# Classification & Regression (K-NN)

Qus How classification works? → +ve/-ve

$x_i$ → Text → Vector
(360K+) [BoW / tfidf / W2V]

{Amazon food ..... }
↓
real world eg
MNIST (PCA / tSNE)

## classification

work

364K+ $x_i \longrightarrow (v_i^{(x_i)}, +ve/-ve)$

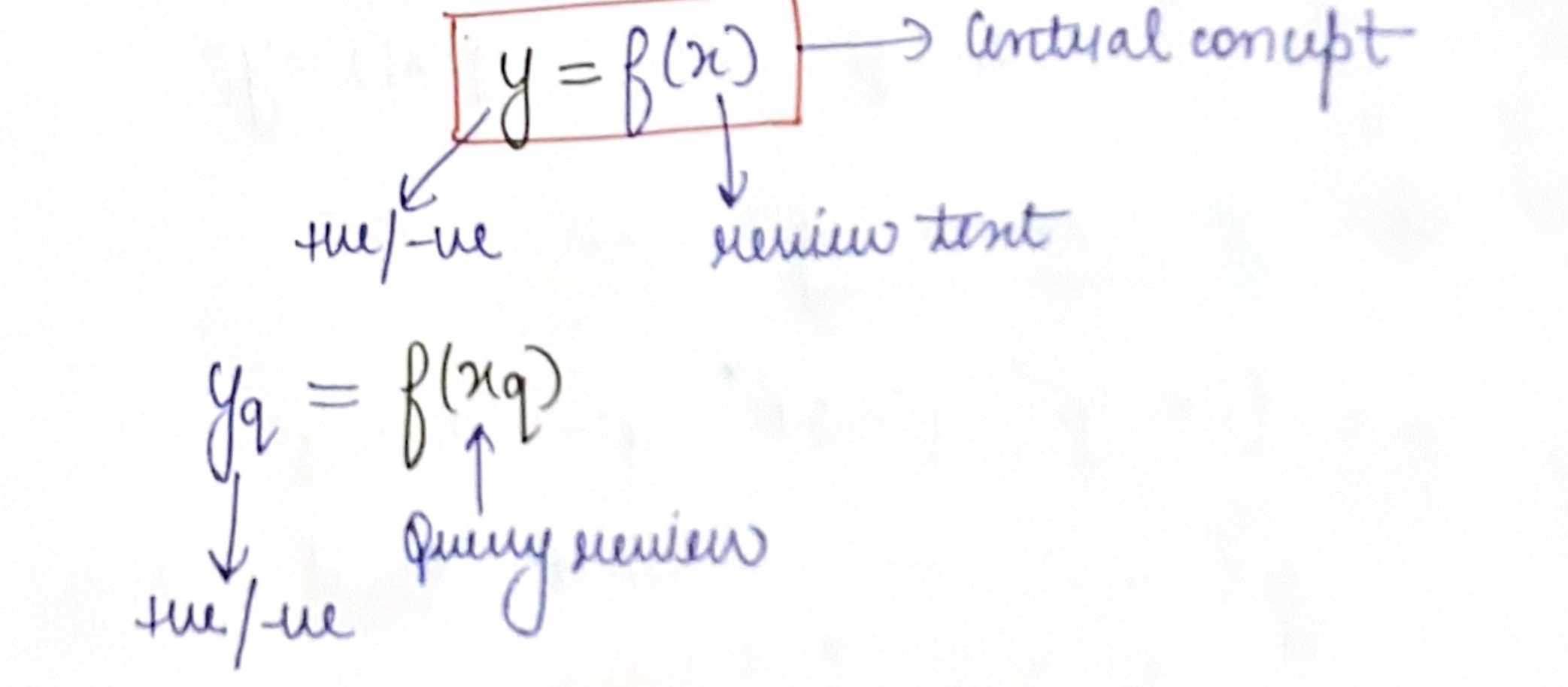
{ Given a new review determine / predict if review is -ve/+ve.

$x_q \longrightarrow$ +ve/-ve
classifying

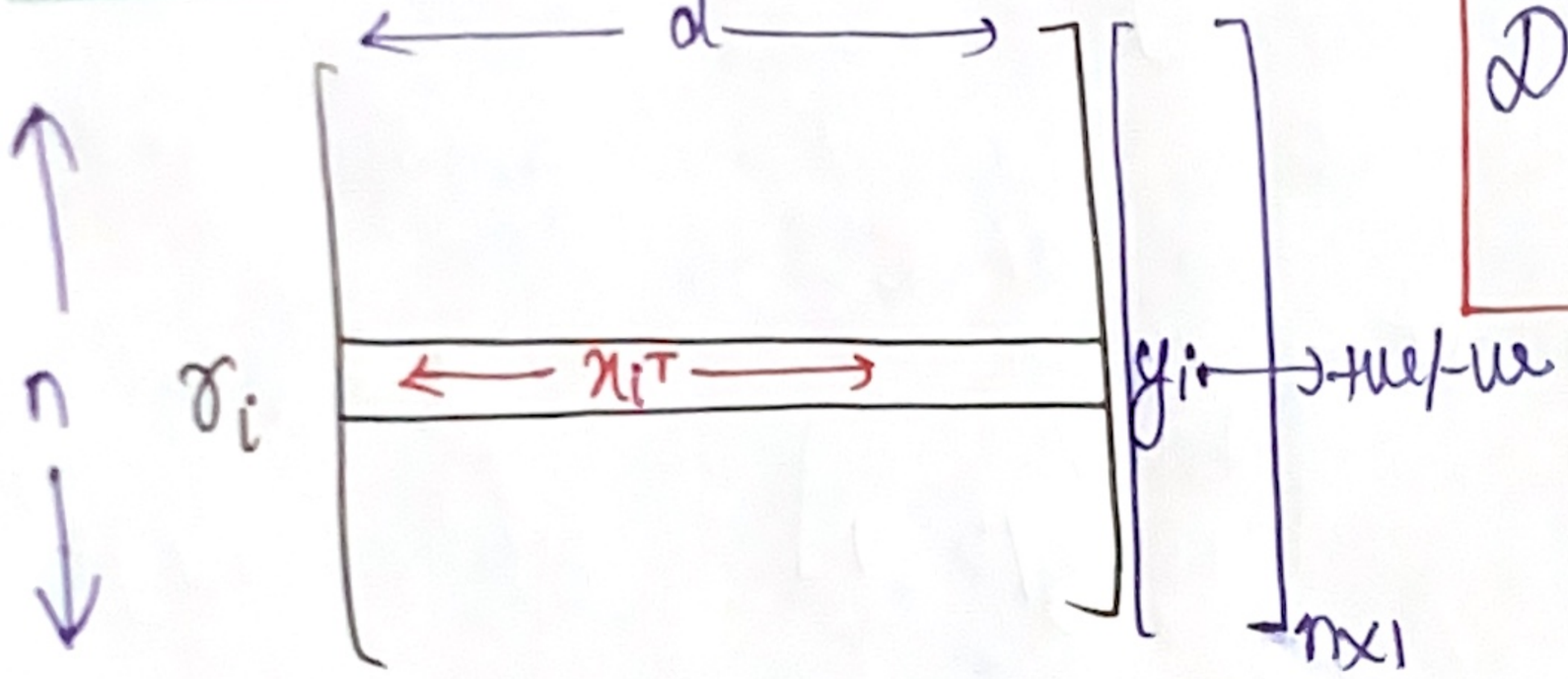
$y = f(x)$ → Actual concept
+ve/-ve ← review text

$y_q = f(x_q)$
↓ ↑
+ve/-ve Query review

## Classificaton algorithm work



## Notation




$$\mathcal{D} = \left\{ (x_i, y_i)_{i=1}^{n} \;\middle|\; x_i \in \mathbb{R}^d, \; y_i \in \{0,1\} \right\}$$

$x_i \rightarrow$ vector $(x_i)$
↳ column vector

↳ class(+ve/-ve)

$y = f(x)$

Set

$$\mathcal{D}_n = \left\{ (x_i, y_i)_{i=1}^{n} \;\middle|\; x_i \in \mathbb{R}^d, \; y_i \in \{0,1\} \right\}$$

such that

$y_i$ ∈ {0, 1} : 0 → -ve, 1 → +ve

$x_i \rightarrow x_i$

↳ $\{0,1\}$ $y_i$

**classification:**

$f(x_i) = y_i$

## Classification vs Regression

$$\mathcal{D} = \left\{ (x_i, y_i)_{i=1}^{n} \;\middle|\; x_i \in \mathbb{R}^d, \; y_i \in \{0,1\} \right\}$$

$y_i \in \{0,1\}$ ← Amazon food review

↓ ↓ → 2 classes

-ve +ve

↳ 2 class - classification / binary

MNIST:

$y_i \in \{0,1,2,3,4,5,6,7,8,9\}$

↳ 10-class / Multi class classification

What if $y_i \in R \longrightarrow$ Regression

↳ $y_i$ is no more part of small finite set of classes

$i = 1 \to 10K$

$x_i = \langle$weight, age, gender, race$\rangle$

↳ $y_i$ = height

↳ real no

(182.6cm, 152.7cm)

$f(x_i) = y_i$

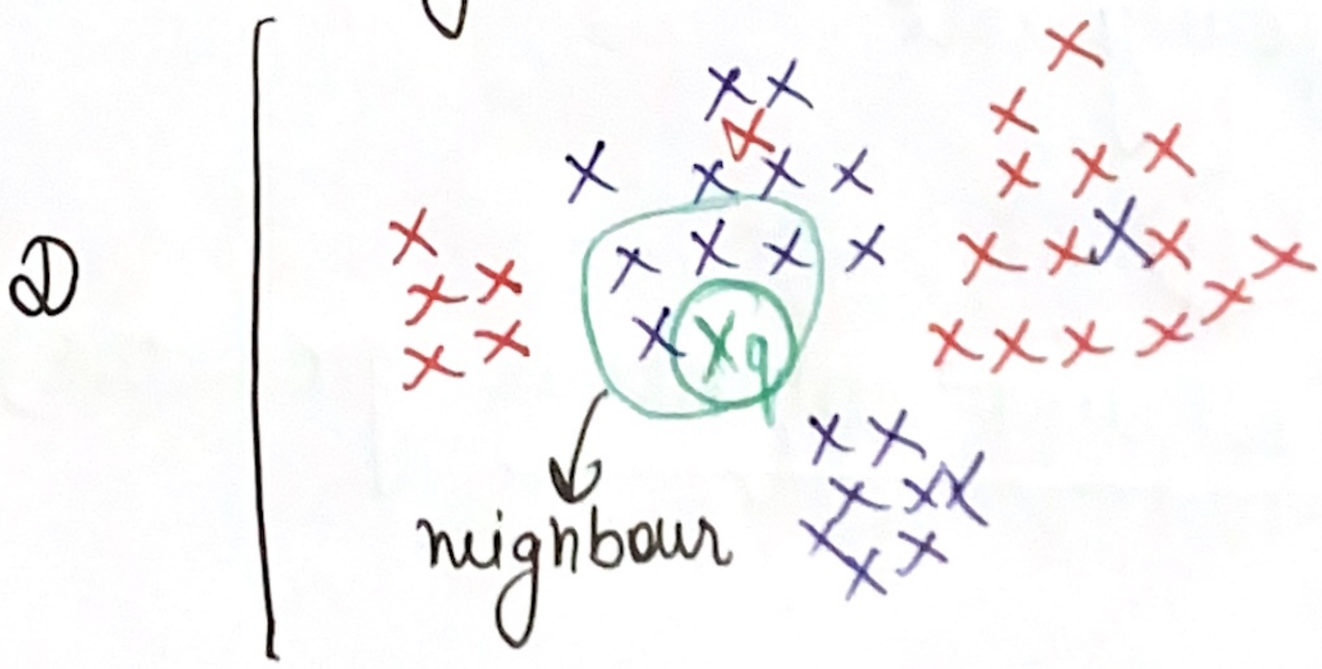
$y_i \in \{0, 1\} \longrightarrow$ Classification

$y_i \in R \longrightarrow$ Regression

K-Nearest Neighbours (Geometric)

(binary classification)

2D - toy dataset




x: +ve data pt

x: -ve class data pt

$D = \{(x_i, y_i) \mid x_i \in R^2, y_i \{0,1\}\}$

given $x_q$ find $y_q$ {0,1}

conclude $x_q \to$ blue (+ve)

① Find k-nearest points to $x_q$ in $D$

Let $K=3$ $\quad [x_1, x_2, x_3]$ – 3 neighbours to $x_q$

$\downarrow \quad \downarrow \quad \downarrow$

$y_1 \; y_2 \; y_3$

②

K=odd $\{y_1, y_2, y_3\} \rightarrow$ majority work / vote

$y_1, y_2, y_3 \rightarrow$ Majority vote

$+, +, + \rightarrow \oplus \; y_q$ = +ve (blue)

$+, +, - \rightarrow y_q$ = +ve

if K=4

$+, +, -, - \rightarrow \oplus$ or $\ominus$

---

$x_q \rightarrow$ K-NN of $x_q$

$x_1, x_2 - - x_K$

$\downarrow$

$y_1, y_2 - - y_K$

$\downarrow$

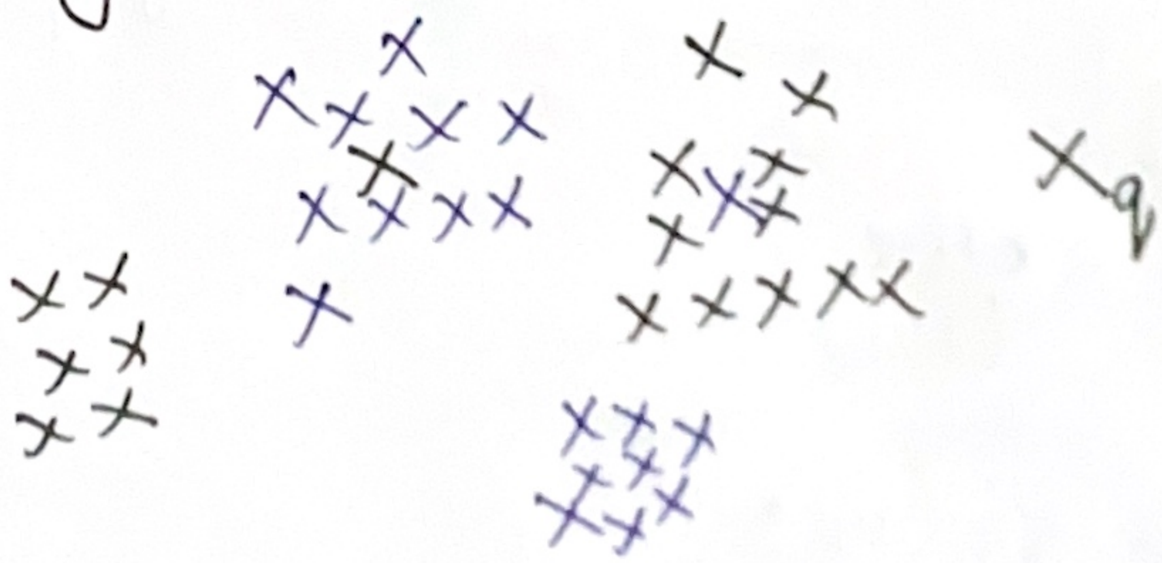
Majority vote

$\downarrow$

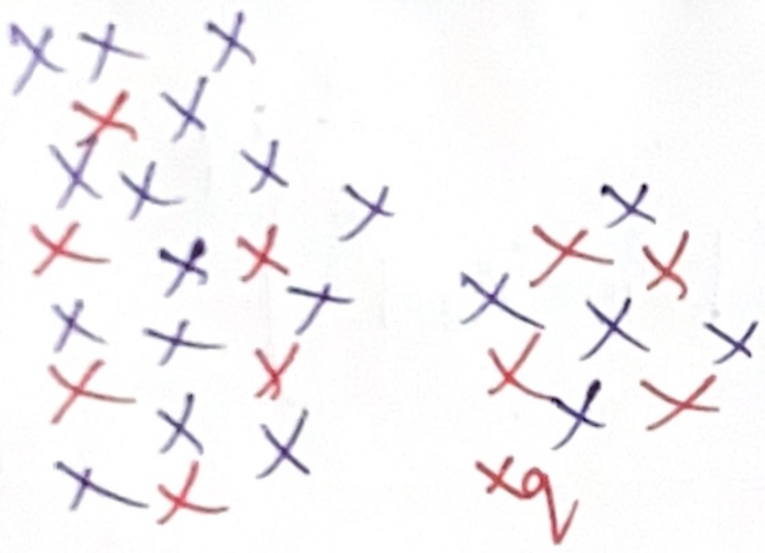
$y_q$

## Failure of K-Nearest Neighbour (geometric)

2D toy dataset



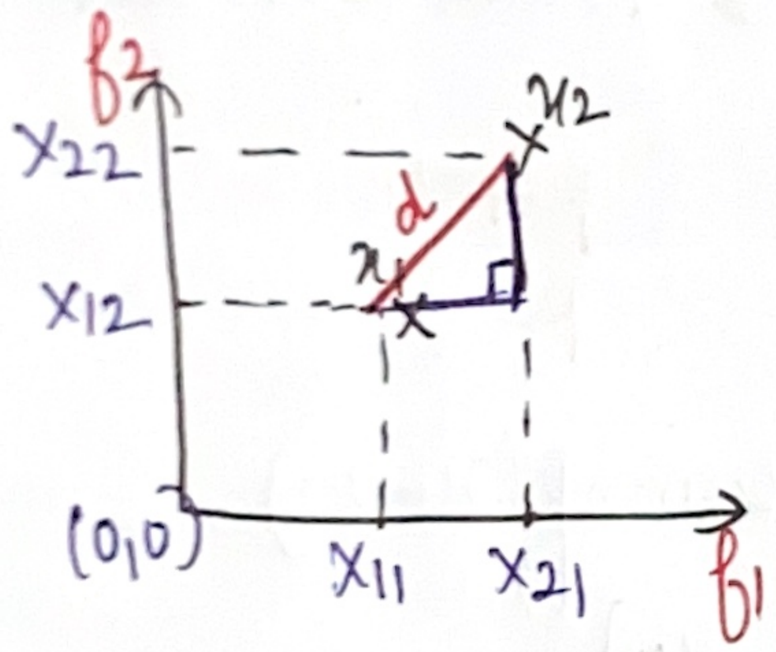

$x_q$ is very far away from other points



Data is very jumbled

KNN algorithm works ↓ does not

## Distance Measures




$x_1 = (x_{11}, x_{12})$ ($\beta_1$, $\beta_2$)

$x_2 = (x_{21}, x_{22})$

d = length of shortest line from $x_1$ to $x_2$

**Euclidean distance** ← $d = \sqrt{(x_{21}-x_{11})^2 + (x_{22}-x_{12})^2} = \|x_1 - x_2\|$ ↓ length

$x_i \in \mathbb{R}^d, \quad x_2 \in \mathbb{R}^d$

**Eucd. dist** $\|x_1 - x_2\|_2 = \left(\sum_{i=1}^{d} (x_{1i} - x_{2i})^2\right)^{1/2}$

$\|x_1 - x_2\|_2 \longrightarrow$ $L_2$ norm of vector

$\|x_1\|_2 =$ Euc dist of $x_1$ from origin $= \left(\sum_{i=1}^{d} x_{1i}^2\right)^{1/2}$

## Manhattan dist:

$L_1$ norm of vect $(x_1 - x_2)$ $\sum_{i=1}^{d} |x_{1i} - x_{2i}|$ ← absolute

$\|x_1 - x_2\|_1$

$\|x_1\|_1 = \sum_{i=1}^{d} |x_{1i}|$

Lp norms → Minkowski distance

$$||x_1 - x_2||_p = \left( \sum_{i=1}^{d} |x_{1i} - x_{2i}|^p \right)^{1/p}$$

p=2 → Minkowski dist → Euclidean dist

p=1 → " → Manhattan dist

$||x_1||_p = \left( \sum_{i=1}^{d} |x_{1i}|^p \right)^{1/p}$

↙ Lp norm

p≠0
p>0

Eucl dist $(x_1, x_2)$ = $L_2$ norm of $(x_1 - x_2)$

$= ||x_1 - x_2||$

dist $(x_1, x_2)$ [b/w two points]

norm → vector [b/w two vectors]

Hamming dist (boolean vector)

$x_1, x_2$ → boolean vector → Binary BoW

$x_1 = [0, 1, 1, 0, 1, 0, 0 \ldots]$

$x_2 = [1, 0, 1, 0, 1, 0, 1 \ldots]$

Hamming dist $(x_1, x_2)$ = # locations / dimensions where binary vectors differ

↳ 3

## Strings

$x_1 = a\ b\ c\ a\ d\ e\ f\ g\ h\ i\ k$ ← gene code

$x_2 = a\ c\ b\ a\ d\ e\ g\ f\ f\ h\ i\ k$ eg AGTC

hamming dist $(x_1, x_2) = 4$

## Cosine similarity & cosine distance

$x_1, x_2$

Similarity ↓ ↑

distance ↑ inc ↓ dec (opposite)

v. Similar
$CosSim(x_1, x_2) = +1$

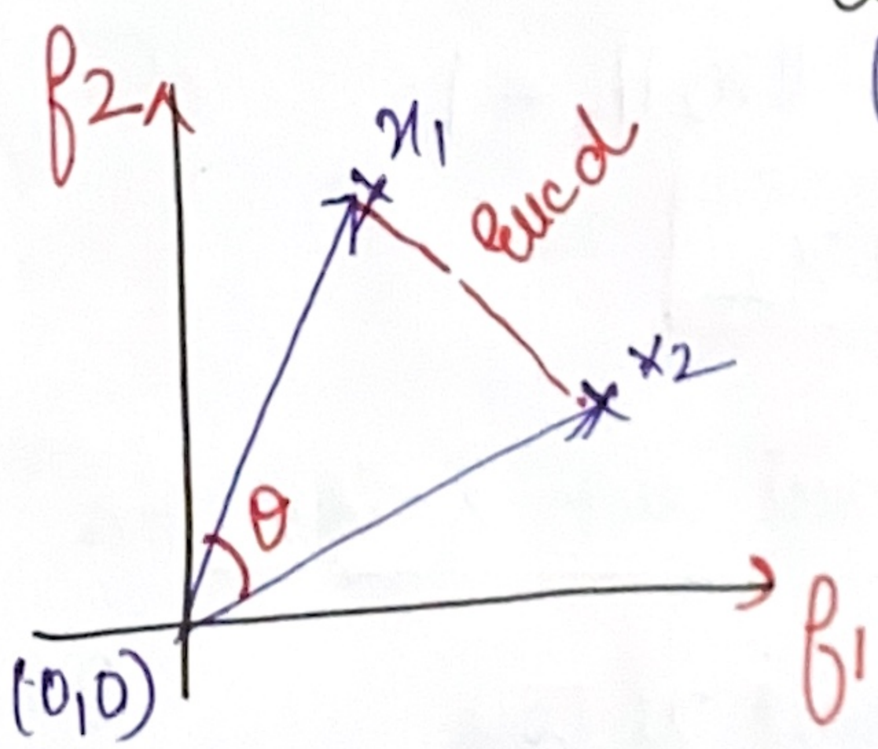
$1 - Cos\ Sim(x_1, x_2) = Cos\ dist(x_1, x_2)$

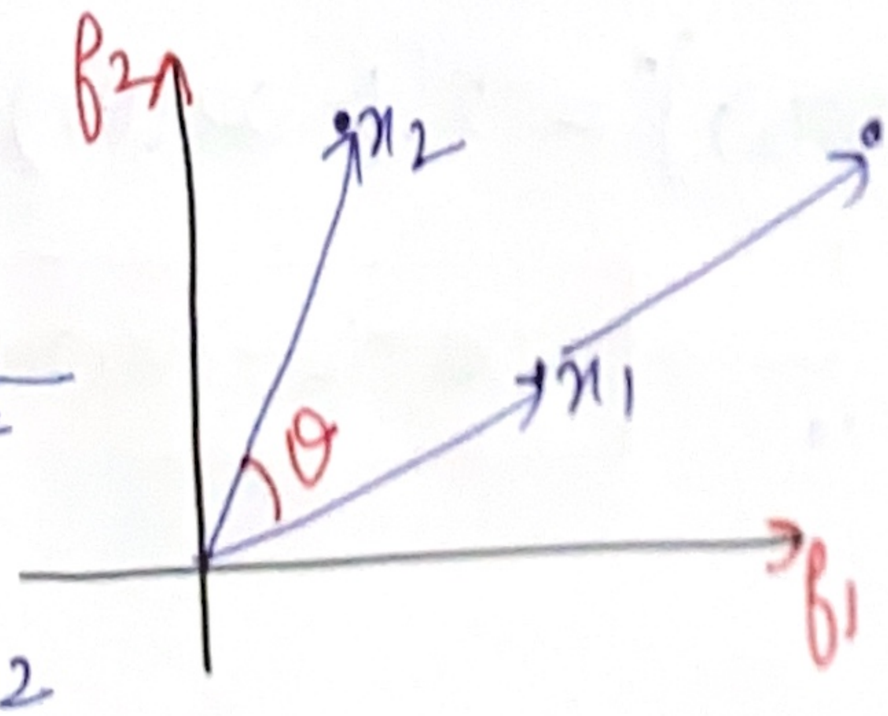
v. diff
$CosSim(x_1, x_2) = -1$

det $[-1, +1]$

$Cos\ Sim = \cos\theta$
$(x_1, x_2)$

$\theta$ : angle b/w $x_1$ & $x_2$




$Cos\ dist = 1 - \cos\theta$
$(x_1, x_2)$

Eucl dist
$d_{13} > d_{12}$

$Cos\ dist_{13} < Cos\ dist_{12}$

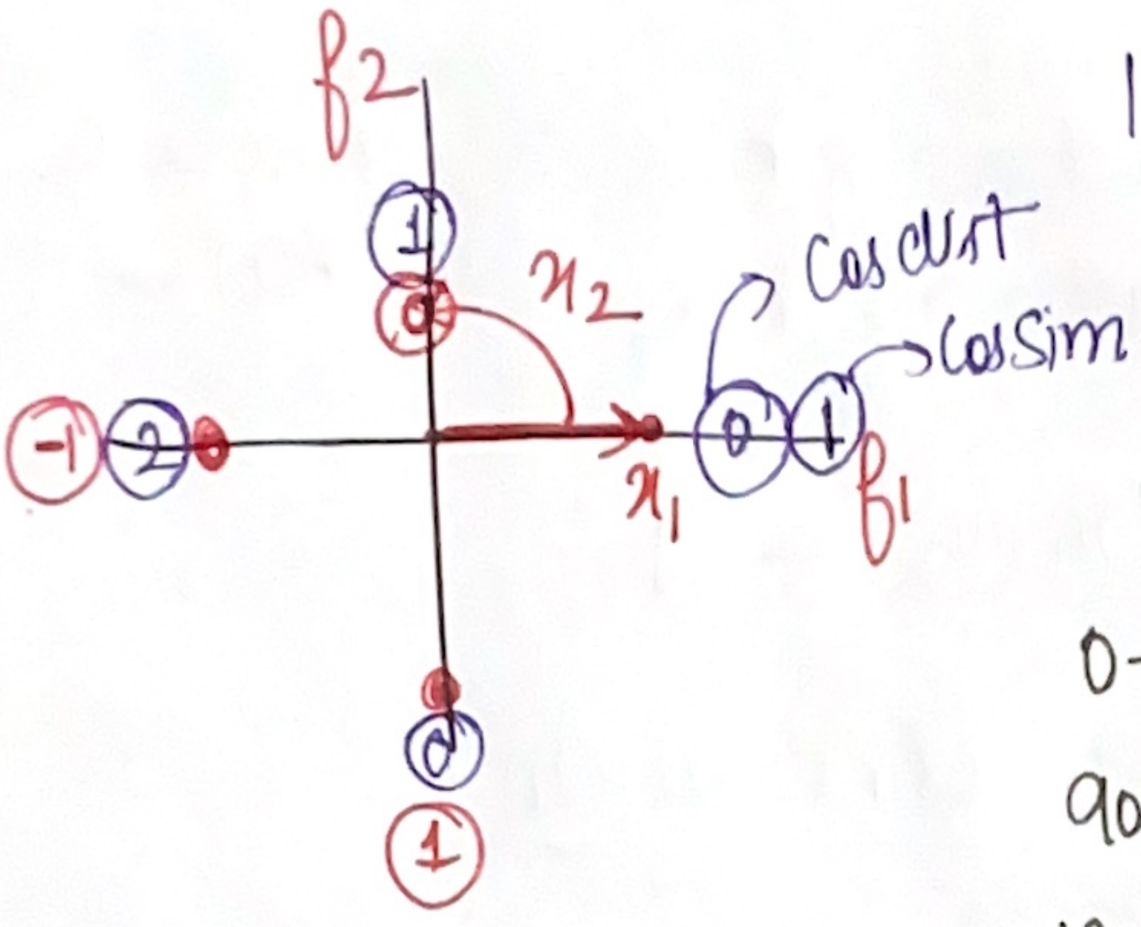
$Cos\ dist = 1 - 1 = 0$

$Cos\ Sim(x_1, x_2) = \cos\theta$

$Cos\ Sim(x_1, x_3) = 1$

$\theta x_1, x_3 = 0°$

$\cos 0° = 1$

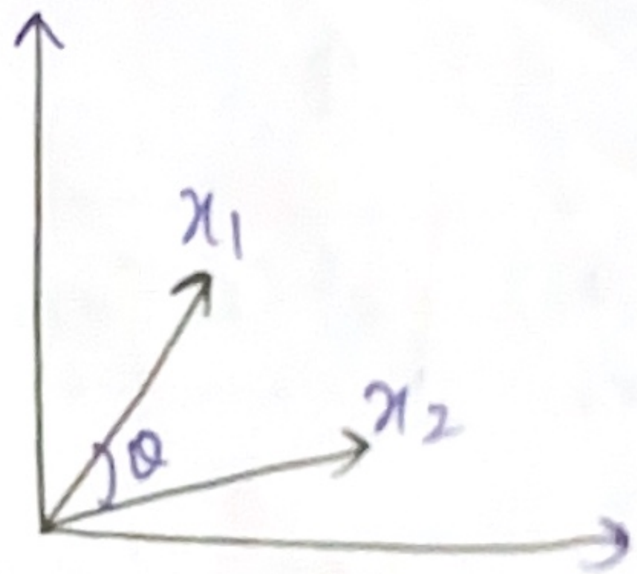



$1 - \text{CosSim}(x_1, x_2) =$

$\text{Cos dist}(x_1, x_2)$

| | dist |
|---|---|
| $0 \rightarrow 90^\circ$ | $0 \rightarrow 1$ |
| $90 - 180^\circ$ | $1 \rightarrow 2$ |
| $180^\circ - 270^\circ$ | $2 \rightarrow 1$ |
| $270^\circ - 360^\circ$ | $1 \rightarrow 0$ |

$$\text{Cos}(\theta) = \frac{x_1 . x_2}{\underbrace{\|x_1\|_2 \|x_2\|_2}_{\hookrightarrow \text{l2 norm}}}$$




① If $x_1$ & $x_2$ are unit vectors

$$\|x_1\|_2 = \|x_2\|_2 = 1$$

$$\boxed{\text{Cos}(\theta) = x_1 . x_2}$$

## Relationship b/w Eucl. dist & Cos-Sim

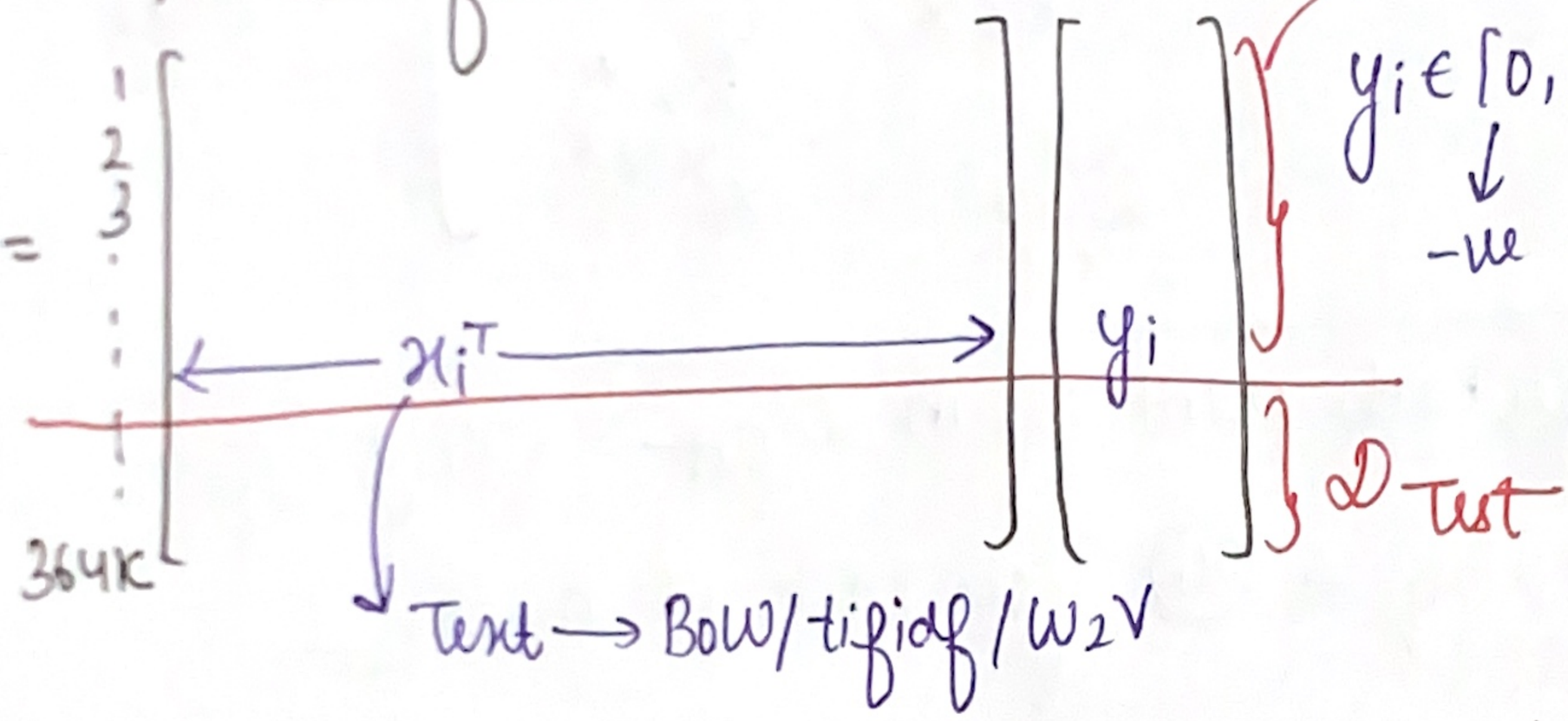
if $x_1$ & $x_2$ are unit vectors $\quad \theta = \text{angle b/w } x_1 \& x_2$

$$\left[\text{euc dist}(x_1, x_2)\right]^2 = 2(1 - \text{Cos}(\theta))$$

$$\hookrightarrow = \boxed{2\,\text{Cos dist}(x_1, x_2)}$$

# Measuring how good K-NN is

Amazon food reviews:




Problem: Given a new food review, what is its polarity (+ve/-ve)

$x_q \rightarrow y_q$

$x_q \rightarrow$ Text $\rightarrow x_q$

"Measure" k-NN → (k-NN + Majority vote)

$\mathcal{D}_n$ → $\mathcal{D}_{train}$, $\mathcal{D}_{Test}$

$\mathcal{D}_{train} \cup \mathcal{D}_{Test} = \mathcal{D}_n$

$\mathcal{D}_{train} \cap \mathcal{D}_{Test} = \phi$

$\forall x_i$

$\mathcal{D}_{train} \rightarrow n_1$

$n_1 + n_2 = n$

$\mathcal{D}_{Test} \rightarrow n_2$

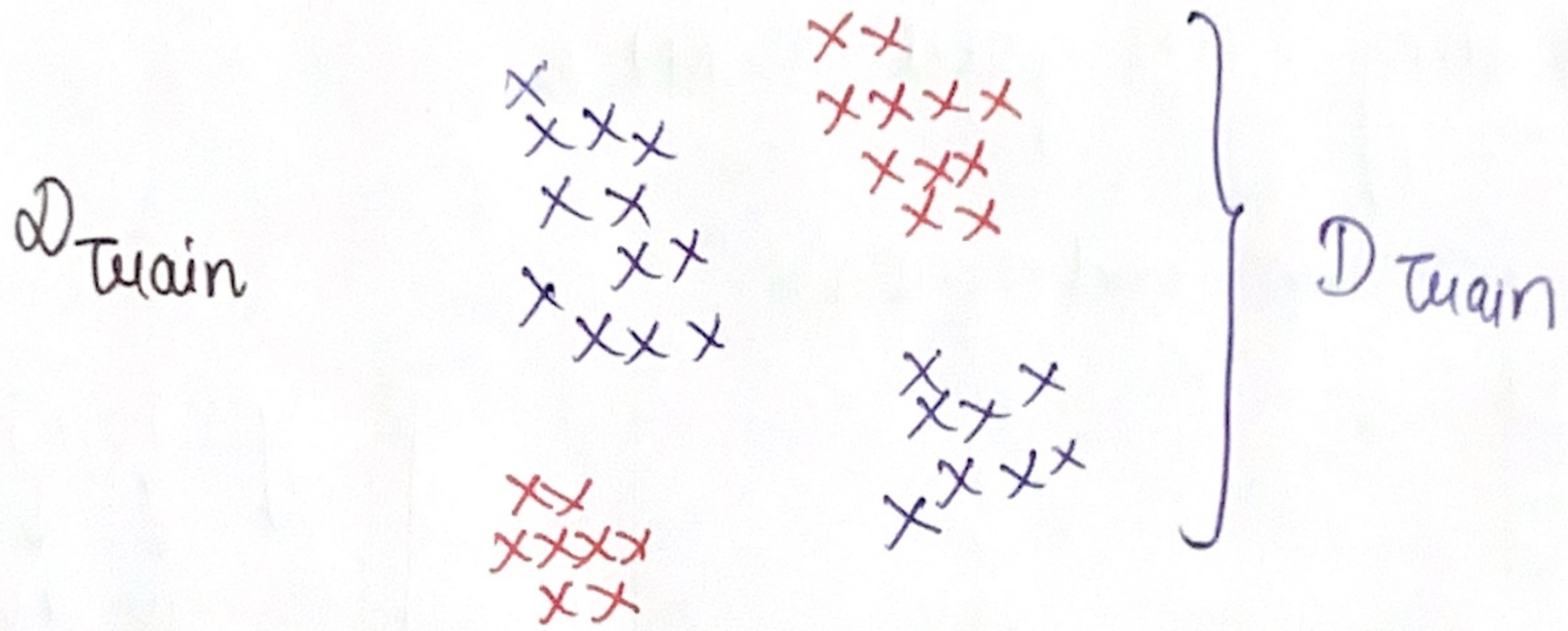
Split $\mathcal{D}_n$ → $\mathcal{D}_{train}$ (70%), $\mathcal{D}_{Test}$ (30%) → randomly

(h1) $\mathcal{D}_{train}$ → k-NN

$(x_i, y_i)_{i=1}^{n_1}$

(n2) $\mathcal{D}_{Test}$

$(x_i, y_i)_{i=1}^{n_2}$




for each point $x_i$ in $\mathcal{D}_{test} = (x_i, y_i)_{i=1}^{n_2}$

$x_q = (x_1)$ for each pt in $\mathcal{D}_{test}$

$\longrightarrow x_q = pt$

$\longrightarrow$ use $\mathcal{D}_{train}$ & KNN to predict $y_q$

if $y_q == y_{pt}$

cnt += 1

end

cnt = # pts for which $\mathbb{D}_{Train}$ + KNN gave a correct class label

$$\text{Accuracy} = \frac{cnt}{n_2}$$

$\longrightarrow$ # for which $\mathbb{D}_{train}$ + KNN gave a correct class label

$\swarrow$ # pts in $\mathcal{D}_{test}$

$0 \leq \text{Accuracy} \leq 1$

Acc = 0.91 $\Rightarrow$ 91% of times

$x_q \rightarrow y_q$

Conclude KNN on Amazon food review using D_train gives me an accuracy of 91% / assumption

## Test / Evaluation time & Space Complexity

$x_q \rightarrow y_q$

Input: $\mathcal{D}_{train}$, K, $x_q \in \mathbb{R}^d$; output: $y_q$

K is small 5 or 10

KNN pts = [ ]

→ npts; d-dim
↳ BOW (10K)

for each $x_i$ in $\mathcal{D}_{train}$:
O(d) — compute $d(x_i, x_q) \rightarrow d_i$
— kup the smallest K- distances $(x_i, y_i, d_i)$

cnt pos = 0  cnt-neg = 0

for each $x_i$ in KNN pts
if $y_i$ is +ve
cnt_past += 1
else
cnt_neg += 1

O(1) { if cnt pos > cnt-neg
return $y_q$ = 1 → +ve
else
$y_q$ = 0 → -ve.

Time Complexity = $O(nd) + \cancel{O(1)} + \cancel{O(1)}$

$O(nd)$

if d is small

if $d << n$

$\boxed{O(n)} \longrightarrow$ Time

## Amazon food review

$n \approx 364K$

$d \approx 100K$ (BOW)

$300$ (tfidf)

→ lot of memory

RAM / Memory

deploy

$x_q \rightarrow y_q$

Test / evaluation

Dmain → $O(nd)$

Space complexity (evaluation): space which is required to evaluate

$x_q \rightarrow y_q$

$O(nd)$

## Limitations of K-NN (Simple implementation)

(Amazon) fine food review: (real time system)

Time complexity $O(nd)$

Space complexity $O(nd)$

$n \approx 364K$
$d \approx 100K$
} a lot of space

$36400M \approx 36GB$

Time Complex: 36 Billion Computations

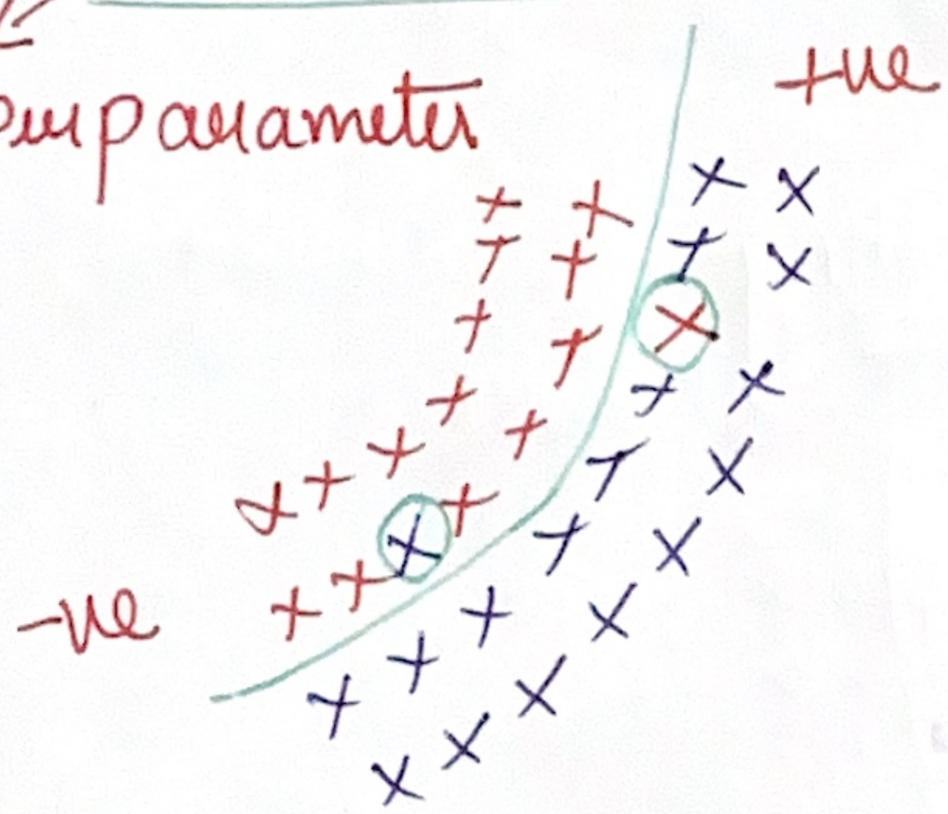
Internet Review $\xrightarrow{1ms}$ +ve/-ve

(low latency system) $x_q \xrightarrow{fast} y_q$

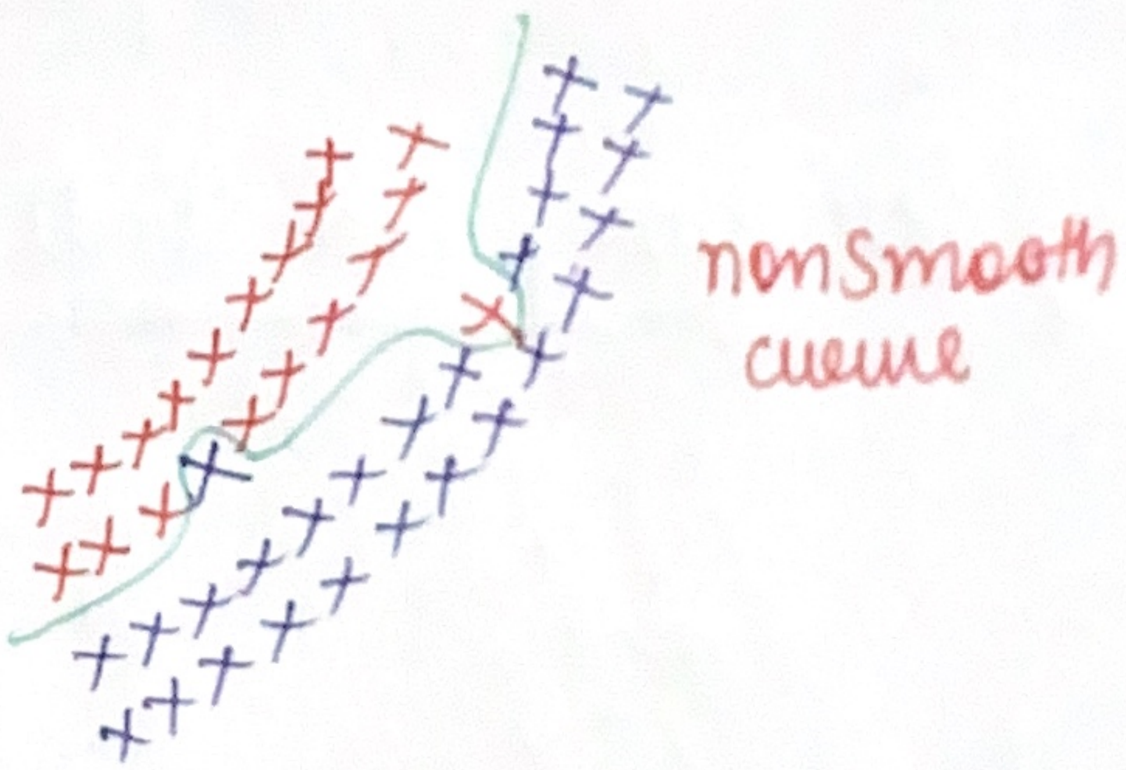
KNN Simple Implementation $\rightarrow$ O(nd) O(nd

'K' in K-NN

Hyperparameter



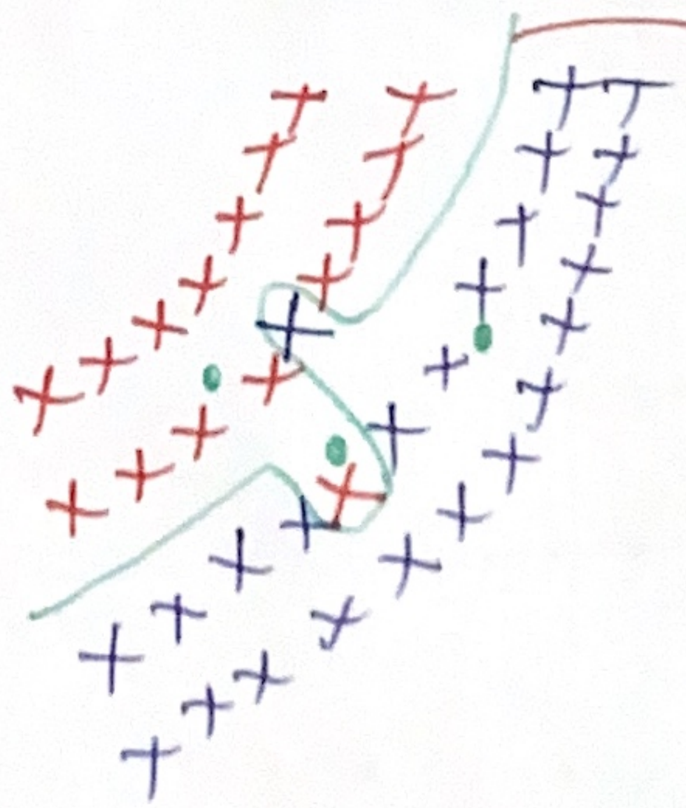

Smooth curve



nonSmooth curve

curves -ve from +ve

↳ decision surfaces



Decision Surface

↓

many query points

1-NN (k=1)

→ separate region using curve

Let K = 1

1 = NN

Let k = 5

Take majority vote

↳ Smooth curve { as k↑ smoothness of curve increases

let say $K=n$

$K = 1, 2 \text{ ---- } n$

max value of $K=n$

$n$ = Total no of points

$n_1$ = +ve

$n_2$ = -ve

$n_1 + n_2 = n = 1000$

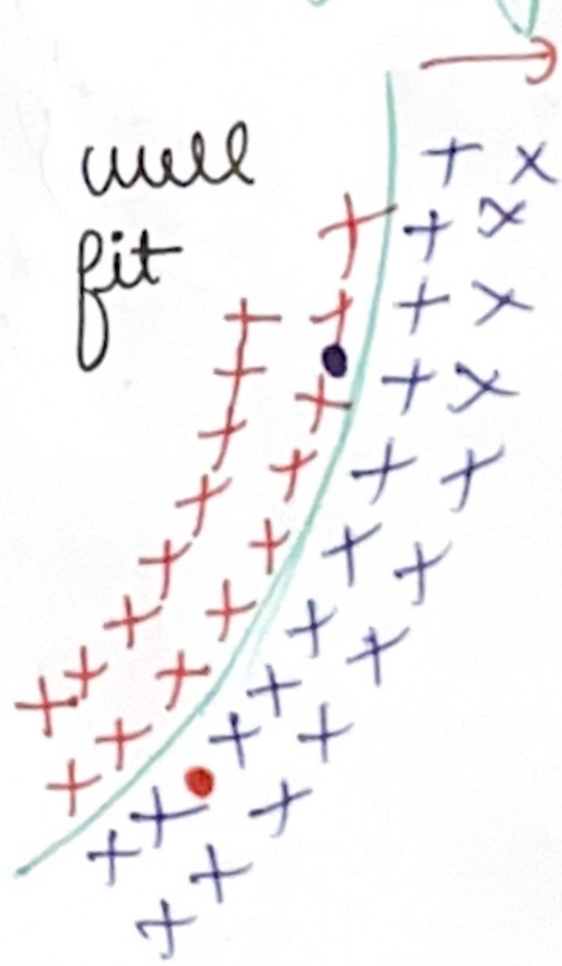
let's say $n_1 > n_2$

$n_1 = 600$

$n_2 = 400$

Everything will turn out +ve as majority points are positive...

## Overfitting and Underfitting (fundamental)


every point belongs to the majority class

$K=1$ ⟶ no-mistakes

$K=5$ ⟶ smooth

$K=n$

fitting a function to data is fitting