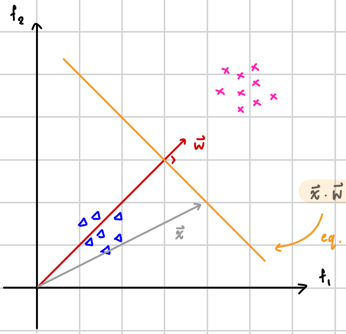
SVM

- principle · maximize margin (m)
- example – 2 features data



$\frac{\vec{x}\cdot\vec{w}}{\|\vec{w}\|} = c$

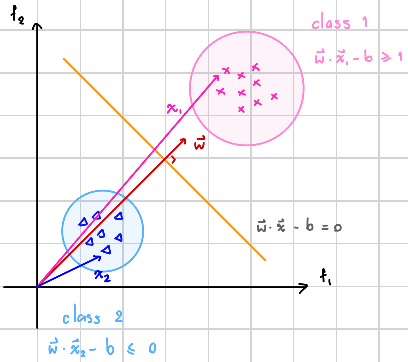
$\vec{x}\cdot\vec{w} = c\,\|\vec{w}\|$

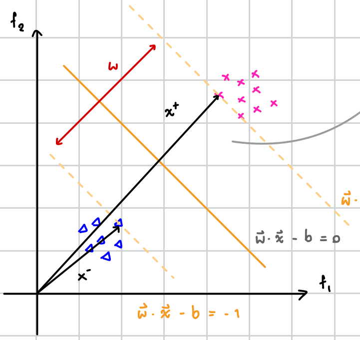
$\vec{x}\cdot\vec{w} = b$

$\vec{x}\cdot\vec{w} - b = 0$ ⟶ $\vec{w}$ is weight matrix

eq. of line

↳ 2 features : $\vec{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$





width (w) $= (x^+ - x^-)\cdot\frac{\vec{w}}{\|\vec{w}\|}$

sub

$= \frac{x^+\vec{w} - x^-\cdot\vec{w}}{\|\vec{w}\|} \longrightarrow \frac{1-b-(-1-b)}{\|\vec{w}\|}$

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

↳ try to max this

} same as " try to min $\frac{1}{2}\|\vec{w}\|^2$ "

$\vec{w}\cdot\vec{x} - b = 1$

} $y(\vec{w}\cdot\vec{x} - b) \geq 1$

< constrain >

Lagrange multiplier

$\mathcal{L} = \frac{1}{2}\vec{w}\cdot\vec{w} - \sum \lambda_i(y_i(\vec{w}x_i + b))$

↳ solve for $\vec{w}$, b

Decision Tree

- information gain · entropy(S) = $\sum -p_i \log_2 p_i$
  - $p_i < 1$, $\log p_i < 0$
  - ex

    all use same size (2 bits)

data	prop	encode
cations	1/2	00
ready	1/4	01
go	1/8	10
back	1/8	11

    $p_i \times \log_2 p_i$

    $S_1 = \frac{1}{2} \cdot \log_2 \frac{1}{2} + \ldots$

    $= 2$

    important use less

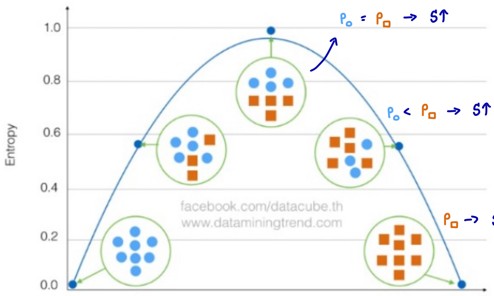
data	prop	encode
cations	1/2	0
ready	1/4	00
go	1/8	000
back	1/8	001

    $S_2 = \frac{1}{2} \log_2 \frac{1}{2} + \frac{1}{4} \log \frac{1}{4} + \ldots$

    $= 1.75$

    $S_1 > S_2$
    ↳ case 1 have more randomness
    ↳ case 2 more deterministic

  - Apply



$P_{\circ} = P_{\square} \rightarrow S\uparrow$

$P_{\circ} < P_{\square} \rightarrow S\uparrow$

$P_{\square} \rightarrow S_{max}$

use S as loss function
↳ $S\downarrow$ seperate class correctly

Decision Tree

- information gain · entropy $(S) = \sum -p_i \log_2 p_i$ $p_i < 1,\ \log p_i < 0$

- find $S(flu|headache)$

  ↳ entropy of flu given headache

CASE	TEMPERATURE	HEADACHE	NAUSEA	Flu
1	high	YES	no	YES
2	very_high	YES	YES	YES
3	normal	no	no	no
4	high	YES	YES	YES
5	high	no	YES	no
6	normal	YES	no	no
7	normal	no	YES	no

CASE	TEMPERATURE	HEADACHE	NAUSEA	Flu
1	high	YES	no	YES
2	very_high	YES	YES	YES
3	normal	no	no	no
4	high	YES	YES	YES
5	high	no	YES	no
6	normal	YES	no	no
7	normal	no	YES	no

- (flu, headache)

(yes, yes) → 3

(yes, No) → 1

$$-\frac{4}{7}\left[\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right]$$

(No, yes) → 0

(No, No) → 3

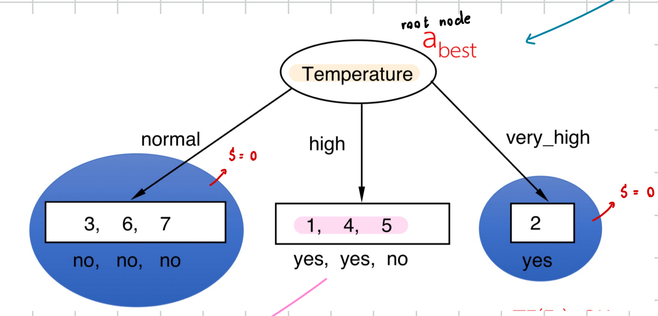
$$-\frac{3}{7}\left[\frac{0}{3}\log_2\frac{0}{3} + \frac{3}{3}\log_2\frac{3}{3}\right]$$

sum = 0.464 ← S(flu | headache)

↓ do the same for the other features

H (flu | Temp) = 0.394 → lowest : chosen as Root node

H (flu | Nausea) = 0.620

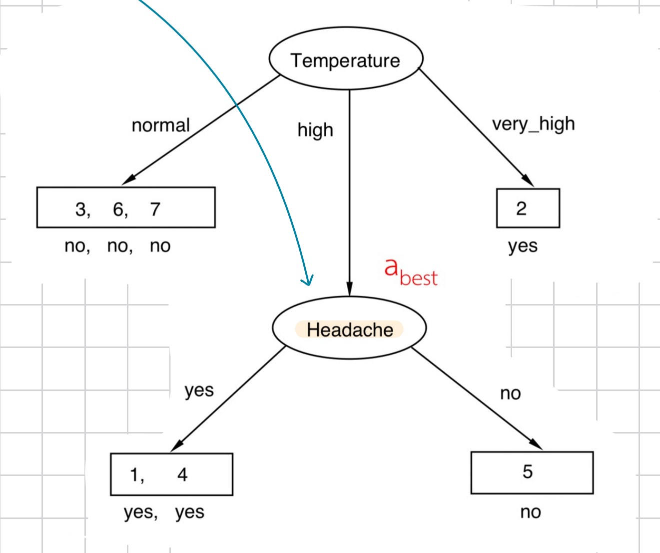


CASE	TEMPERATURE	HEADACHE	NAUSEA	Flu
1	high	YES	no	YES
2	very_high	YES	YES	YES
3	normal	no	no	no
4	high	YES	YES	YES
5	high	no	YES	no
6	normal	YES	no	no
7	normal	no	YES	no

1) $S(flu \mid headache) = -\frac{2}{3}\left[\frac{2}{2}\log_2\frac{2}{2} + 0\cdot\log 0\right] - \frac{1}{3}\left[0\cdot\log 0 + \frac{1}{1}\log\frac{1}{1}\right]$

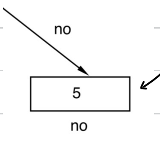
$= 0$ → lowest

2) $S(flu \mid nausea) = 0.67$



Decision Tree

- flaws → overfit → complex tree



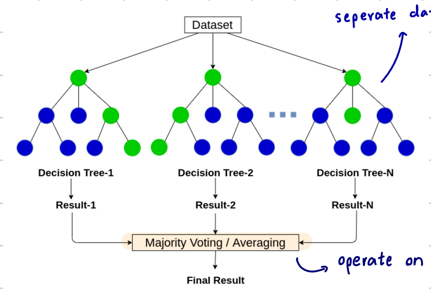
node with only one member

- Random forest
  - solve overfit problem

determine by human

seperate dataset to n group

↳ create tree of each group

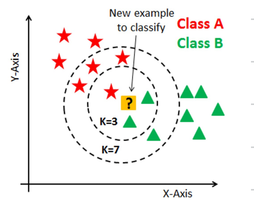


↳ operate on result of each tree

major voting (classification)

averaging (regression)

K - Nearest neighbor

- 

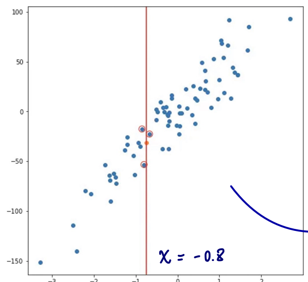
count the data belong to each class within k nearest unit

- very less computation, very simple
- large memory require

} flaws → overfit & memory issue

- k : amount of data take in to consider
  - ↳ k closest one
  - ↳ set by human

- Regression (not in exam)

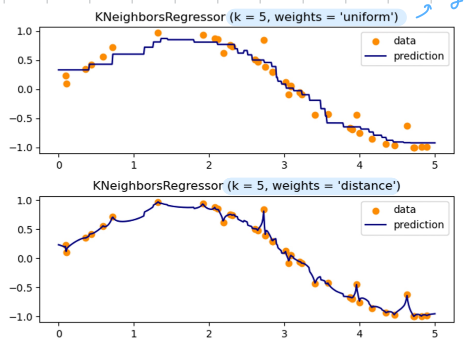


find the data close to line $x = -0.8$ ← example

- calculate average of those data

use to build prediction function

set by human



- more on regression
  - Logistic regression → can be used for classification
    - ↳ use sigmoid function

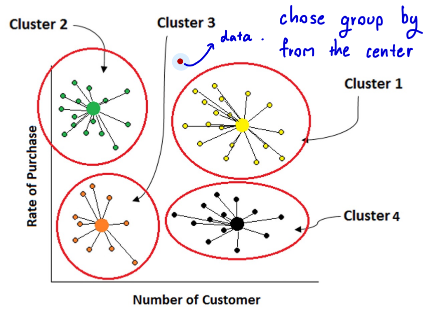
Other classification

- logistic : find seperate line with sigmoid function
- SVM : find line / plane with maximum margin
  - ↳ straight line / plane only
  - use kernel to transform raw input to feature space
    - ↳ technique to solve non-linear classification

supervised

Clustering

- one of unsupervised → can associate things
  - ↳ find relation between data
- can only seperate group not identify
  - ↳ AKA clustering
- idea : same class → Similar feature
- task
  - assigning group
  - find group mean (center point)



chose group by calculate distance from the center

method → K- mean , GMM

K- mean

- Euclidean distance
- $\|d\| = \left[(x_1 - x_2)^2 + (y_1 - y_2)^2 \right]^{1/2}$

1D : variance

2D : variance & covariance

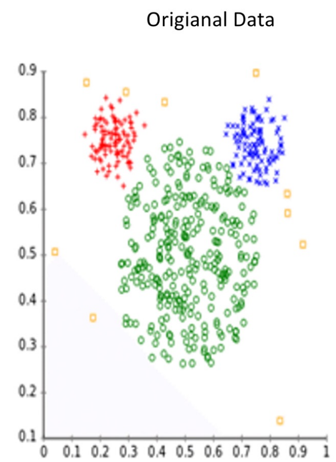
GMM : Gaussian mixture model

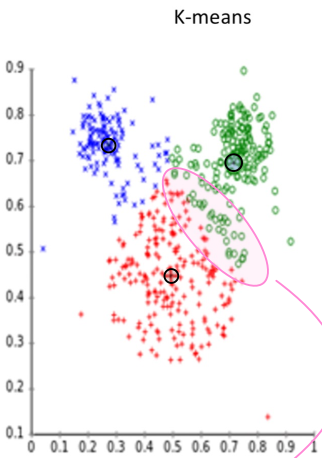
- probability model
- ↳ mahalanobis distance
  - ↳ based on gaussion distribution

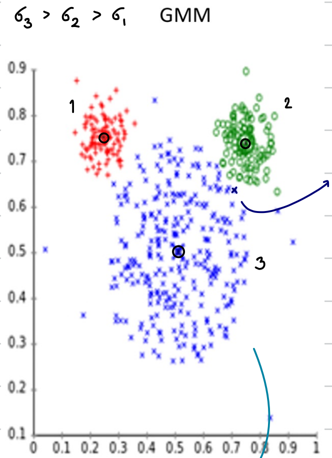
$$d_M = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \xrightarrow{1D} d_M = \left[\frac{(x-\mu)^2}{\sigma} \right]^{1/2}$$

covariance matrix

$\sigma_3 > \sigma_2 > \sigma_1$







$x - \mu_2 < x - \mu_3$

but $\sigma_2 < \sigma_3$

$\Downarrow$

$\frac{x - \mu_2}{\sigma_2} > \frac{x - \mu_3}{\sigma_3}$

value is significantly affected by variance

note that this is normally use with 1D with 2D like this must use the full eq.

error

- suit for data that feature is far apart
- regardless of group size only position

good for difference size data

↓

required more computation