Previous Class – 22 June 2023

1) Dataset & Problem Statement (Blinkit)
2) Visualising data & Motivation for kNN
3) kNN algorithm & assumptions
4) Code: scratch implementation
5) Train and Test-time Complexity for kNN

Today's agenda

1) Review of code: scratch implementation ✓
2) Sklearn's KNN implementation ✓
3) Different distance metrics ✓
4) Weighted kNN overview ✓
5) Bias – Variance Trade off in kNN ✓
6) Impact of outliers ✓
7) How to handle categorical features ✓
8) Applications: Google Image Search ✓
9) KNN based imputation ✓
10) Overview of Employee Attrition Dataset
11) AMA Session (11:15 pm)

Different Distance Metrics

1) Euclidean → $\sqrt{\sum_{j=1}^{d} (x_{ij} - x_{qj})^2}$

2) Manhattan → $\left[\sum_{j=1}^{d} |x_{ij} - x_{qj}| \right]^1$

3) Minkowski →

4) Cosine → $\left[\sum_{j=1}^{d} |x_{ij} - x_{qj}|^p \right]^{1/p}$

$p = 1 \Rightarrow$ Manhattan

$p = 2 \Rightarrow$ Euclidean

$p = 1.5, 2.5, 4, 6, \ldots$

Cosine: $(x_{i1}, x_{i2}, \ldots, x_{id}) \rightarrow x_i$

$(x_{q1}, x_{q2}, \ldots, x_{qd}) \rightarrow x_q$

$\overline{x_i}$ = L2 norm of x_i = $\sqrt{x_{i1}^2 + x_{i2}^2 + \ldots x_{id}^2}$

$\hat{x}_i = \frac{x_i}{\overline{x_i}} = \left(\frac{x_{i1}}{\overline{x_i}}, \frac{x_{i2}}{\overline{x_i}}, \frac{x_{i3}}{\overline{x_i}}, \ldots \frac{x_{id}}{\overline{x_i}} \right)$

↓

unit vector

$\overline{x_q}$ = L2 norm of x_q = $\sqrt{x_{q1}^2 + x_{q2}^2 + \ldots x_{qd}^2}$

$\hat{x}_q = \frac{x_q}{\overline{x_q}} = \left(\frac{x_{q1}}{\overline{x_q}}, \frac{x_{q2}}{\overline{x_q}}, \ldots, \frac{x_{qd}}{\overline{x_q}} \right)$

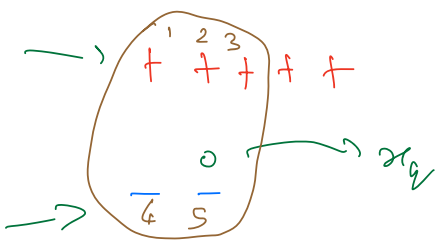
$$\text{cosine}(\hat{x}_q, \hat{x}_i) = \hat{x}_{i1} \cdot \hat{x}_{q1} + \hat{x}_{i2} \cdot \hat{x}_{q2} + \cdots + \hat{x}_{id} \cdot \hat{x}_{qd}$$

↳ 0 to 1

As a general practice:

1) Lower dimensions → Euclidean Distance
2) Higher Dimensions → Cosine Similarity

Weighted KNN



x_i:

$$w_i = \frac{1}{d_i}$$

$K = 5$

↓

- ve

$d_{1q} = 0.5 = d_{2q} = d_{3q}$

$d_{4q} = 0.1 = d_{5q}$

$w_{1q} = \frac{1}{0.5} = 2 = w_{2q} = w_{3q}$

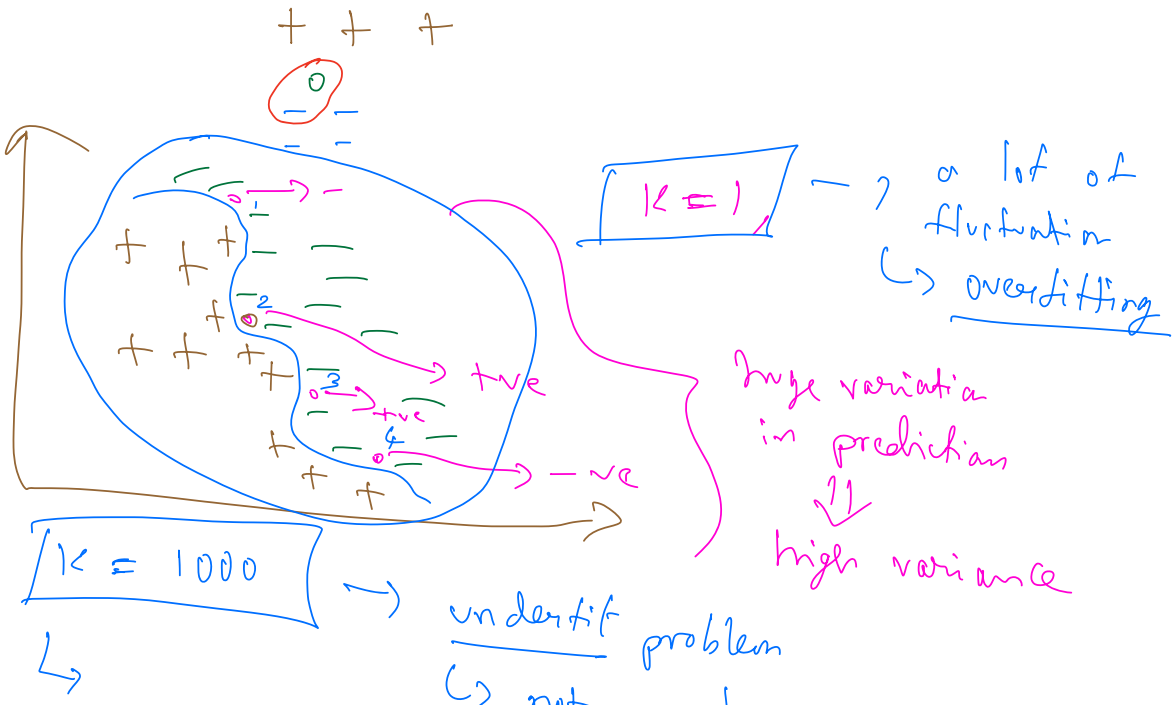
$w_{4q} = \frac{1}{0.1} = 10 = w_{5q}$

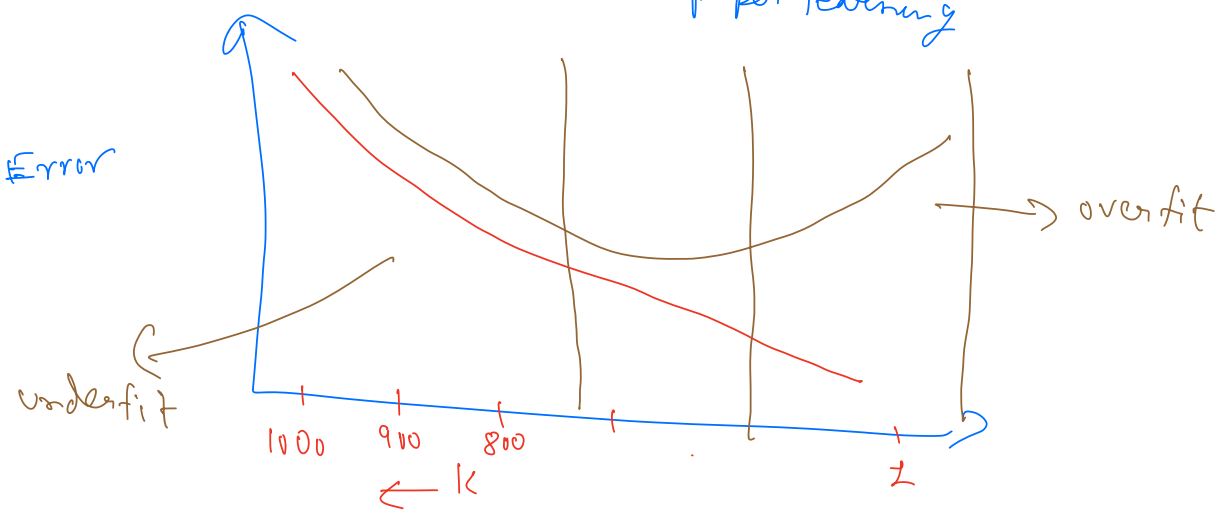
+ve → 2 + 2 + 2 = 6

-ve → 10 + 10 = 20

20 > 6 ⇒ pred → -ve

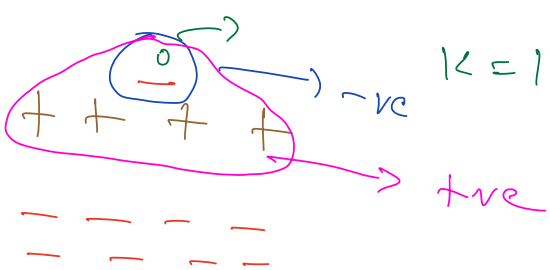
Extend it to more than 2 classes





K = 5

K is small, outliers can impact predictions



small K values tries to fit the noise ⇒ one definition of overfitting

Break till 10:20 pm

How do we handle categorical features

1) One hot encoding →
2) Label encoding →
3) Target encoding →

OHE → 3 classes → 2 D vectors

Dog → 01
Cat → 10
Horse → 00

N classes → N-1 dimensions

Cat	Dog	
0	1	→ Dog
1	0	→ Cat
[Horse] 0	0	→ neither Cat nor Dog

Label encoding

Dog → 1
Cat → 2
Horse → 3

Target encoding

C_1	
Dog →	p_1
Cat →	p_2
Horse →	p_3

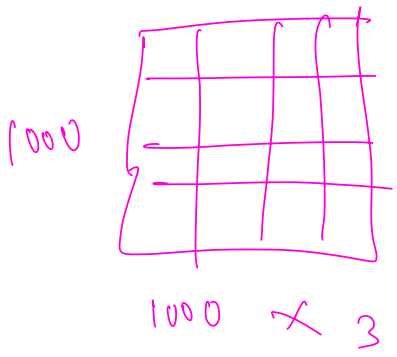
y label → 0/1

$p(y=1 / C_1 = \text{Dog})$
↳ p_1

$p(y=1 / C_1 = \text{Cat}) = p_2$

$p(y=1 / C_1 = \text{Horse}) = p_3$

Categorical Embeddings



→ CNN → 512 dimensional vector

I_1 → Z M or Z B Images → top k

I_2 → top k

FAISS → Facebook

SCANN → Google

Youtube? Similar Content Search

Imputation

	f_1	f_2	f_3	y
x_1	10	17	18	0
→ x_2	11	=	4	0
x_3	15	1	2	1
x_4	9	23	3	1

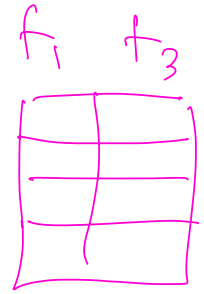
x_{22} ?

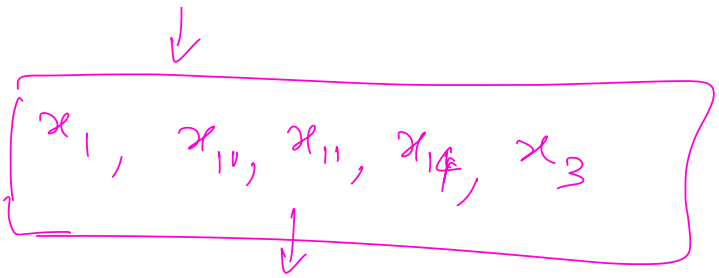
↓

Mean, Median, Same Statistic value

Drop → f_2

top k closest points to x_2 →

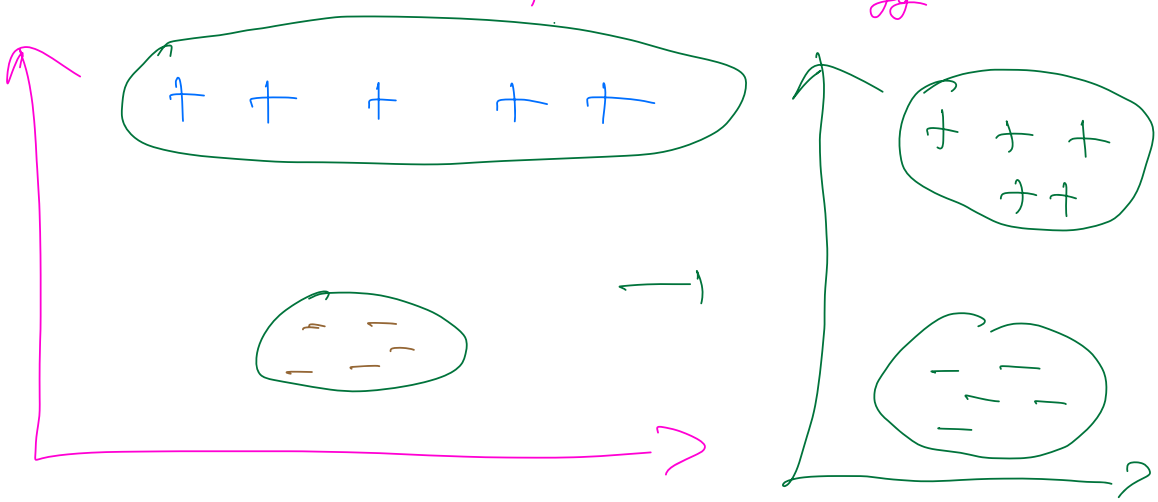


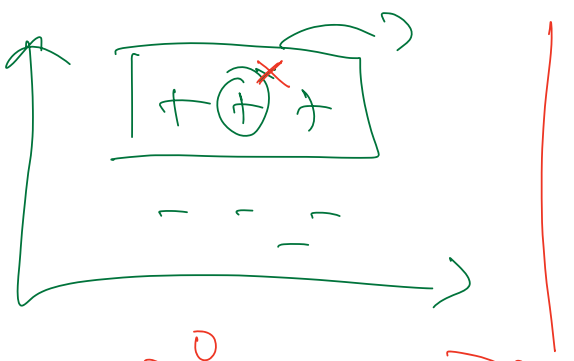


$x_{2,2}$ = avg of ($x_{1,2}, x_{10,2}, x_{11,2}, x_{14,2}, x_{3,2}$)

↓ continuous data.

For other data, other strategy





$k = 1$ ~~does it train~~

↳ impact on train accuracy

$$-\left[y_i \log(p_i) + (1-y_i) \log(1-p_i) \right]$$

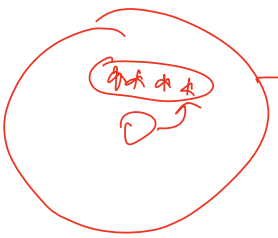
y_i : 0 / 1

p_i = prob of falling into class 1

undefined

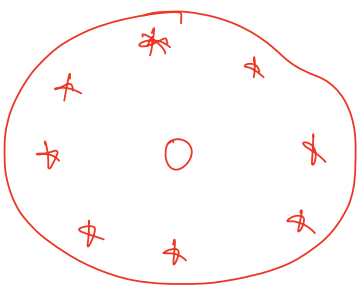
→ low variance

high bias

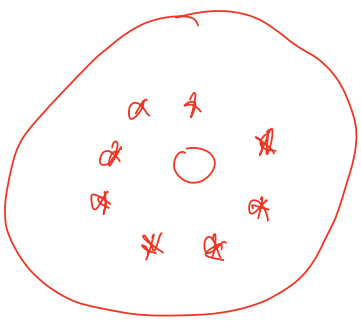


→ low variance

low bias



High Bias



Low Bias