

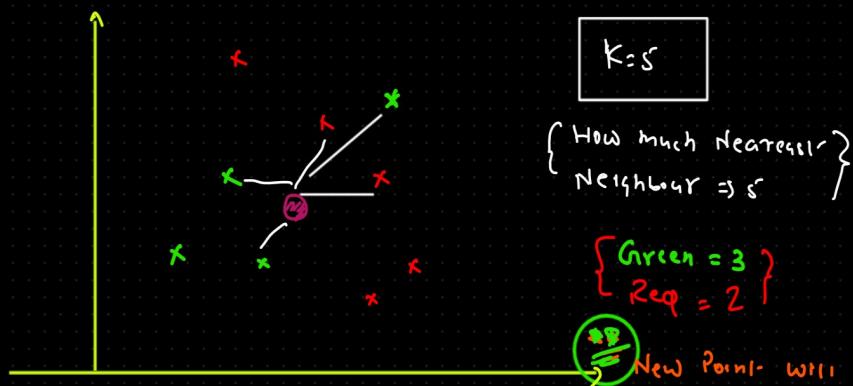
KNN implement-

- 1 Weighted KNN
- 2 KD tree
- 3 Ball tree
- 4 $K \leq 1, K = N$ (Odd, even)
- 5 Prox Cons

Unsupervised learning

- 1 K-means | k-means++
- 2 hierarchical
- 3 DBSCAN

Weighted KNN



$$\begin{aligned} \text{Basic KNN} &\rightarrow \text{Uniform weight} \\ &= \text{Data Point} \Rightarrow \{ \text{uniform weight} \} \\ &= \text{Distance} \Rightarrow \text{Euclidean Distance} \end{aligned}$$

2 Weighted KNN

Assume

Point	Label	Distance	Weight (1/Distance)
(x_1, y_1)	Red	0.2	$\frac{1}{0.2} = 5$
(x_2, y_2)	Red	0.5	$\frac{1}{0.5} = 2$
(x_3, y_3)	Green	0.7	$\frac{1}{0.7} = 1.4$
(x_4, y_4)	Green	1.2	$\frac{1}{1.2} = 0.8$
(x_5, y_5)	Green	1.5	$\frac{1}{1.5} = 0.6$

$$\text{Green} = 1.9 + 0.8 + 0.6 = 2.3$$

$$\text{Req} = \frac{5+2}{2} = 7$$

Simple vs Weighted

\Rightarrow Conclusion = RED

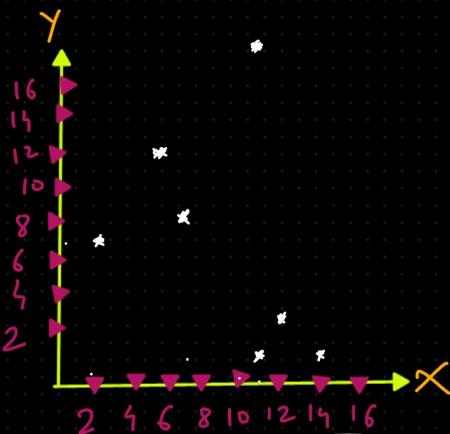
Brute force

{ We navigate to every point }
 { Data Point }

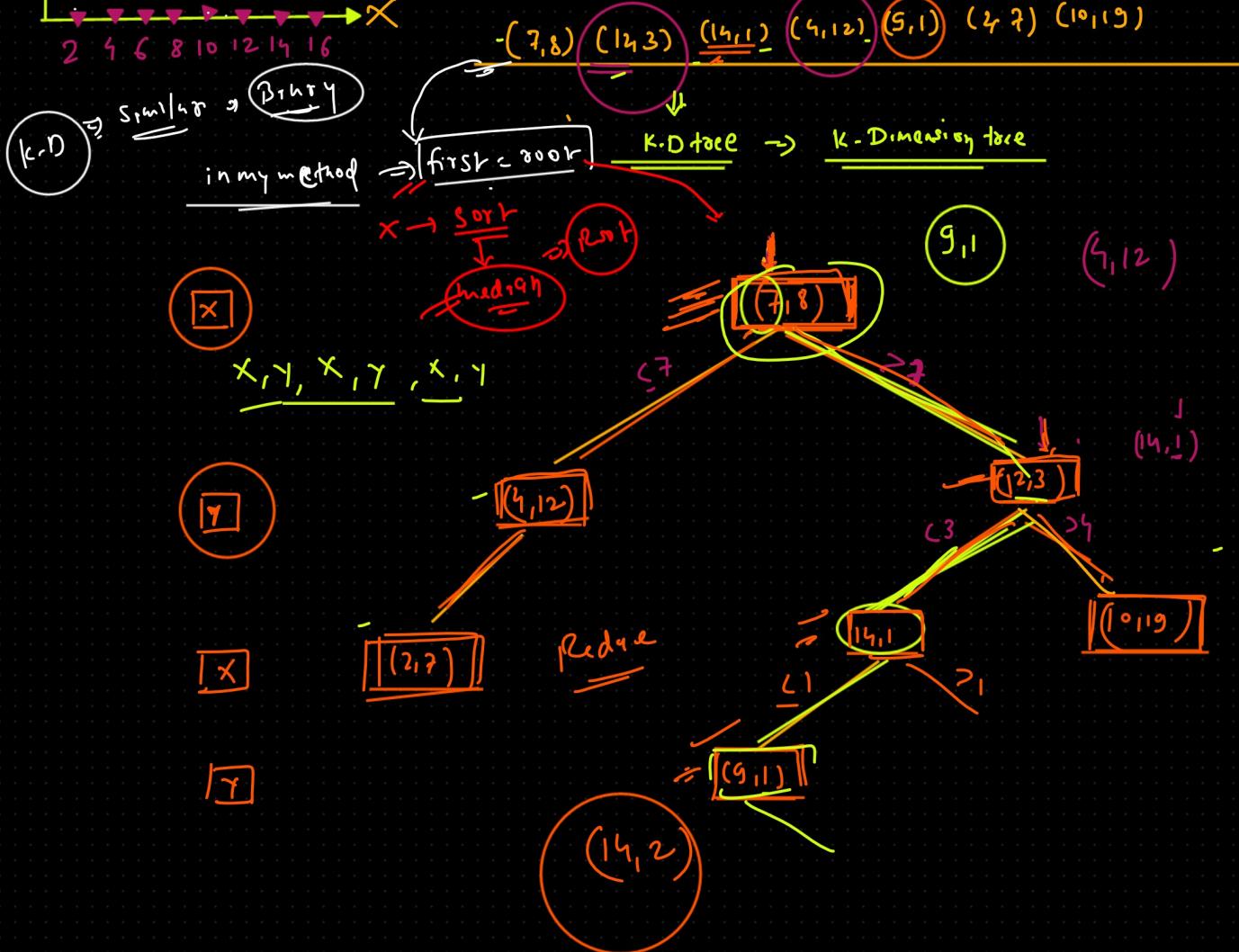
Index Based

{ 1 K-D tree } { 2 Ball tree } (We define our data in such a way so we can minimize the effort)
 { Arrangement of the data } { No need to traverse each and every point }
 (Time Complexity) (Space Complexity)

K-D tree :-

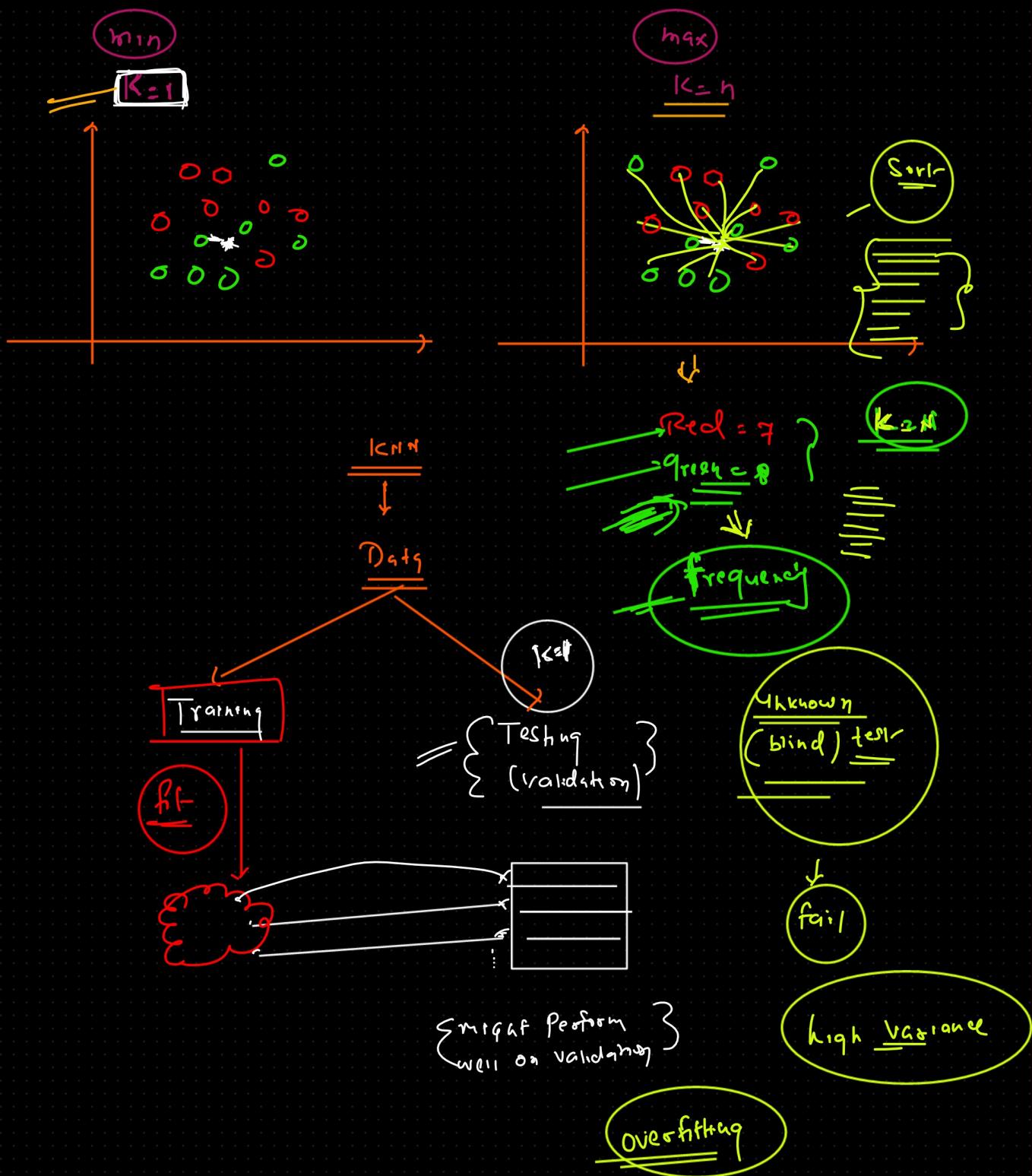


X	Y
7	8
12	3
14	1
9	12
9	1
2	7
10	19



Ball tree

One more example of K-D tree



CONS

1

Large data and higher dimension

$(m \times n)$

ICR 1000

huge

kNN

Distance

Folath

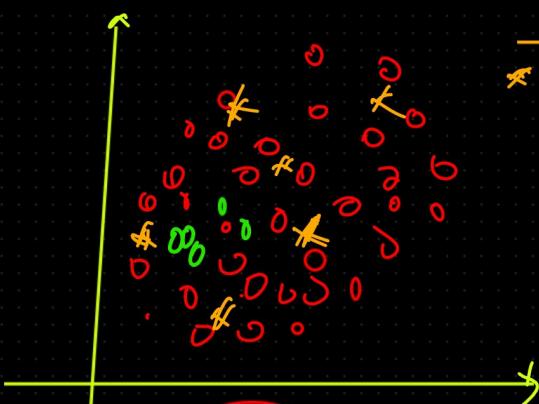
Distance

Solution

2

~~kNN~~

unbalanced data set

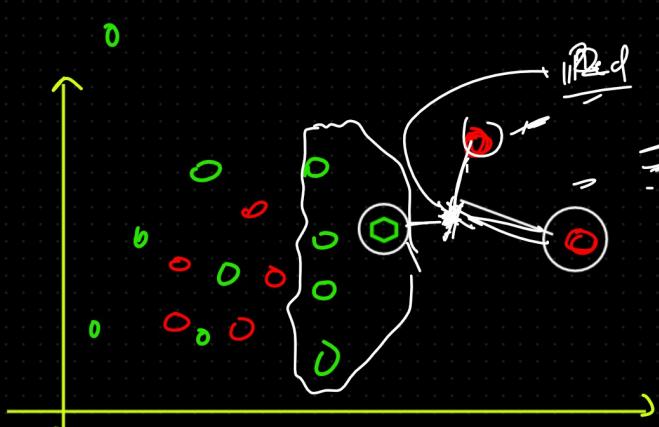


Red

100

Classify as 100

Wrong Result



Outlier

$(1-10)$	$\underline{\text{GPA}}$	$\underline{\text{Salary}} \quad [0-\infty]$
	5.8	100000
	6.7	30000
g		500000

Scale is $0.1R^2$

Scale

$$(100000) \Rightarrow 100001$$

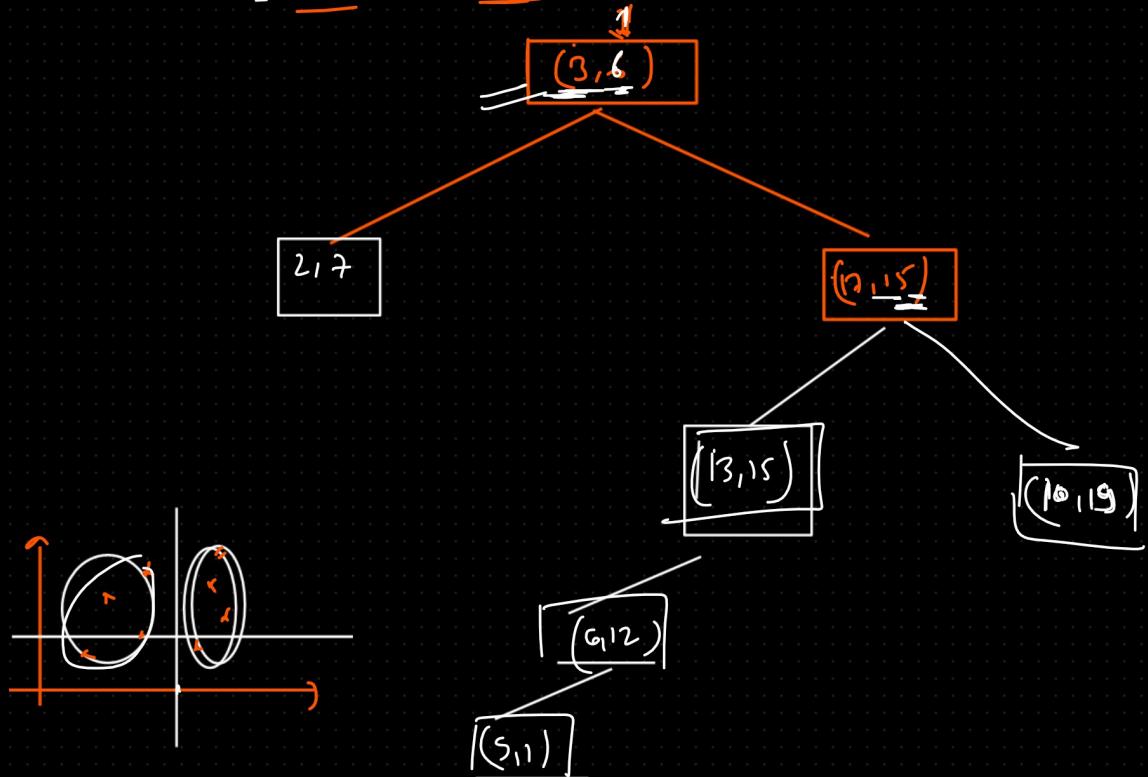


{ Higher Dim and large Instance }

Basic CNN, k-D tree

K-D tree

$(3, 6)$ $(17, 15)$ $(13, 15)$ $(6, 12)$ $(5, 1)$ $(2, 7)$ $(10, 19)$



$(3, 6)$ $(17, 15)$ $(13, 15)$ $(6, 12)$ $(5, 1)$ $(2, 7)$ $(10, 19)$

$\rightarrow x = \underline{3, 17, 13, 6, 9, 2, 10}$

$2, 3, 6, \boxed{9}, 10, 13, 17$

