- Long training time
- Poor interpretability → Too many parameters

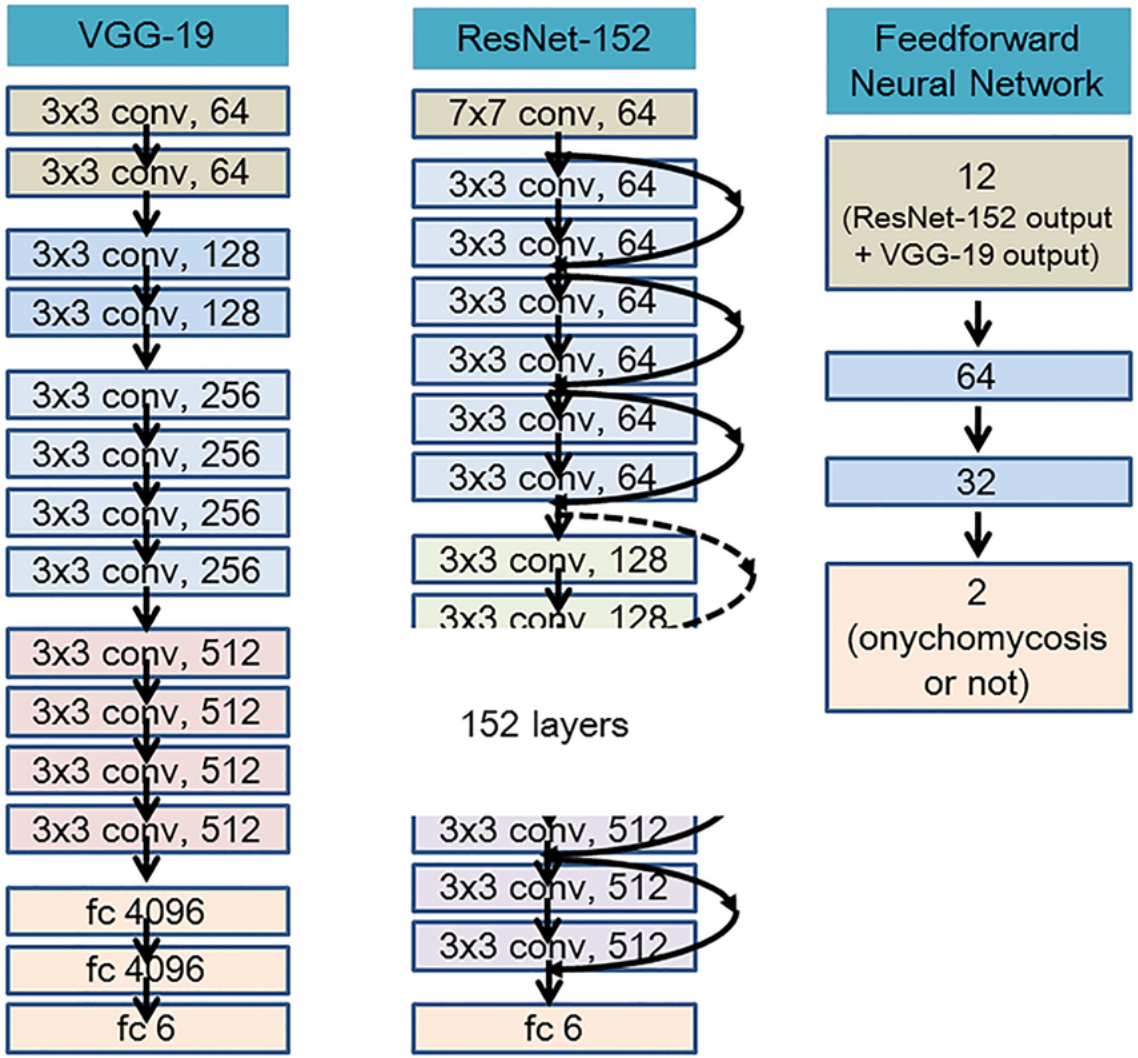


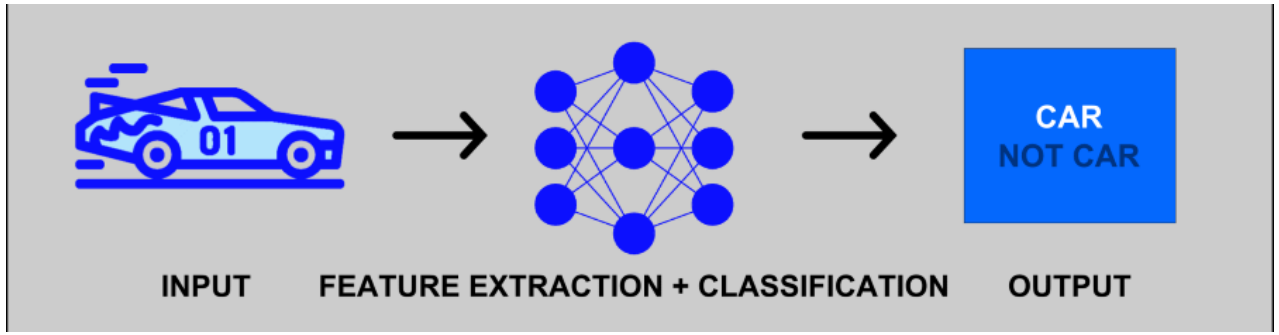
Logistic Regression



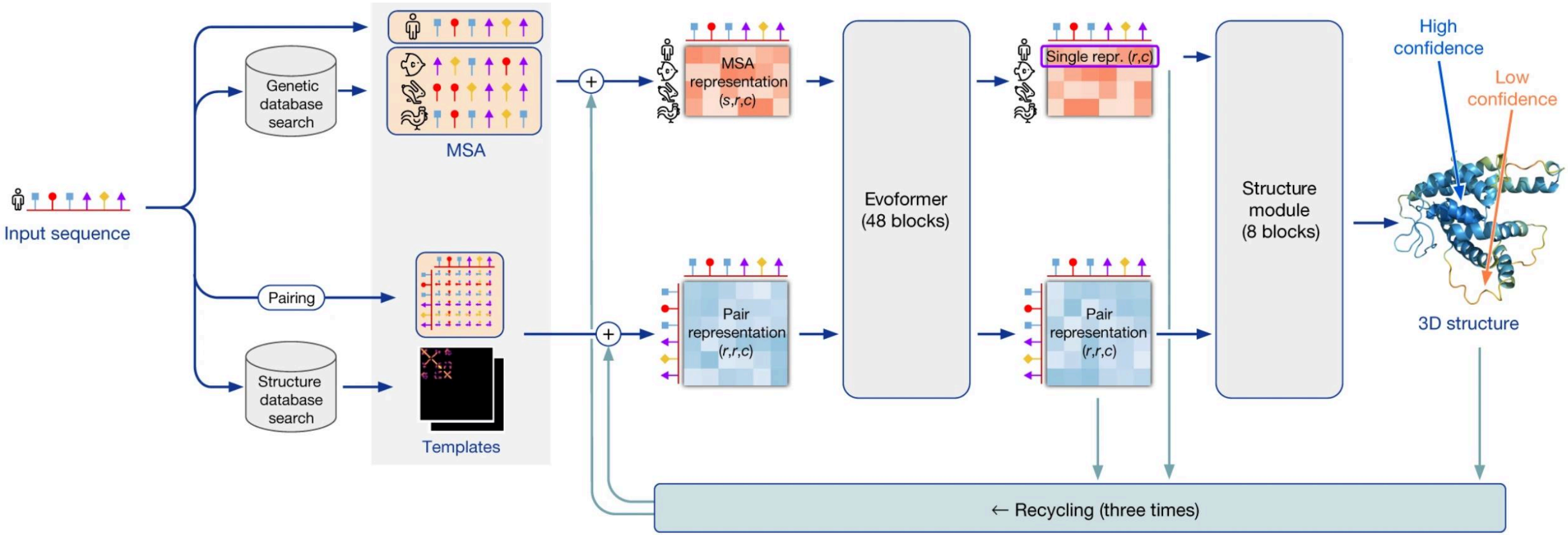
Neural Network

From neural network to deep learning

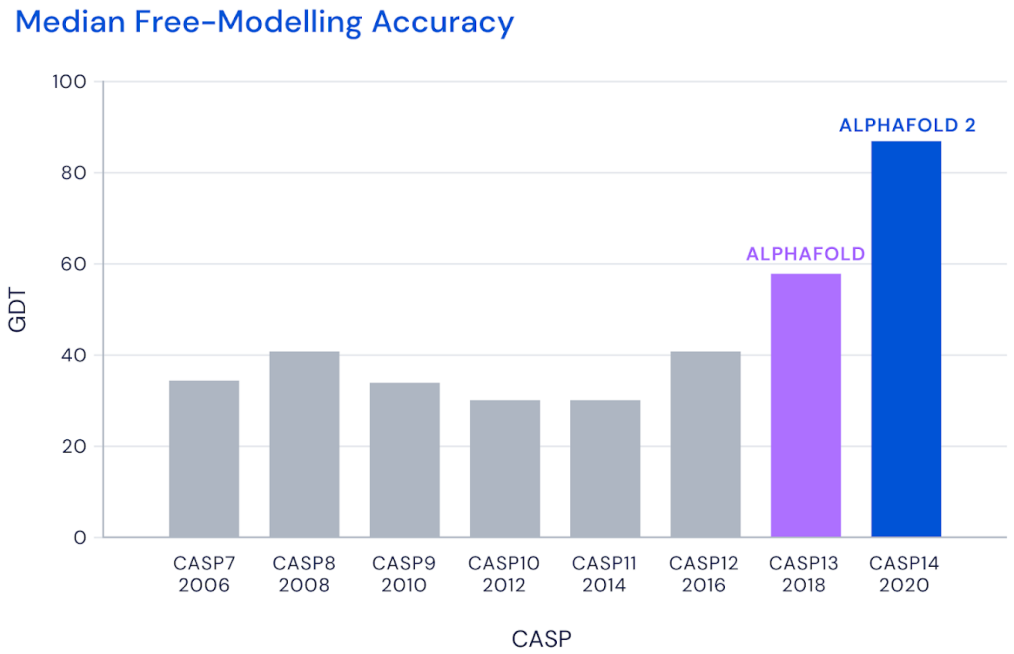




AlphaFold: the most successful deep learning application



AlphaFold: the most successful deep learning application

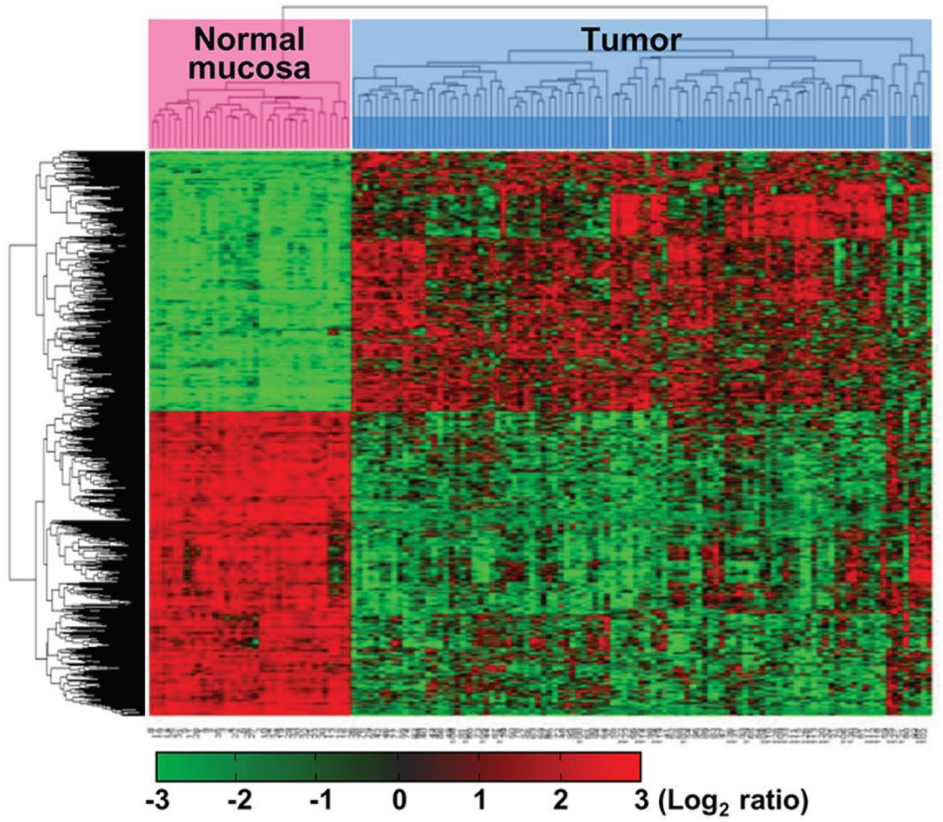


> This computational work represents a stunning advance on the protein-folding problem, a 50-year-old grand challenge in biology. It has occurred decades before many people in the field would have predicted. It will be exciting to see the many ways in which it will fundamentally change biological research.

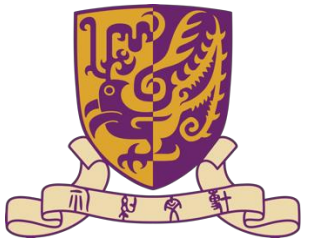
PROFESSOR VENKI RAMAKRISHNAN
NOBEL LAUREATE AND PRESIDENT OF THE ROYAL SOCIETY

Potential project-3

- Data preprocessing for the gene expression matrix
  - Data collecting and merging (if needed)
  - Exploration
  - Visualization
  - Data cleaning
  - Get distance matrix
  - Perform classification

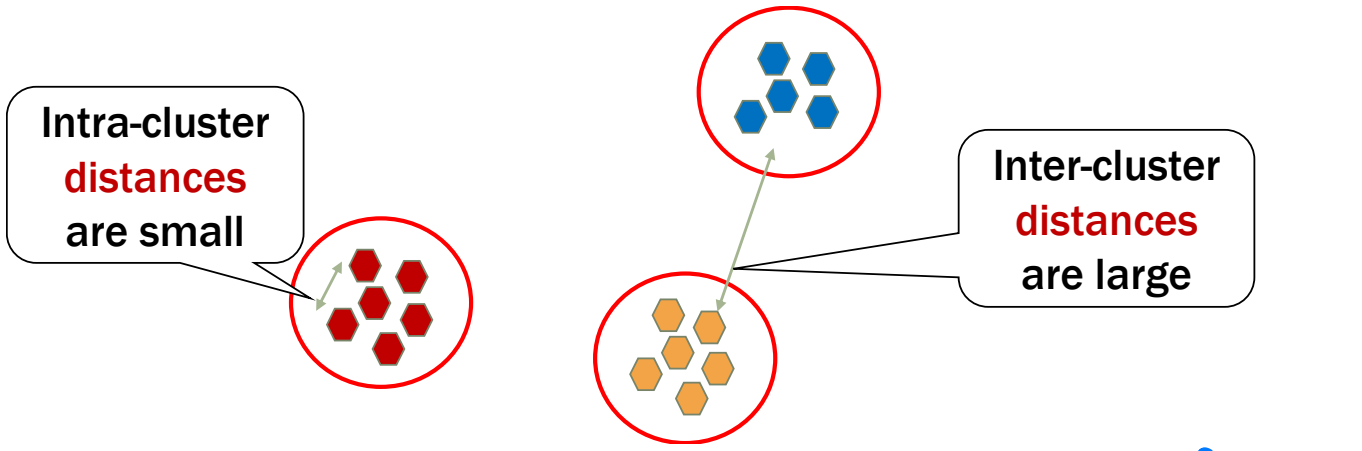


Today's agenda

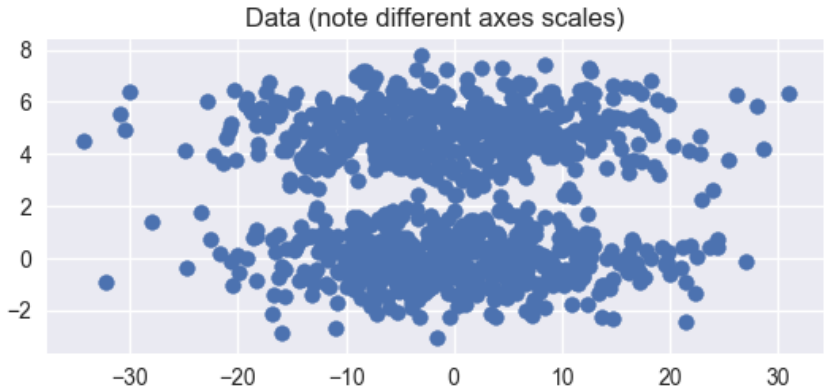


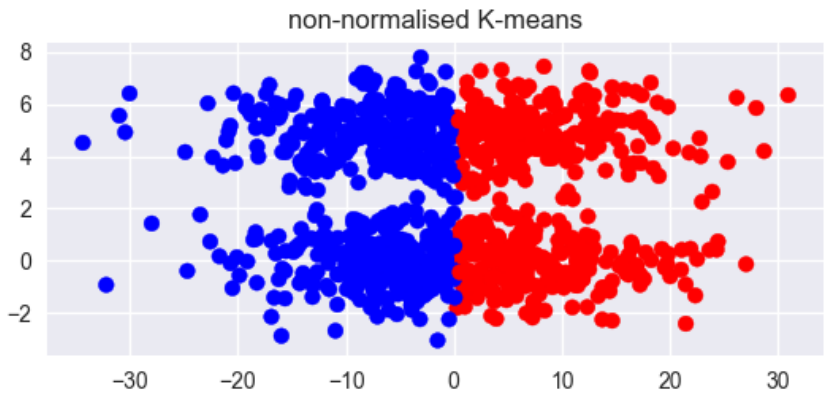
- Performance evaluation
  - Binary classification evaluation
- Cross-validation
- Multi-class classification
- Clustering evaluation

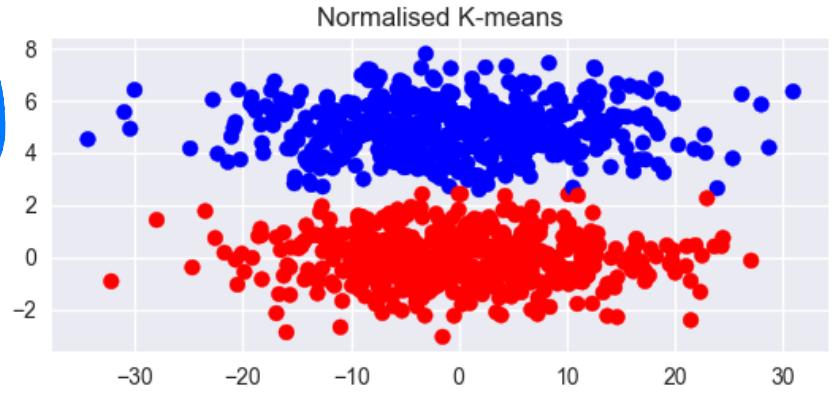
Which clustering method is better?



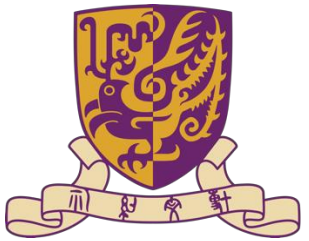
Higher Interpretability







Which classification method should we trust?



Person	Height(m)	Weight(kg)	Gender
P1	1.79	75	M
P2	1.64	54	F
P3	1.70	63	M
P4	1.88	78	M
P5	1.75	70	??

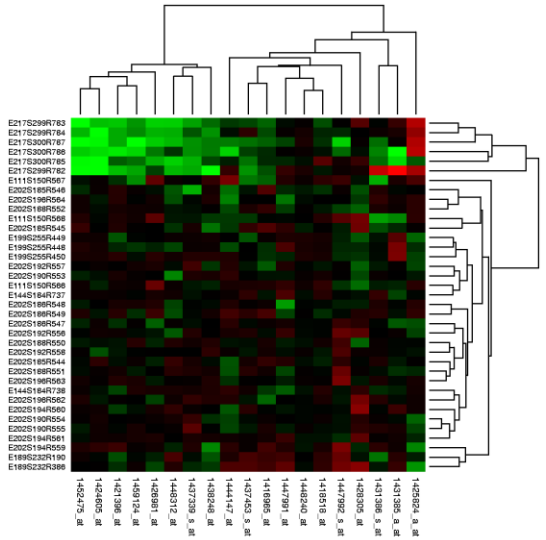
Assign the class accurately

Person	Method 1	Method 2	Method 3
P1	M	F	M
P2	M	F	F
P3	M	M	M
P4	M	M	M
P5	F	F	M

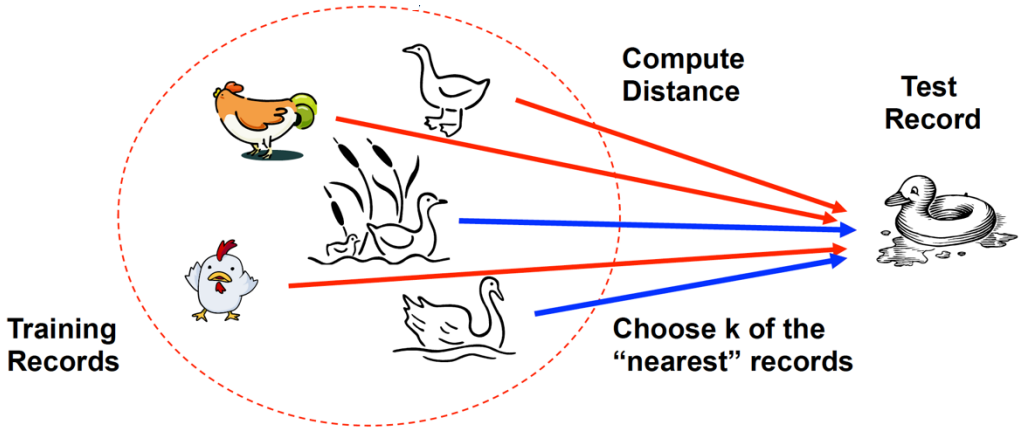
We need some quantitative values to summarize the performance of different methods

The purpose of model evaluation

- Characterize the performance of a model
  - Pinpoint the strong points and weak points of a method
  - Method selection/Model selection

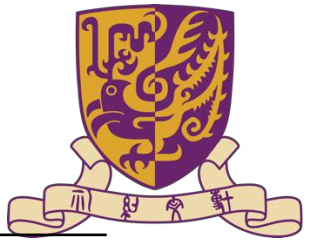


1. Normalization methods
2. Distance measurements
3. Distance between different clusters
4. …



1. Normalization methods
2. Distance measurements
3. K
4. …

Classification performance evaluation



❖Confusion matrix

		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes	a(TP)	b(FN)
	Class=No	c(FP)	d(TN)

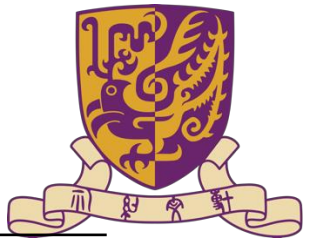
Person	Height(m)	Weight(kg)	Male?	Prediction
P1	1.79	75	Yes	Yes
P2	1.64	54	No	No
P3	1.70	63	Yes	No
P4	1.88	78	Yes	Yes
P5	175	70	Yes	No
P6	1.65	52	No	Yes

TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative

Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Classification performance evaluation



❖ Confusion matrix

		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes		
	Class=No		

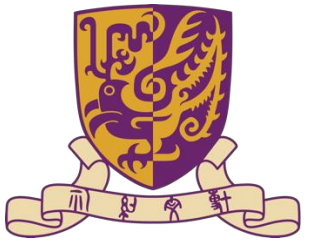
Person	Height(m)	Weight(kg)	Male?	Prediction
P1	1.79	75	Yes	Yes
P2	1.64	54	No	No
P3	1.70	63	Yes	No
P4	1.88	78	Yes	Yes
P5	175	70	Yes	No
P6	1.65	52	No	Yes

TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative

Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Accuracy



	Predicted class		
Actual class		Class=Yes	Class=No
	Class=Yes	45(TP)	4(FN)
	Class=No	6(FP)	45(TN)

$$Accuracy = \frac{TP + TN}{TP + TN + \mathrm{FP} + TN} = \frac{45 + 45}{45 + 45 + 4 + 6} = 0.9$$

What if we have a bad classifier and predict everything as Yes?

	Predicted class		
Actual class		Class=Yes	Class=No
	Class=Yes	49(TP)	0(FN)
	Class=No	51(FP)	0(TN)

$$Accuracy = \frac{TP + TN}{TP + TN + \mathrm{FP} + TN} = \frac{49}{49 + 51} = 0.49$$

Accuracy: limitation

		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes	45(TP)	4(FN)
	Class=No	6(FP)	45(TN)

$$Accuracy = \frac{TP + TN}{TP + TN + \mathrm{FP} + TN} = \frac{45 + 45}{45 + 45 + 4 + 6} = 0.9$$

What if we have a bad classifier and predict everything as Yes?

		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes	4949(TP)	0(FN)
	Class=No	51(FP)	0(TN)

Imbalanced classes

Not very reliable for biased classes

$$Accuracy = \frac{TP + TN}{TP + TN + \mathrm{FP} + TN} = \frac{4949}{4949 + 51} = 0.99$$

Maybe misleading for imbalanced data

Precision, recall, and F1 score

		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes	a(TP)	b(FN)
	Class=No	c(FP)	d(TN)

$$Precision = \frac{a}{a+c}$$

Among the predicted positive samples, how many of them are correct?

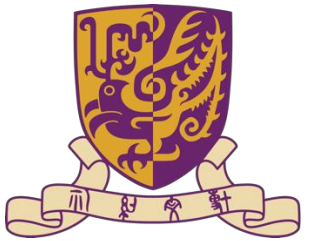
$$Recall = \frac{a}{a+b}$$

How many actual positive samples are predicted to be positive?

$$F1\ score = \frac{2 * precision * recall}{presicion + recall}$$

The weighted average of precision and recall

Precision, recall, and F1 score: example



		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes	4949(TP)	0(FN)
	Class=No	51(FP)	0(TN)

$$Precision = \frac{a}{a+c} = \frac{4949}{4949+51} = 0.99$$

Among the predicted positive samples, how many of them are correct?

$$Recall = \frac{a}{a+b} = 1$$

How many actual positive samples are predicted to be positive?

$$F1\ score = \frac{2 * precision * recall}{presicion + recall} = 0.995$$

The weighted average of precision and recall

Still maybe misleading for imbalanced data

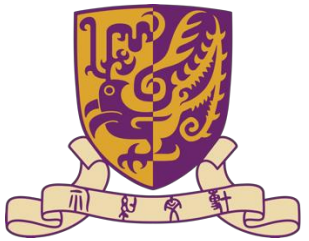
Balanced accuracy

		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes	4949(TP)	0(FN)
	Class=No	51(FP)	0(TN)

$$Balanced\ accuracy = 0.5 * \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) = 0.5$$

Personally, if I know it's an imbalanced dataset, I will look at the confusion matrix directly

Binary classification evaluation



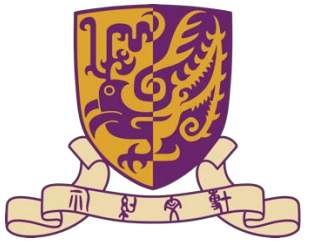
- Accuracy
- Precision
- Recall
- F1-score
- Balanced accuracy
- …

		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes	2(TP)	0(FN)
	Class=No	50(FP)	50(TN)

Is this the prediction performance terrible?

https://scikit-learn.org/stable/modules/model_evaluation.html

Binary classification evaluation



- Accuracy
- Precision
- Recall
- F1-score
- Balanced accuracy
- …

		Predicted class	
		Class=Yes	Class=No
Actual class	Class=Yes	2(TP)	0(FN)
	Class=No	50(FP)	50(TN)

Is this the prediction performance terrible?

What if this is for rare cancer pre-screening?

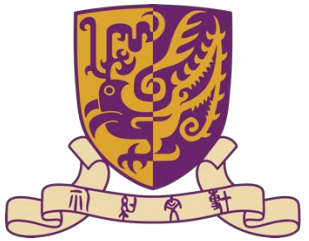
Value is not absolute. Context matters.

https://scikit-learn.org/stable/modules/model_evaluation.html

To make you awake

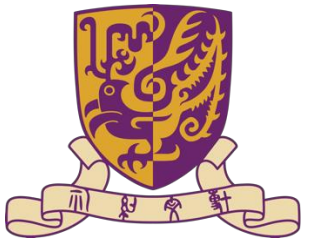
https://ureply.mobi/teacher

Today's agenda



- Performance evaluation
  - Binary classification evaluation
- Cross-validation
- Multi-class classification
- Clustering evaluation

The standard procedure of KNN



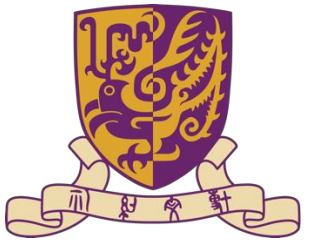
❖ Suppose we have chosen the distance metric and K

- ➢ Normalization
- ➢ Compute distances
- ➢ Identify the K most similar data
- ➢ Take their class out and find the mode class

Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

How to choose K, based on what we have?

What do we have?



What do we have?

❖The data is all we have

Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

How to find a good K for KNN with the below data?

- What is a good K?
  - The K can give us good prediction accuracy

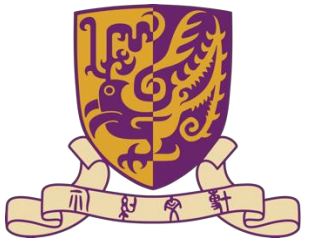
How many neighbours

- Problem: we do not have the label for testing data

Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

- Solution: use part of the training data as the testing data
  - Use each part one by one
  - Calculate the average over all the parts

K=1



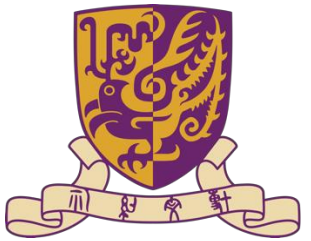
Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

P1: P4—M
P2: P3—M
P3: P2—F
P4: P1—M

Accuracy=0.5

	P1	P2	P3	P4
P1	0	0.875	0.5	0.375
P2	0.875	0	0.375	1
P3	0.5	0.375	0	0.75
p4	0.375	1	0.75	0

K=3



Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

P1: M
P2: M
P3: M
P4: M

	P1	P2	P3	P4
P1	0	0.875	0.5	0.375
P2	0.875	0	0.375	1
P3	0.5	0.375	0	0.75
p4	0.375	1	0.75	0

Accuracy=0.75

Model selection

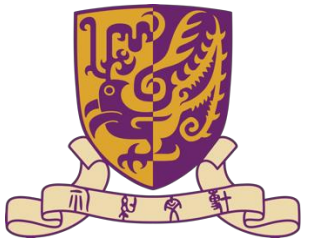
❖K=1: 0.5

❖K=3: 0.75

Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??

	P1	P2	P3	P4
P1	0	0.875	0.5	0.375
P2	0.875	0	0.375	1
P3	0.5	0.375	0	0.75
p4	0.375	1	0.75	0

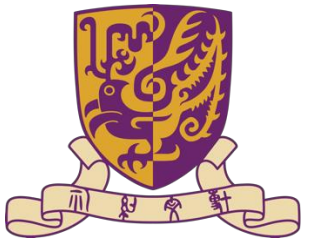
Cross-fold validation



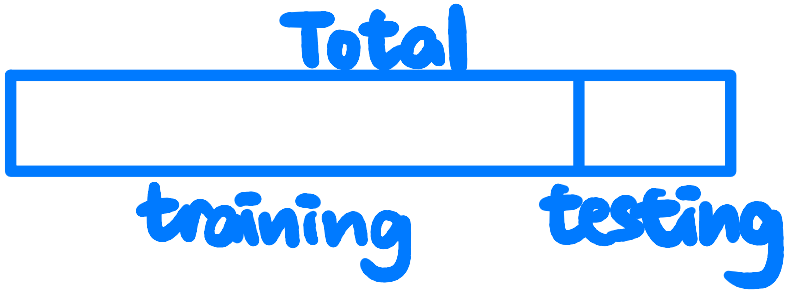
- Cross-validation/rotation estimation, is a technique for assessing how the results of a machine learning analysis will generalize to an independent data set
  - A procedure to measure the performance of models

- One round of cross-validation involves partitioning a set of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the testing set)

n-fold cross-validation



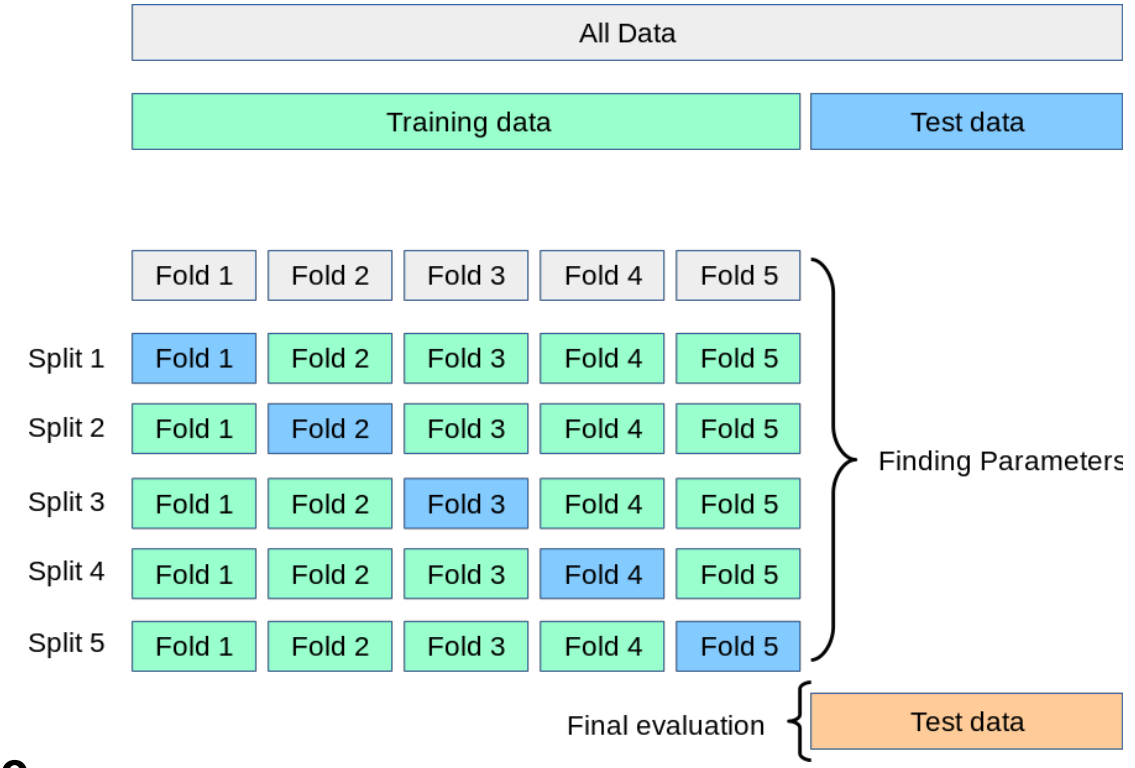
- Idea: train multiple times, leaving out a disjoint subset of data each time for validation. Average the validation set accuracies



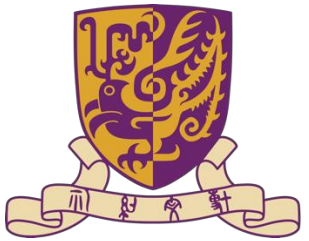
- Process:
  - Randomly partition data into n disjoint subsets
  - For i = 1 to n
    - Validation Data = i-th subset
    - h <- classifier trained on all data except for Validation Data
    - Accuracy(i) = accuracy of h on Validation Data
  - Final Accuracy = mean of the n recorded accuracies

5-fold cross-validation

- 10 data points:
  - P1-P10
- 5-fold
  - P1-2, P3-4, P5-6, P7-8, P9-10
  - The grouping can be random
- Procedure
  - P1-2's results based on the model from P3-10
  - ...
  - P9-10's results based on the model from P1-8
  - Averaging



Leave-one-out cross-validation



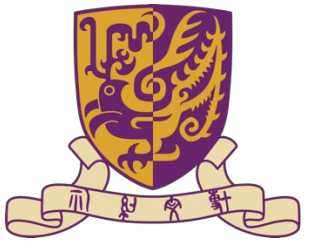
- Idea: a special case of n-fold cross-validation, where n = N

- Process:
  - Partition data into N disjoint subsets, each containing one data point
  - For i = 1 to N
    - Validation Data = i-th subset
    - h <- classifier trained on all data except for Validation Data
    - Accuracy(i) = accuracy of h on Validation Data
  - Final Accuracy = mean of the N recorded accuracies

To make you awake

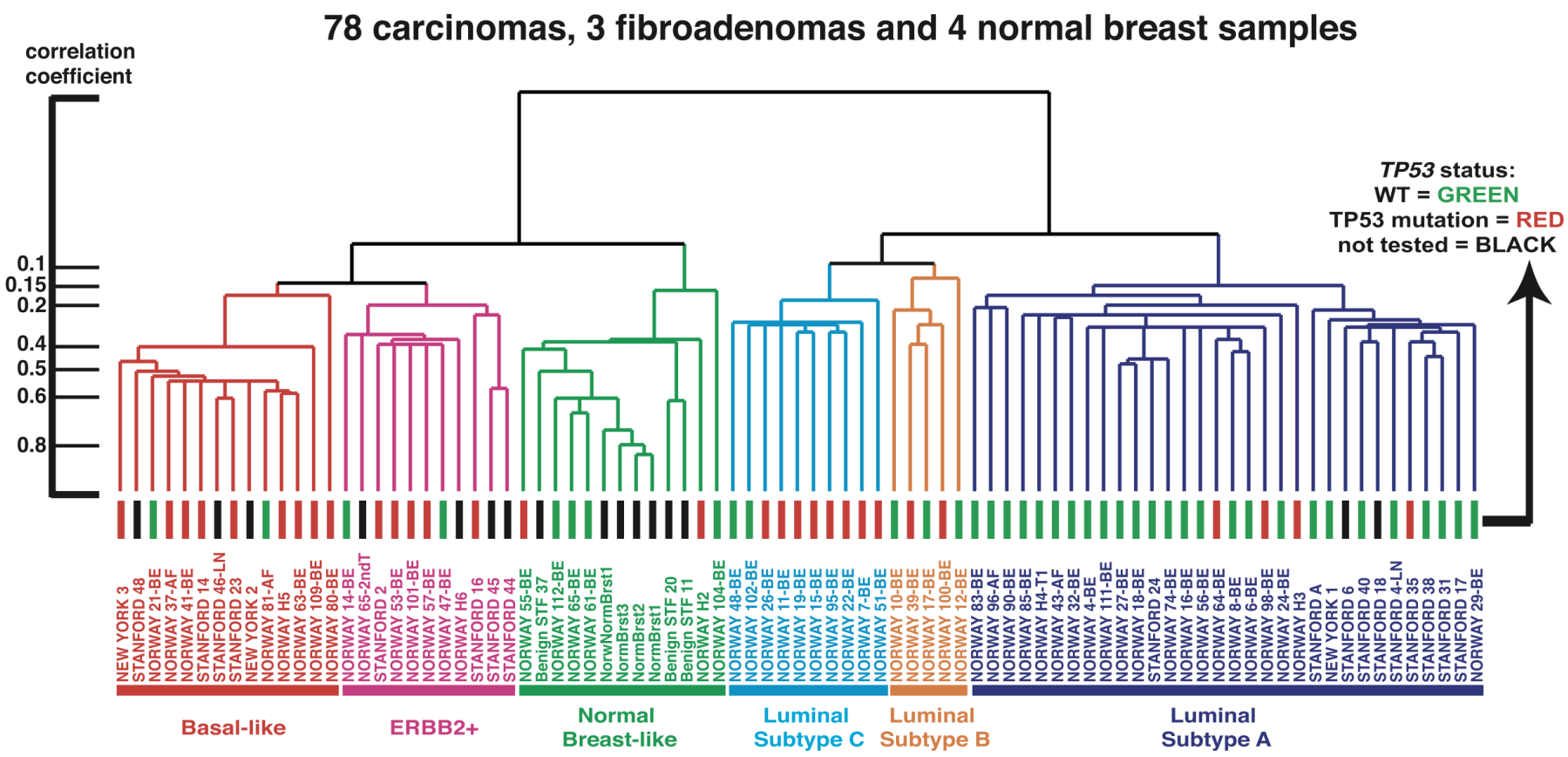
https://ureply.mobi/teacher

Today's agenda



- Performance evaluation
  - Binary classification evaluation
- Cross-validation
- Multi-class classification
- Clustering evaluation

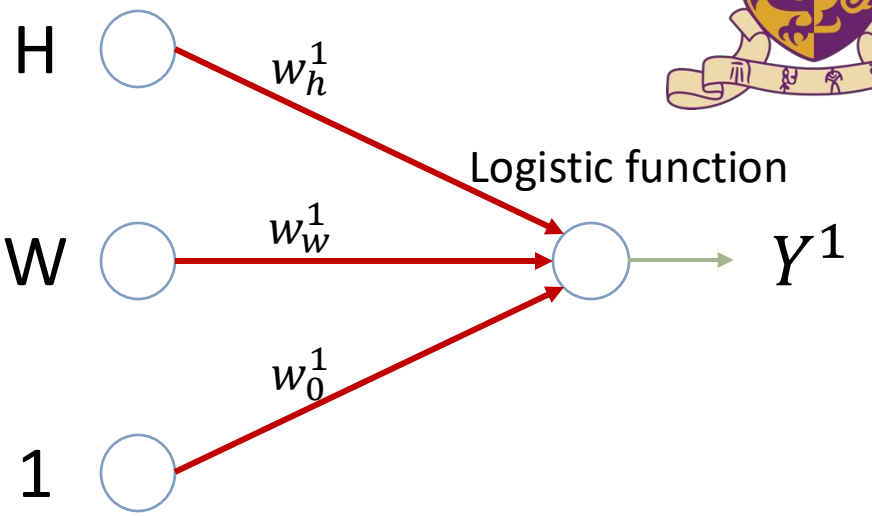
Multi-class classification

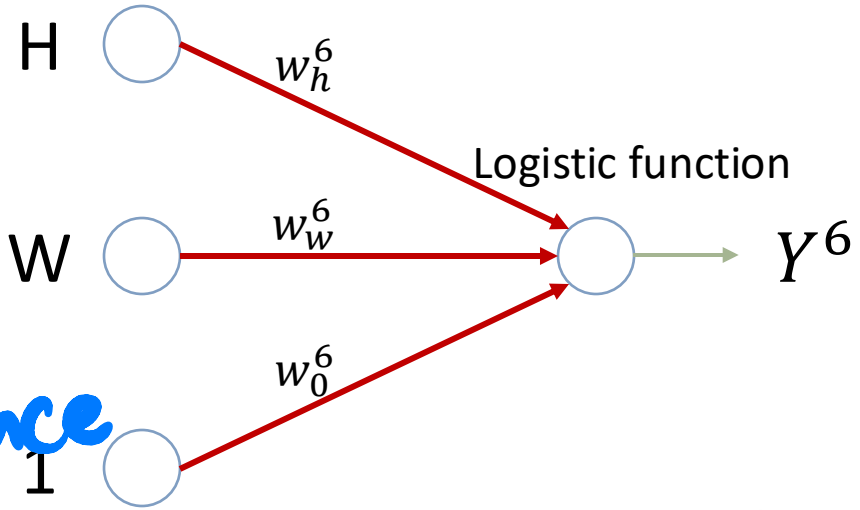


What if we have more than 2 classes? Here we have 6 classes

Multi-class classification

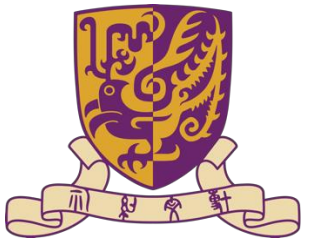
- Classify into sport interest groups
  - Basketball, football, tennis…
- For KNN, it is trivial
  - No need to change the algorithm
- For logistic regression, we need some change
  - Build a logistic regression for each class
  - When predicting, we assign class with highest value
  - When training, we train 3*6=18 parameters





can also get confidence

Multi-class evaluation



- Still using accuracy, precision, recall, F1 score and so on
  - Considering each class as a binary classification problem

- How to aggregate multiple values into one value?

$$Macro - average = \frac{0.9 + 0.95 + \cdots + 0.7 + 0.2}{6} = 0.73$$

$$Micro - average = \frac{0.9 * 150 + \cdots + 0.2 * 10}{150 + \cdots + 10} = 0.85$$

Class	Accuracy	Cells
1	0.9	150
2	0.95	50
3	0.85	100
4	0.8	40
5	0.7	20
6	0.2	10

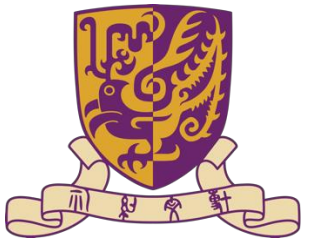
The low-performance of small classes will show up in Macro-average

More criteria at: https://scikit-learn.org/stable/modules/model_evaluation.html

To make you awake

https://ureply.mobi/teacher

Today's agenda



- Performance evaluation
  - Binary classification evaluation
- Cross-validation
- Multi-class classification
- Clustering evaluation

Clustering evaluation

Unsupervised learning - training data can be unlabelled

Clustering is different from classification

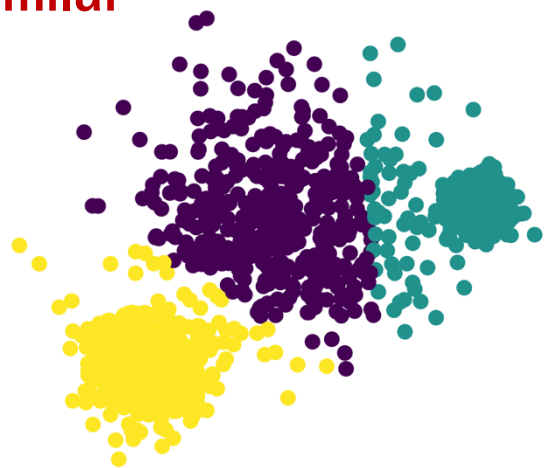
1111000
0000111

The same clustering results

In classification, we are correct for a cancer cell only if we predict it as cancer cell

In clustering, we are correct as long as similar cells are in the same cluster

A messy classification but a good clustering

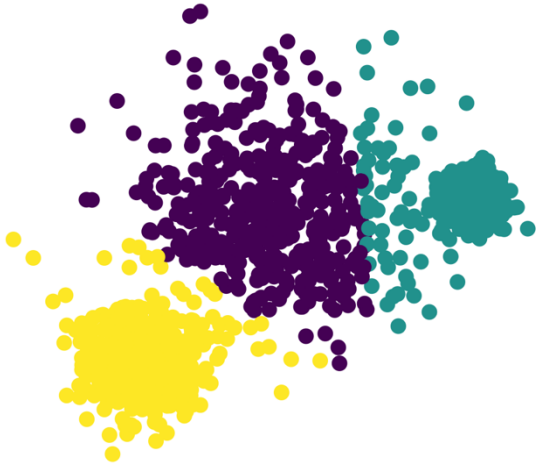


True label

Predicted label

How to evaluate clustering?

- In clustering, we are correct as long as two similar cells are in the same cluster
- We should evaluate a pair of cells
- We also have a confusion matrix



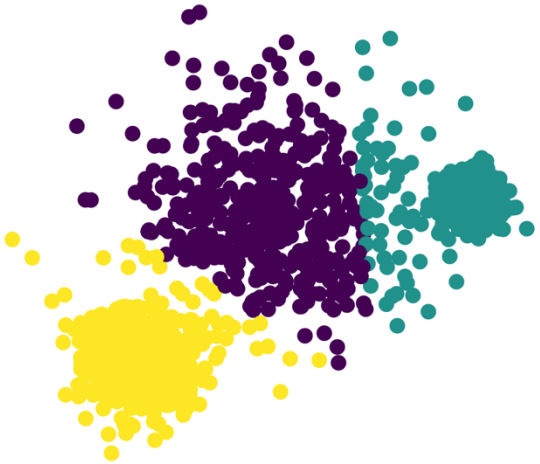
True clusters

Predicted clusters

		Predicted clusters	
		The same	Not the same
Actual clusters	The same	a(TP)	b(FN)
	Not the same	c(FP)	d(TN)

How to evaluate clustering?

	Predicted clusters		
Actual clusters		The same	Not the same
	The same	a(TP)	b(FN)
	Not the same	c(FP)	d(TN)



True clusters

Predicted clusters

For all the pairs in the dataset (how many do we have?):

a: the number of pairs are in the same cluster in the True clusters and also assigned to one cluster in the Predicted clusters

b: the number of pairs are in the same cluster in the True clusters and also assigned to different clusters in the Predicted clusters

c: the number of pairs are in different clusters in the True clusters and also assigned to one cluster in the Predicted clusters

d: the number of pairs are in different clusters in the True clusters and also assigned to different clusters in the Predicted clusters

Evaluate clustering: Rand index

	Predicted clusters		
Actual clusters		The same	Not the same
	The same	a(TP)	b(FN)
	Not the same	c(FP)	d(TN)

True clusters

Predicted clusters

Rand index, R:

$$R = \frac{a+d}{a+b+c+d} = \frac{a+d}{Number\ of\ all\ the\ pair\ combinations}$$

$$Pairs = \binom{n}{2} = \frac{n*(n-1)}{2}$$

n: Total number of points

Rand index: an example

Cell	C1	C2	C3	C4	C5
Real cluster	0	0	0	1	1
Predicted cluster	2	2	3	3	3

How many pairs?

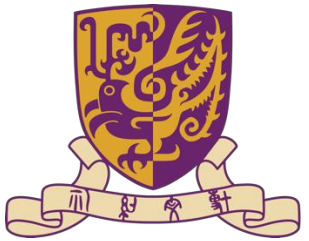
$$Pairs = \binom{5}{2} = \frac{5 * (5-1)}{2} = 10$$

Rand index?

$$R = \frac{a+d}{a+b+c+d} = \frac{6}{10} = 0.6$$

Pair	Real	Predicted	Results
C1, C2	Same	Same	✓
C1, C3	Same	Different	×
C1, C4	Different	Different	✓
C1, C5	Different	Different	✓
C2, C3	Same	Different	×
C2, C4	Different	Different	✓
C2, C5	Different	Different	✓
C3, C4	Different	Same	×
C3, C5	Different	Same	×
C4, C5	Same	Same	✓

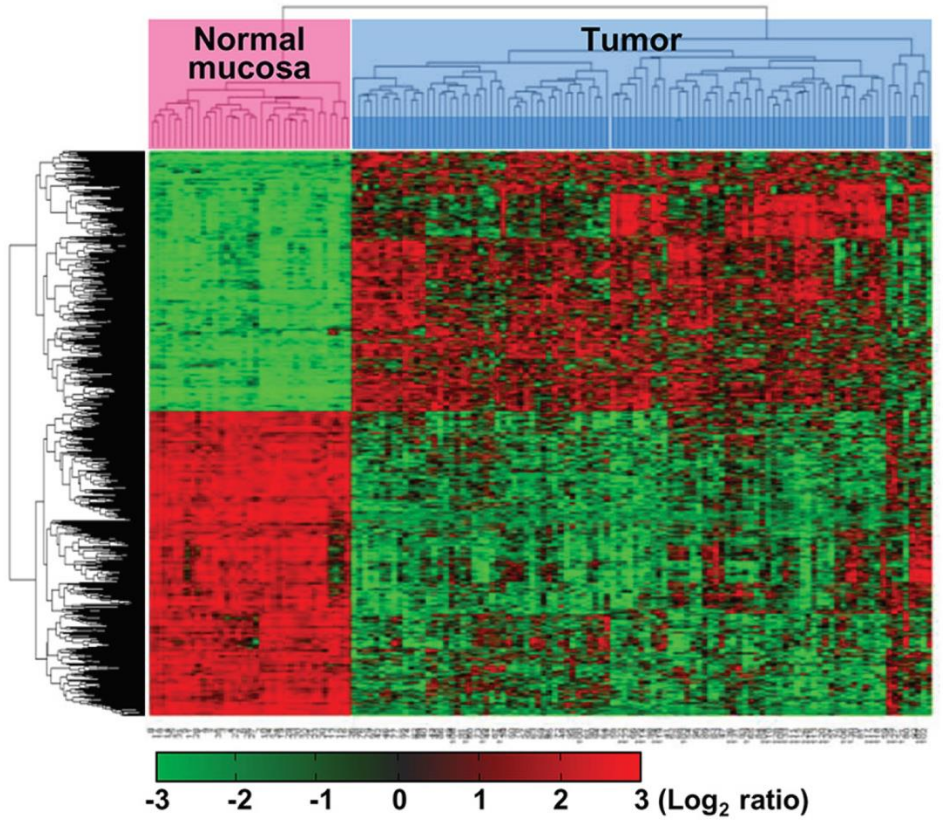
More clustering performance evaluation



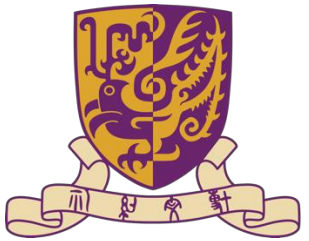
- https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

Potential project-2,3

- Data preprocessing for the gene expression matrix
  - Data collecting and merging (if needed)
  - Exploration
  - Visualization
  - Data cleaning
  - Dimension reduction (next lecture)
  - Get distance matrix
  - Perform classification/clustering
  - Performance evaluation



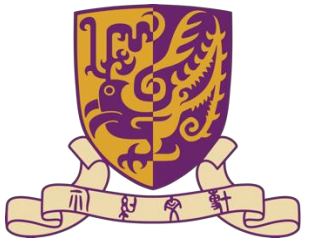
Model evaluation in Python



- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

```
>>> from sklearn.metrics import classification_report
>>> y_true = [0, 1, 2, 2, 2]
>>> y_pred = [0, 0, 2, 2, 1]
>>> target_names = ['class 0', 'class 1', 'class 2']
>>> print(classification_report(y_true, y_pred, target_names=target_names))
              precision    recall  f1-score   support

     class 0       0.50      1.00      0.67         1
     class 1       0.00      0.00      0.00         1
     class 2       1.00      0.67      0.80         3

    accuracy                           0.60         5
   macro avg       0.50      0.56      0.49         5
weighted avg       0.70      0.60      0.61         5

>>> y_pred = [1, 1, 0]
>>> y_true = [1, 1, 1]
>>> print(classification_report(y_true, y_pred, labels=[1, 2, 3]))
              precision    recall  f1-score   support

           1       1.00      0.67      0.80         3
           2       0.00      0.00      0.00         0
           3       0.00      0.00      0.00         0

   micro avg       1.00      0.67      0.80         3
   macro avg       0.33      0.22      0.27         3
weighted avg       1.00      0.67      0.80         3
```

Model evaluation in Python



- https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

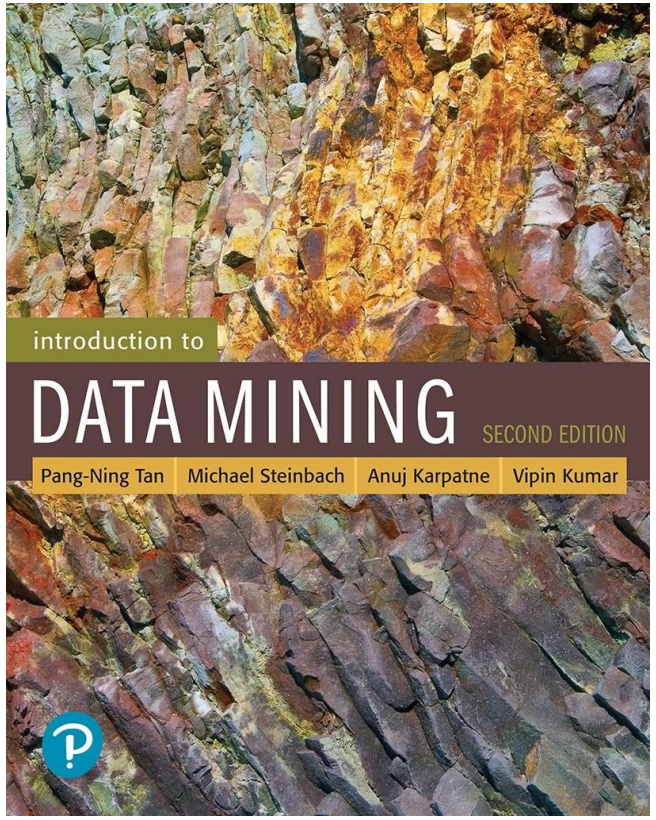
```
>>> from sklearn import metrics
>>> labels_true = [0, 0, 0, 1, 1, 1]
>>> labels_pred = [0, 0, 1, 1, 2, 2]
>>> metrics.rand_score(labels_true, labels_pred)
0.66...
```

- https://scikit-learn.org/stable/modules/cross_validation.html

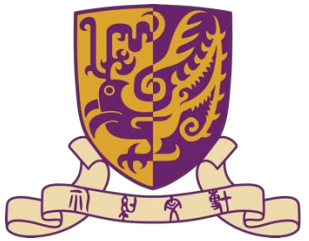
```
>>> from sklearn.model_selection import cross_val_score
>>> clf = svm.SVC(kernel='linear', C=1, random_state=42)
>>> scores = cross_val_score(clf, X, y, cv=5)
>>> scores
array([0.96..., 1. , 0.96..., 0.96..., 1. ])
```

Resource and uncovered topics

- Introduction to data mining: Chapter 4.5 & 4.6 & 5.7 & 5.8 & 8.5
- Bootstrap
- Overfitting and generalization
- Other clustering and classification methods
- Comparison between different methods
  - Clustering
  - Classification

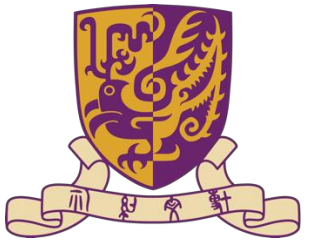


Post-lecture survey

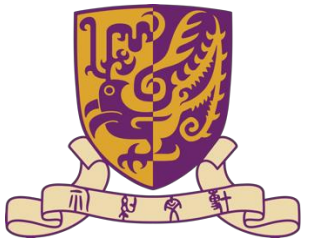


- https://forms.gle/V44GjermmJcnjhMT6

Next time



- Dimension reduction



Thank you!

Yu LI (李煜)

liyu95.com

liyu@cse.cuhk.edu.hk

The Artificial Intelligence in Healthcare (AIH) Group

Department of Computer Science and Engineering (CSE)

The Chinese University of Hong Kong (CUHK)