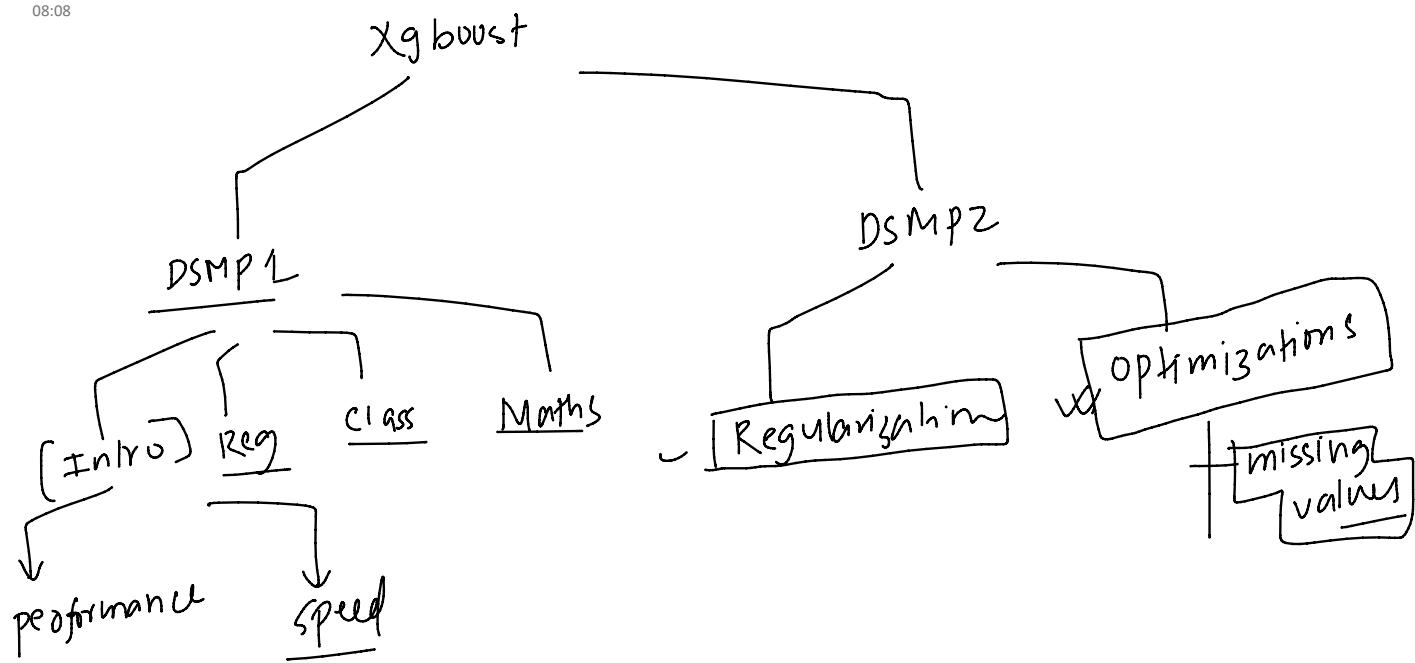


Recap

09 April 2024 08:08

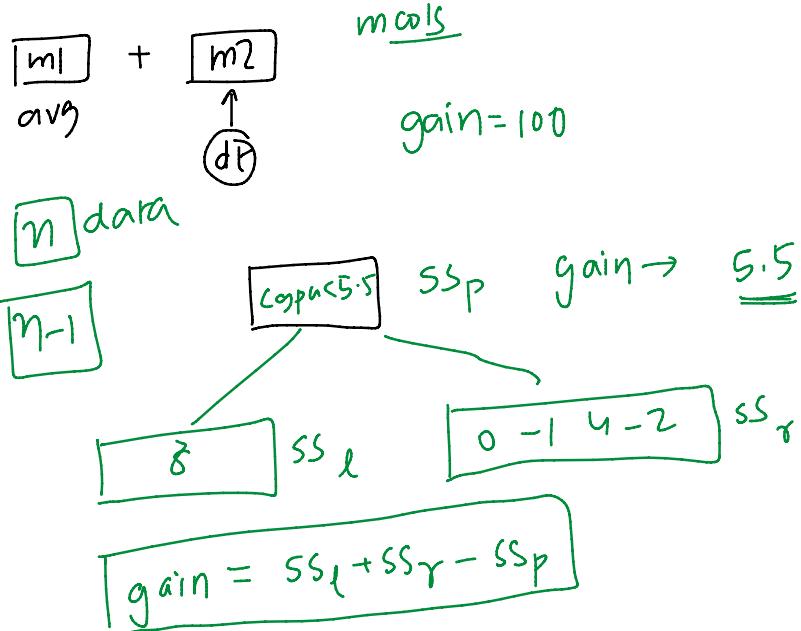
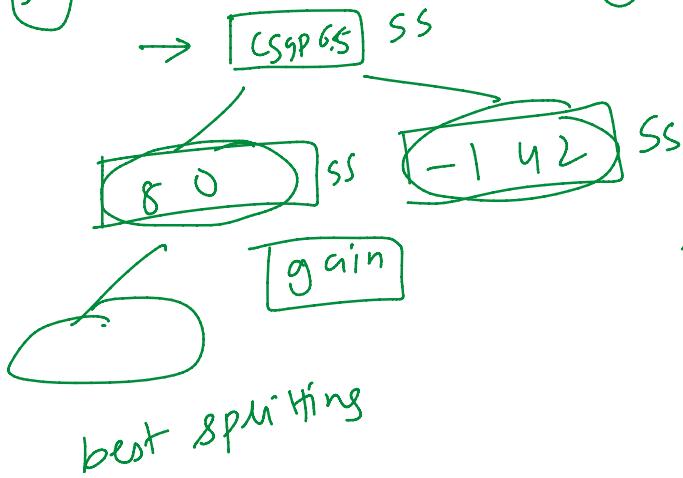


Exact Greedy Split Finding

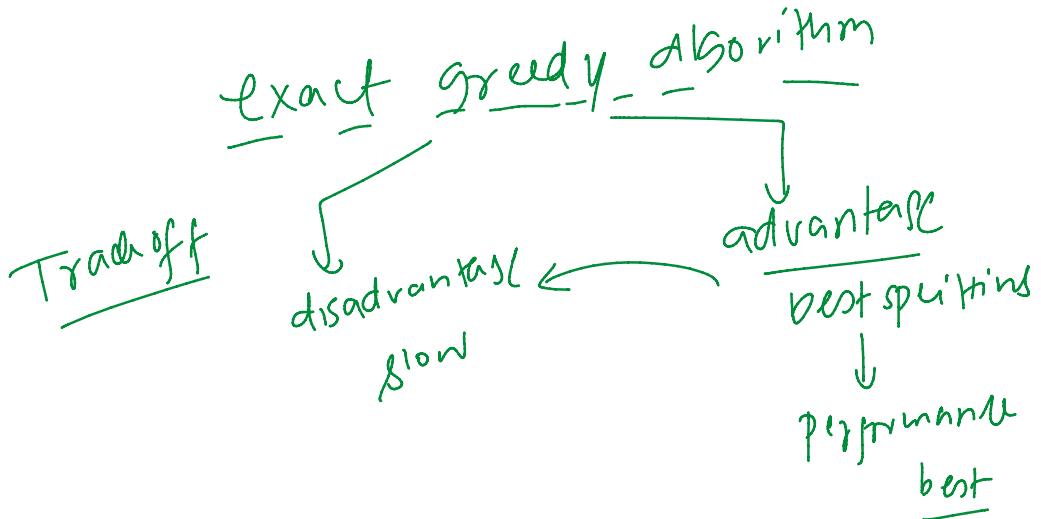
09 April 2024 15:52

CGP1	paufage	pred	xgboost	res1
5	5	13	8	
6	13	13	0	
7	14	13	-1	
8	9	13	4	
9	15	13	-2	

$f_1 \rightarrow \text{CGPm}(5)$ 200 6



100000 rows 100 cols
 $100(\text{aaaa})$



Approximate Method For Split Finding

09 April 2024 08:08

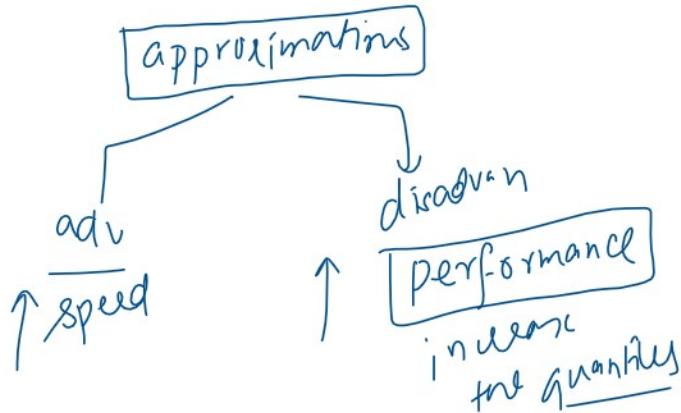
cgpa	package
4	5
5	13
6	14
7	9
8	15
9	4

(approx)
3 quantiles (6)

exact
greedy
(5)

approx
(2)

algo sped up



quantiles
① → approximate → basic idea

Approx

g → gradient
= 3 h → hessian

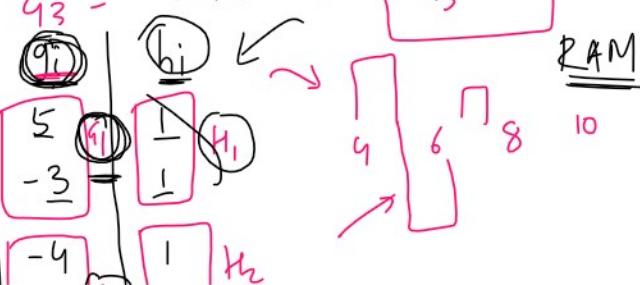
$$\text{reg} \rightarrow \hat{g}_i = \hat{y}_i - y_i$$

G_1, H_1 | L data points g_i

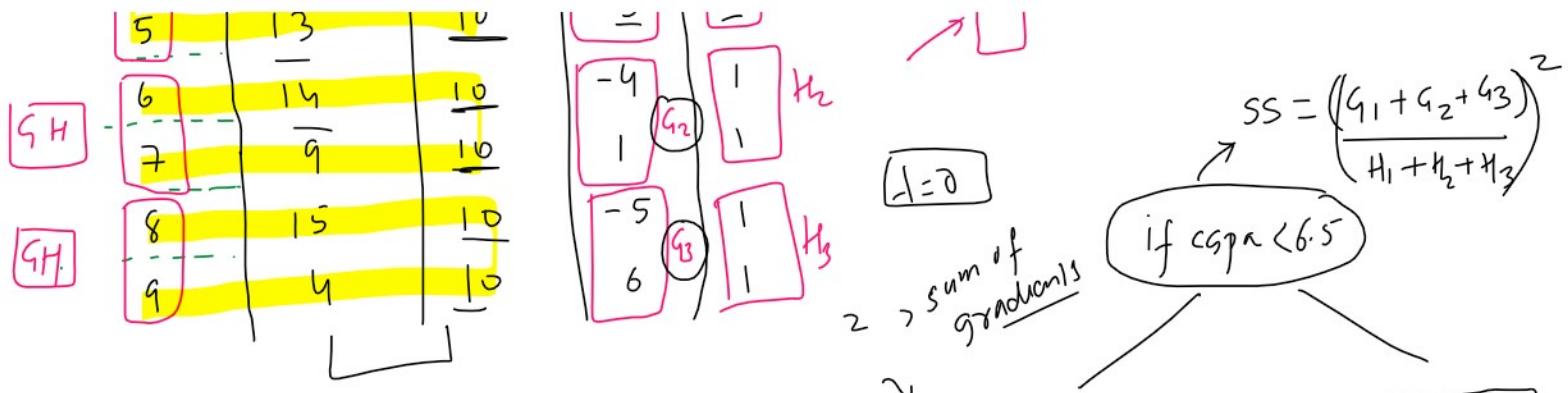
cgpa	package	pred
4	5	10
5	13	10

$$\begin{aligned} g_1 &= 5 - 3 = 2 \\ g_2 &= -4 + 1 = -3 \\ g_3 &= -5 + 6 = 1 \end{aligned}$$

$$\begin{aligned} H_1 &= 2 \\ H_2 &= 2 \\ H_3 &= 2 \end{aligned}$$



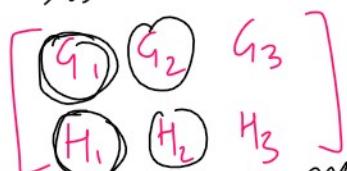
$$or - \|G_1 + G_2 + G_3\|^2$$



Step 1 → No. of quantum bins
buckets

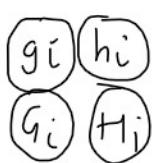
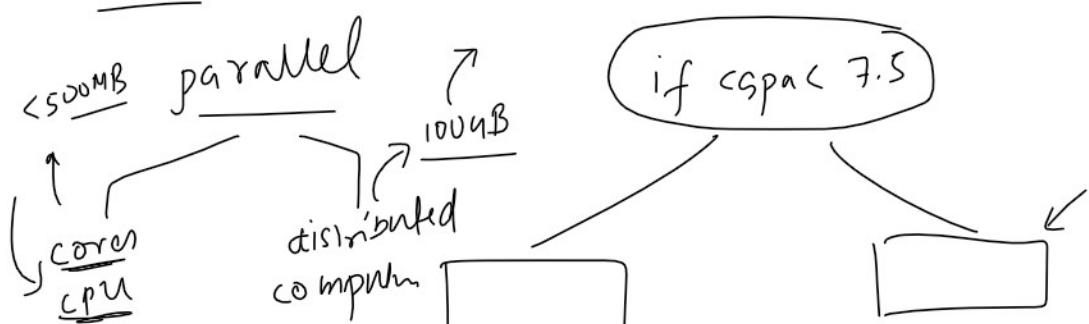
Step 2 → We calculate g and h for every data point
 $(\hat{y}_i) = \underline{1} \circ h$

Step 3 → histogram grad / hessian
based on the quantity



Step 3a → store in cache memory

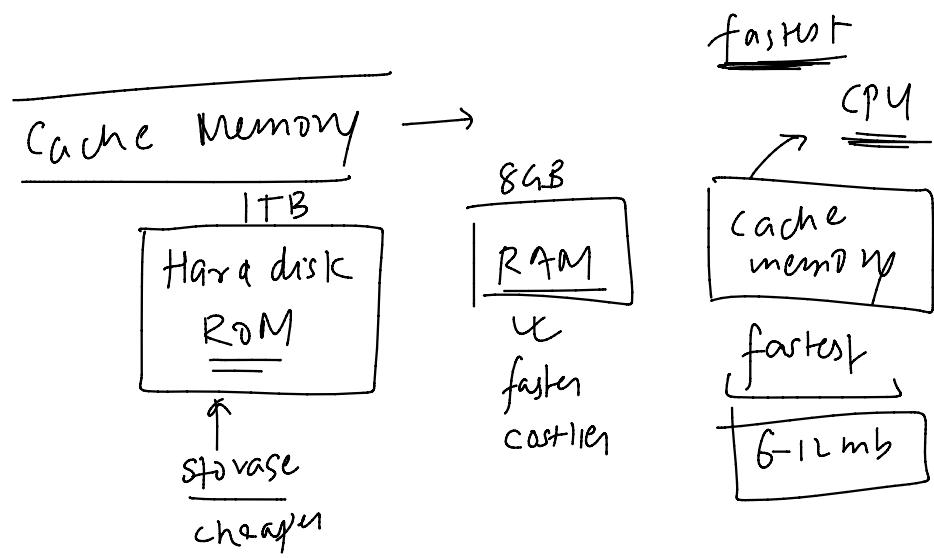
Step 4 → Try out splitting criteriant



$$SS = \frac{(G_1 + G_2)^2}{H_1 + H_2}$$

$$SS = \frac{(G_3)^2}{H_3}$$

[fart]



- 1) reduced # splits \rightarrow quantity \rightarrow histov
- 2) calculate $g_1 \dots g_n h_1 \dots h_n$
- 3) p_a
- 4) Cache

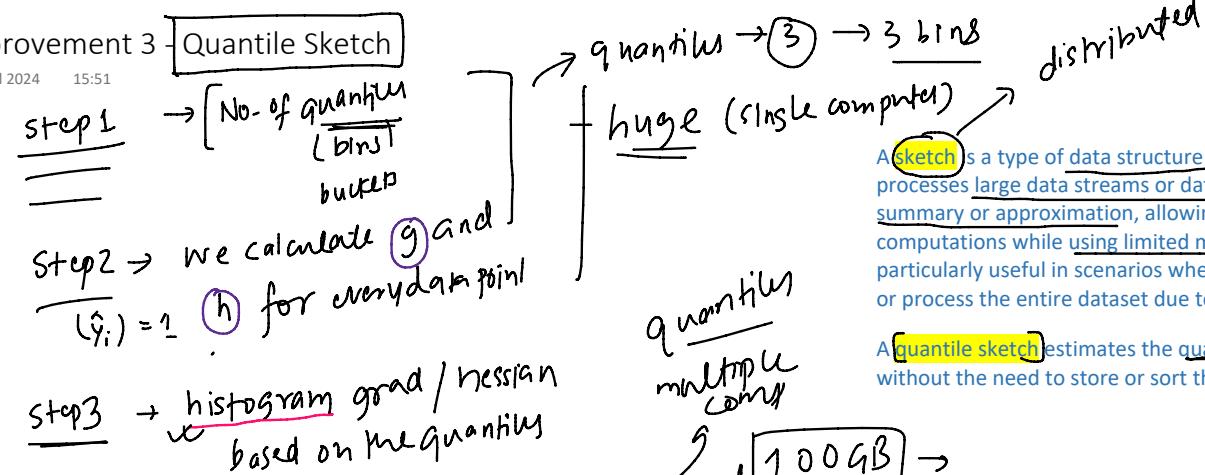
X G Boost
 \downarrow
Xtreme

Improvement 1 - Parallel Processing

09 April 2024 08:08

Improvement 2 - Cache Storage

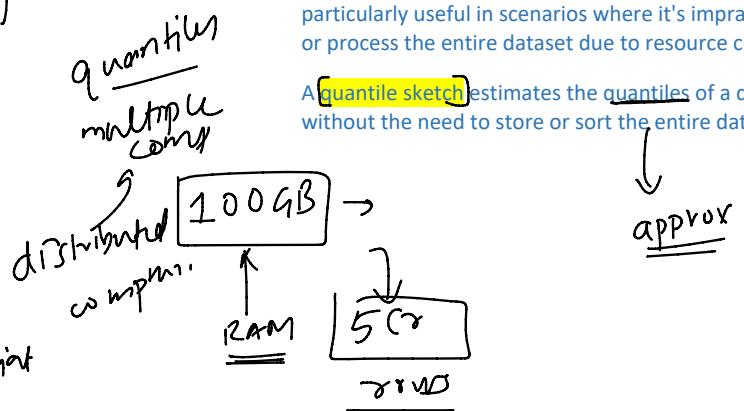
09 April 2024 15:51



quantiles → 3 → 3 bins
huge (single computer) → distributed

A sketch is a type of data structure or algorithm that processes large data streams or datasets to create a summary or approximation, allowing for efficient computations while using limited memory. The concept is particularly useful in scenarios where it's impractical to store or process the entire dataset due to resource constraints.

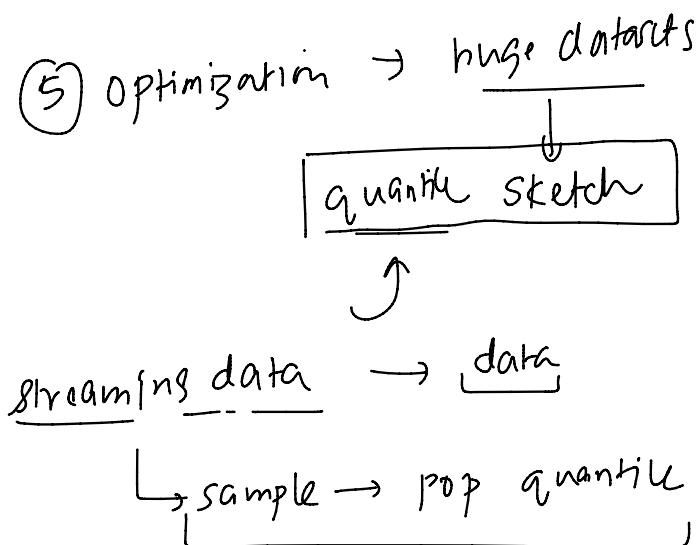
A quantile sketch estimates the quantiles of a dataset without the need to store or sort the entire dataset.



- Efficiency: They enable efficient computation of approximate quantiles without sorting the entire dataset, reducing time and computational resources.
- Scalability: Quantile sketches help XGBoost efficiently handle large datasets by summarizing data distributions compactly, which is crucial for processing big data.
- Memory Usage: They reduce memory requirements by providing a summarized representation of the data, avoiding the need to store all individual data points.
- Split Selection: Quantile sketches allow for effective and approximate identification of potential split points in the data, facilitating the decision-making process in tree construction.
- Distributed Processing: In distributed environments, quantile sketches can be combined across different data subsets, enabling XGBoost to effectively approximate split points even when data is scattered across multiple machines.

5.8 ✓

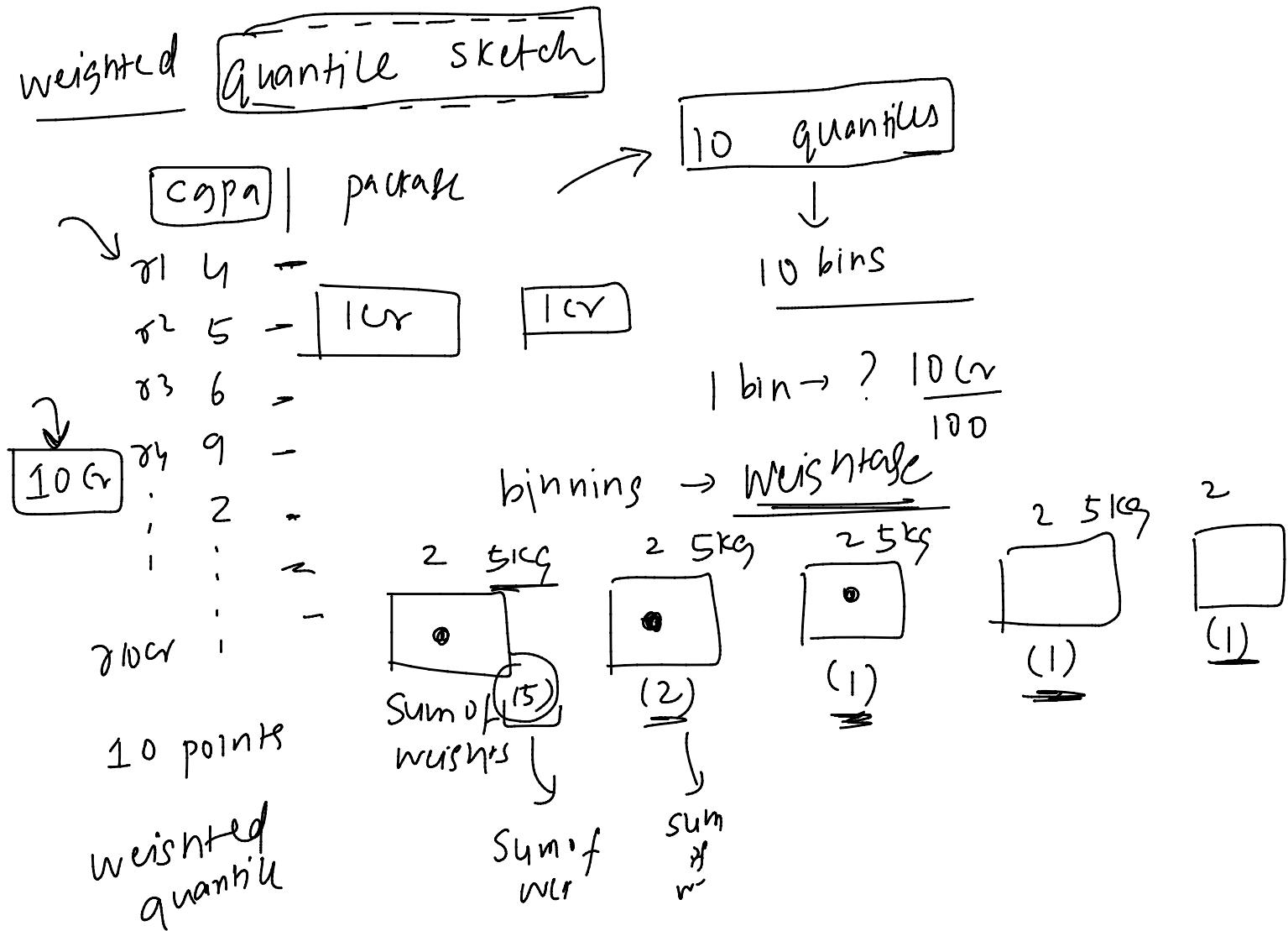
5.5



Improvement 4 - Weighted Quantile Sketch

09 April 2024 15:51

- Purpose: Like a standard quantile sketch, a weighted quantile sketch provides a way to estimate quantiles. However, it accounts for weights associated with each data point, offering a more nuanced summary of the dataset.
- Usage: This is particularly useful in datasets where some observations are more important or occur more frequently than others, and this significance is represented by weights.



normal quantile	vs	weighted quantile	w
classif	<u>weight</u>	$0.7(1-0.7)$	$0.7(1-0.7)$
cgpa placement	pred	$0.1(1-0.1)$	$0.5(1-0.5)$
3	grad	$0.2(1-0.2)$	$0.2(1-0.2)$
!	-0.3	0.11	weight
	0.21		

3
4
6
7
4
8

1
0
0
1
0

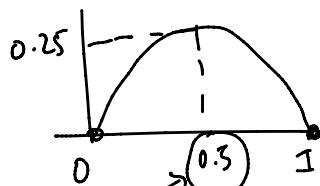
0.7
0.2
0.5
0.8
0.3

-0.3
0.2
0.1
-0.1
0.3

(0.2)
0.16
0.09
0.16
0.21

weight
width

min child weight → lower → sum of Hessians
↓



$$h_i = p_i(1-p_i)$$

Unsure → hessian → weightage

weighted quantile / sum of Hessians

weights quantile → 3

hi	x _i
1	3
2	4
1	6
2	7
1	4
2	8

hi

0.21
0.16
0.09
0.16

→ 1 pt → w

3 points → w

0.21
0.25

2 points + w

weights quantile → 3



hessians



weights of data points

Code

09 April 2024 15:52