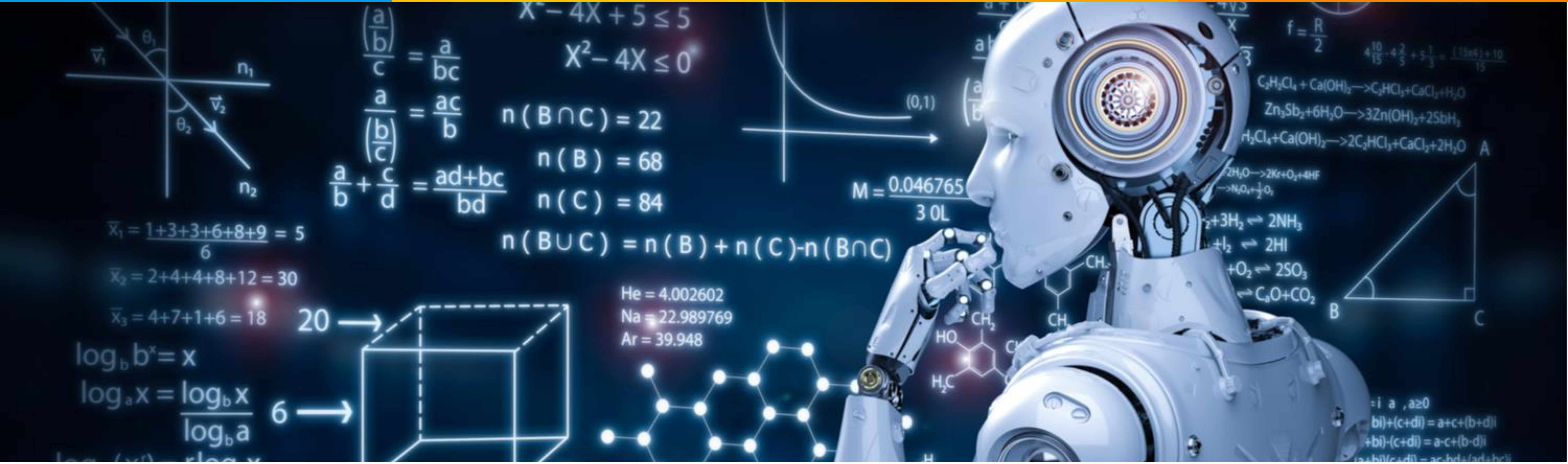
Duration: 60 min

Srikanth Shenoy

AML 5102

M.E. AI & ML



Lecture 12: Clustering 101

Recap

- Multivariate Gaussian distributions
  - Mahalanobis distance
  - Minimum Covariance Determinant
- Using Bayes Rule for generative ML
- Different kind of decision boundaries & corresponding covariance matrix

AML class logistics for next week

- ALA portion for sessional 1 completed
- Using ALA next week hours for AML
  - Tuesday – 11-12 as usual
  - Thursday - 9-10, 11-12

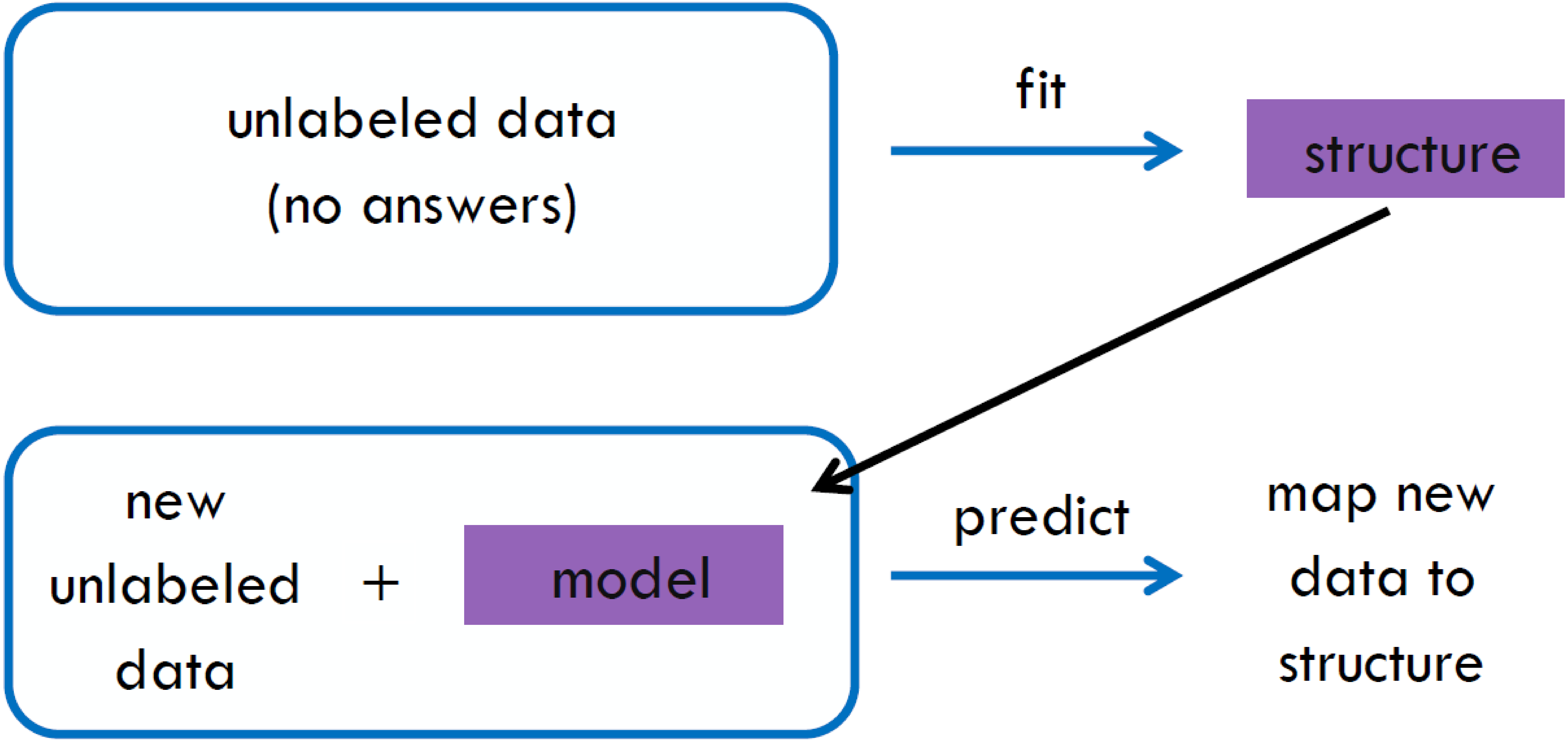
Sessional 1 – Theory

- 20 marks:10+ questions objective type: +2/-1
- 30 marks: No negative marks
  - Problems
  - 1-2 sentences answer of type “why/justify”
  - Given a formula, why is it the best choice?
  - Complexity of algorithm
- Disclaimer: There may be some variations

Sessional 1 – Lab

- 30 minutes: No API (from scratch) coding on paper
  - 1 of 3 algorithms (definitely a variation)
    - NearestCentroid, KNN, Kmeans – with CV
  - Using class, methods
  - Submit at 30 minutes
- 60 minutes: Coding on computer
  - Dataset will be given
  - Sklearn pipelining for EDA and fit/predict
  - Can use sklearn if your code does not work

Clusters, properties & basic metrics



unlabeled data

(no answers)

fit

structure

new

unlabeled

data

+

model

predict

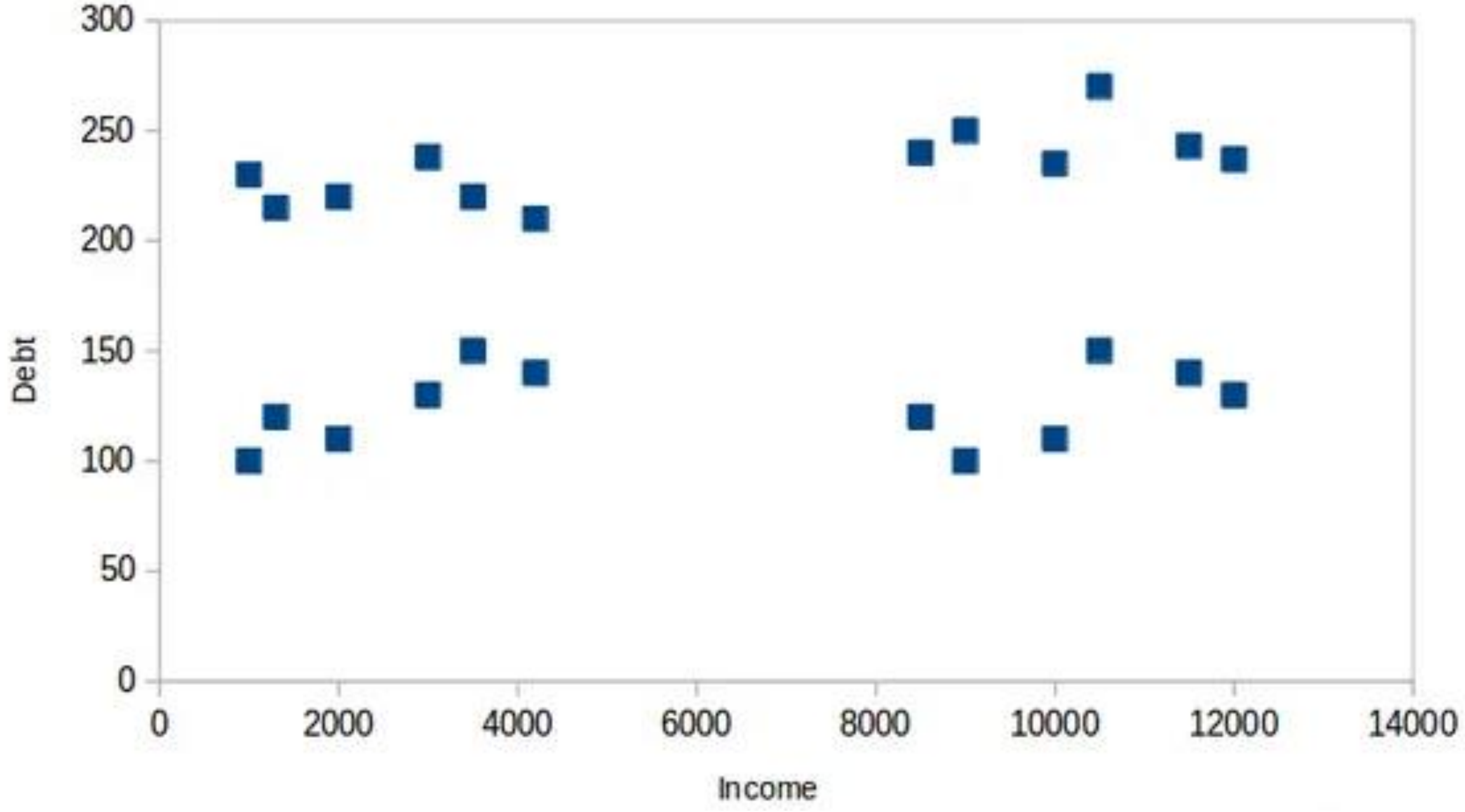
map new

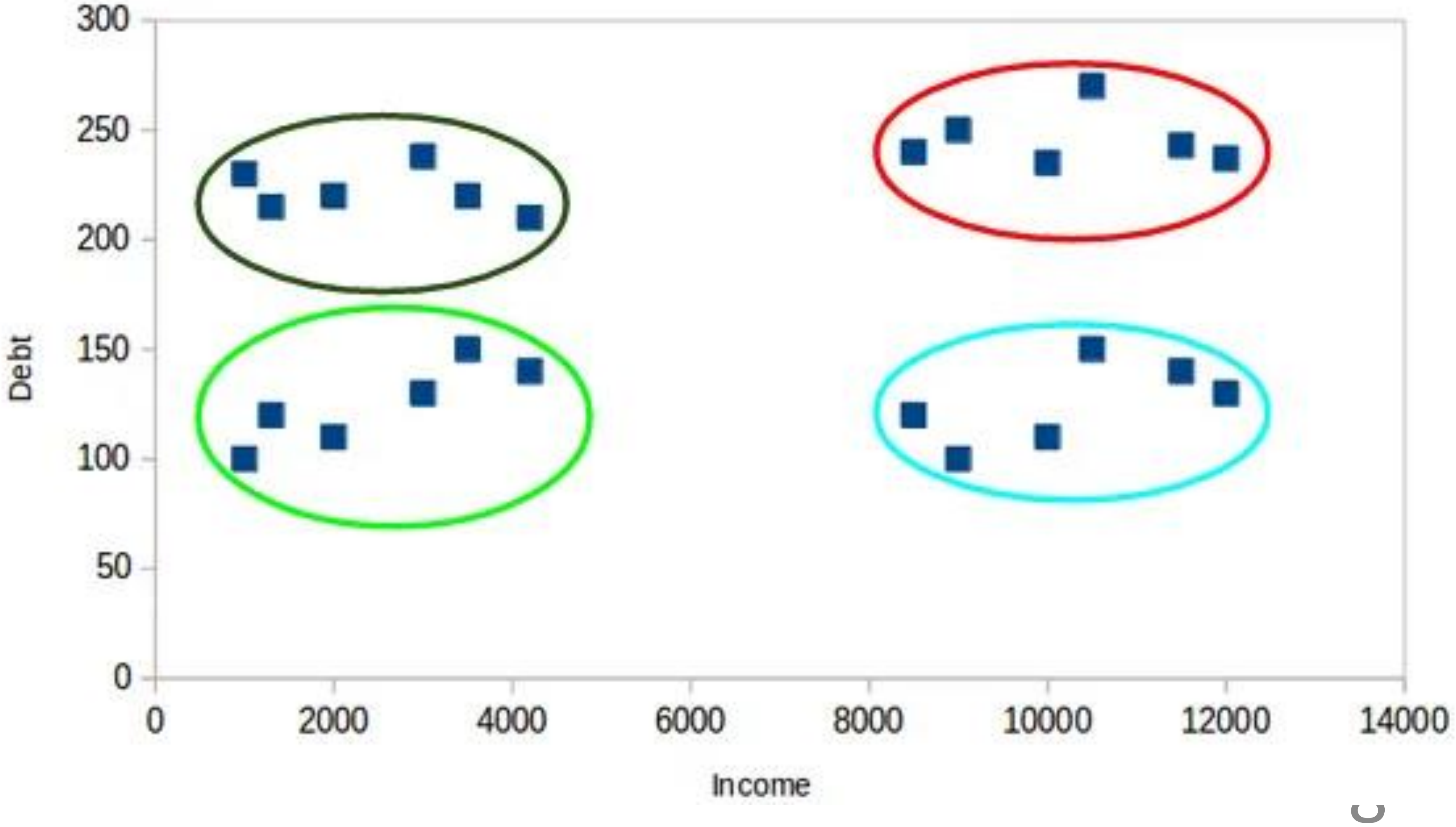
data to

structure

Clustering Intro

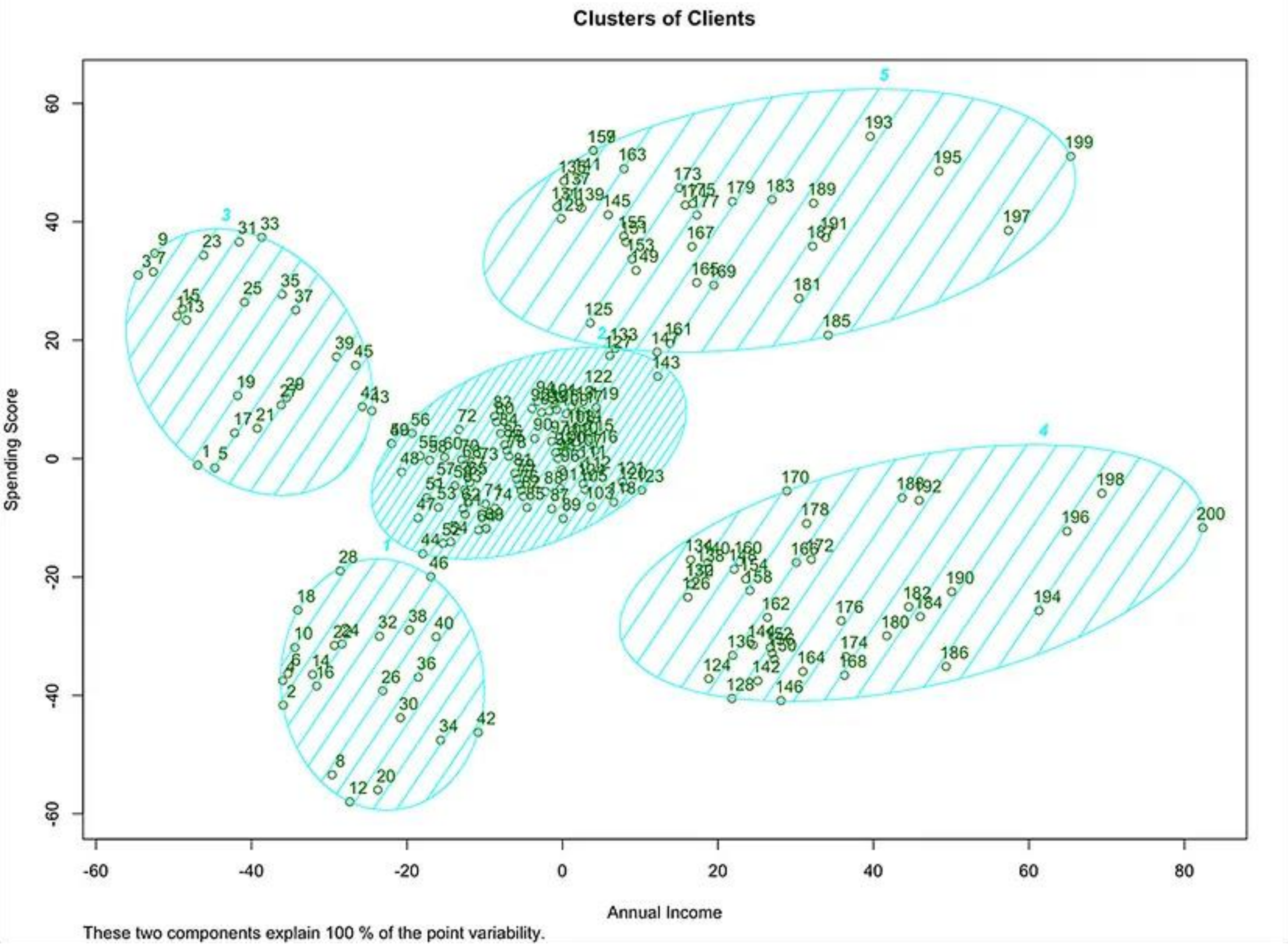
- Unsupervised Learning
  - No labels. Only data
- Dividing data into groups based on some underlying pattern

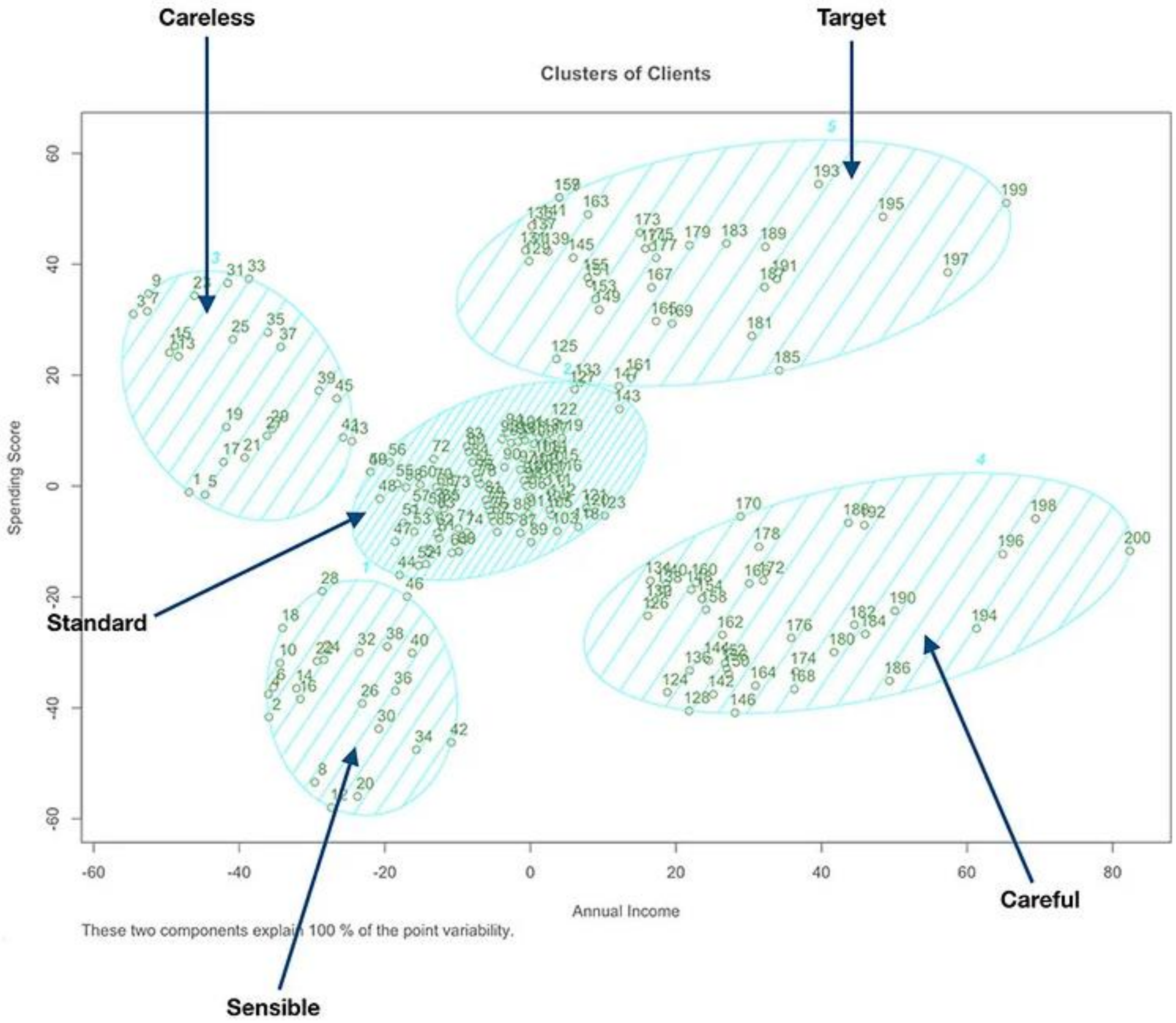




Clustering used in

- Customer segmentation
  - Targeted marketing & advertising
- Document clustering
- A computationally easy way for image segmentation
- Recommender systems


Clusters of Clients
Spending Score
Annual Income
60
40
20
0
-20
-40
-60
-60
-40
-20
0
20
40
60
80
1
2
3
4
5
9
31
33
23
3
7
15
11
13
25
35
37
39
45
19
29
27
17
21
41
43
1
5
159
163
135
141
137
131
129
139
145
173
171
175
177
179
183
189
193
195
199
155
151
153
149
167
187
191
165
169
197
181
125
133
127
147
161
143
185
122
119
94
72
56
49
50
48
55
58
60
70
73
57
51
53
47
52
54
90
83
88
87
89
103
118
121
123
77
74
170
188
192
198
178
196
200
134
140
160
138
148
154
166
172
130
158
126
190
182
184
194
162
176
180
144
152
136
156
150
174
124
142
164
168
186
128
146
28
46
44
18
10
22
24
32
38
40
6
4
14
16
2
26
36
30
34
42
8
20
12
These two components explain 100 % of the point variability.



Careless

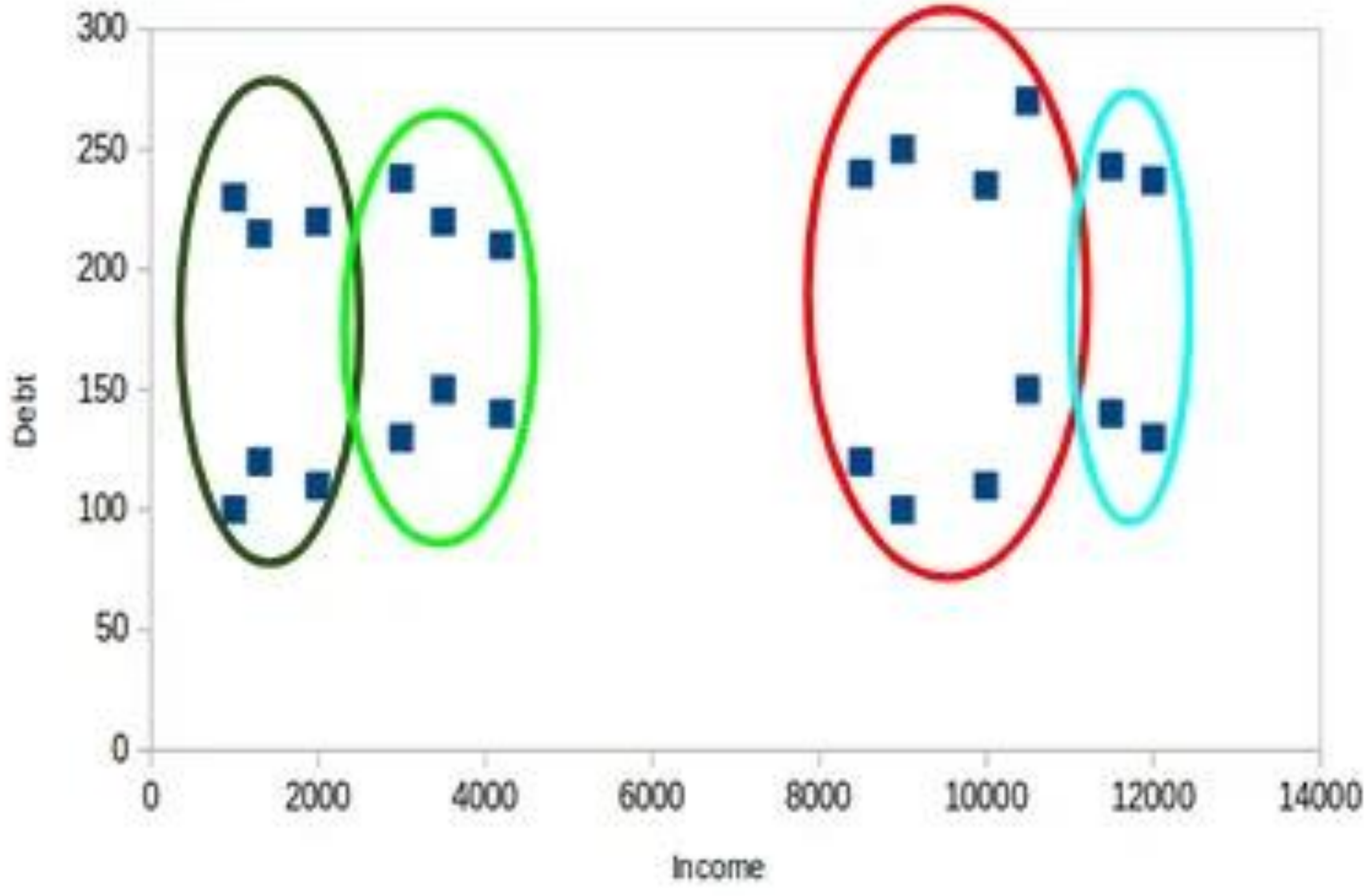
Target

Standard

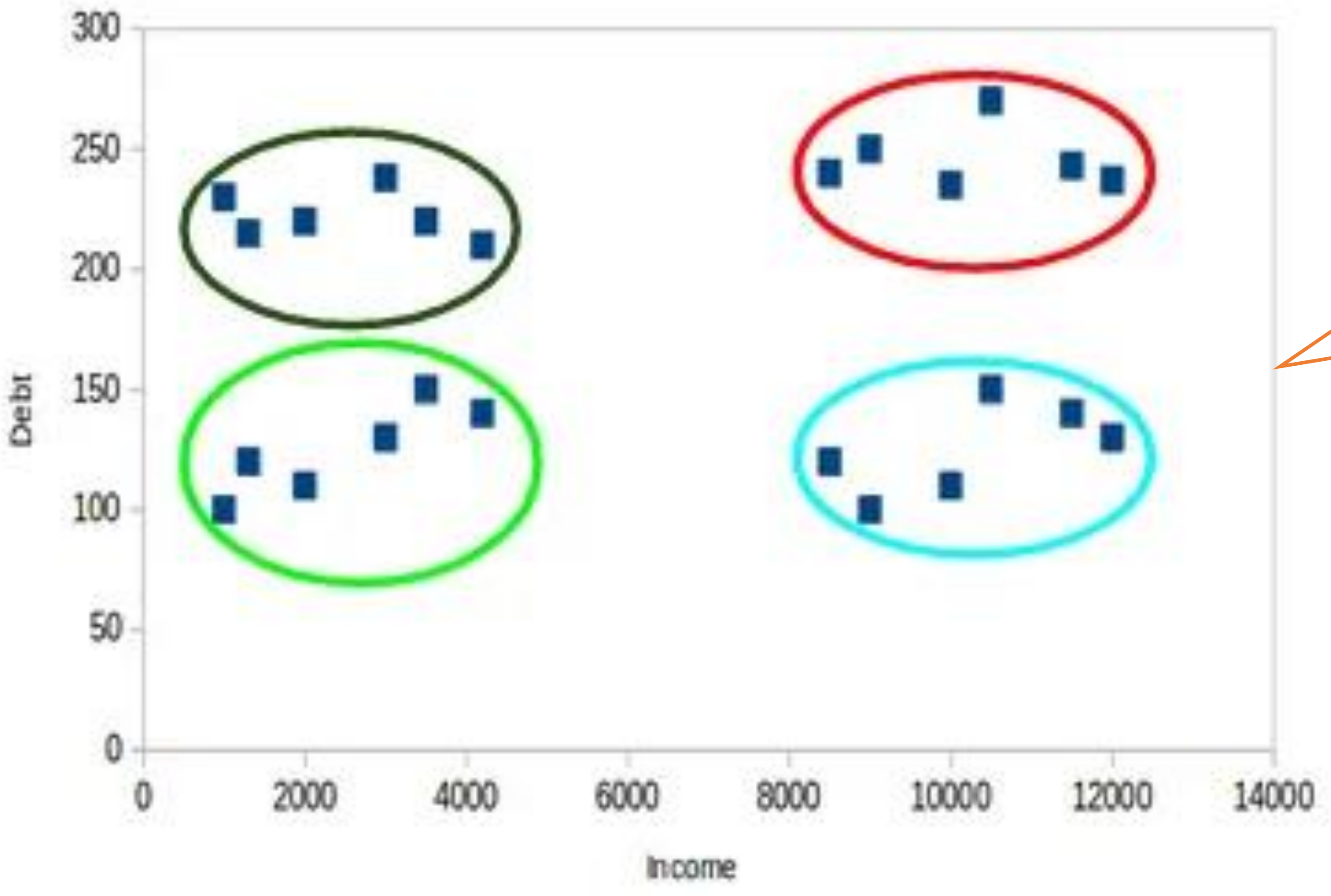
Careful

Sensible

Cluster properties



Case - I



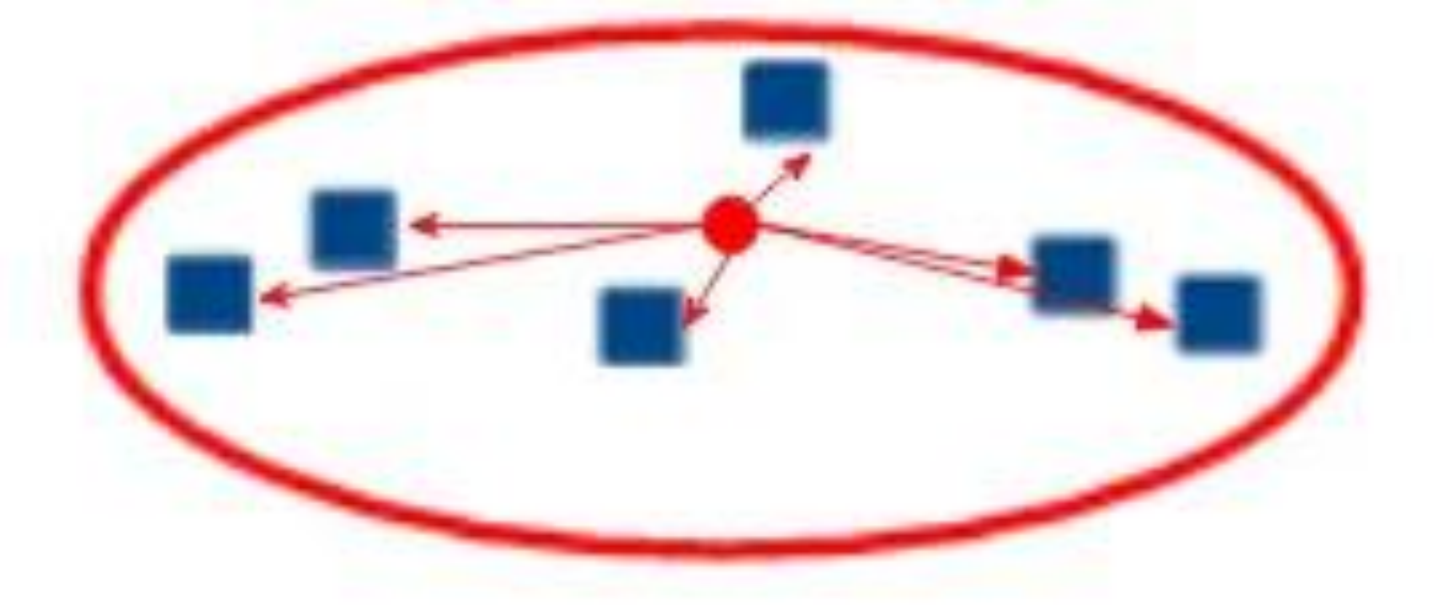
Case - II

Which is better cluster?

- Data points in cluster should have as many similar properties as possible (compactness)
- Data points in different clusters should be as different as possible (separation)

Cluster properties – Similar data points

- Data points in cluster should have as many similar properties as possible
- How to quantify this?
  - Distance between intra-cluster points should be as low as possible



Intra cluster distance

- Sum/avg of Euclidean distances from centroid

$$\frac{1}{|C_i|}\sum_{x\in C_i}(x-\mu_i)^2$$

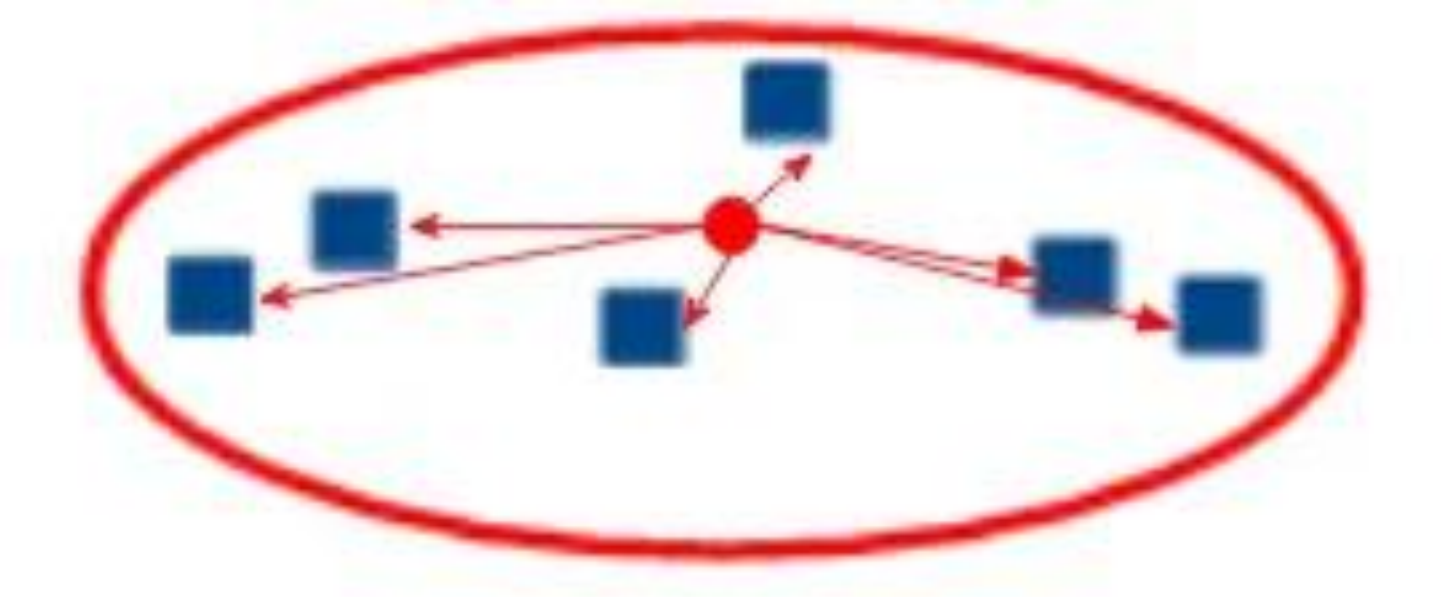
- This is nothing but variance
- Variance is a metric!

Cluster metric – Inertia

- How far the points are within a cluster

Optional normalizing term

$$\frac{1}{|C_i|}\sum_{x\in C_i}(x-\mu_i)^2$$



Intra cluster distance

- Across all clusters

$$Inertia = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{|C_i|}\sum_{x\in C_i}(x-\mu_i)^2$$

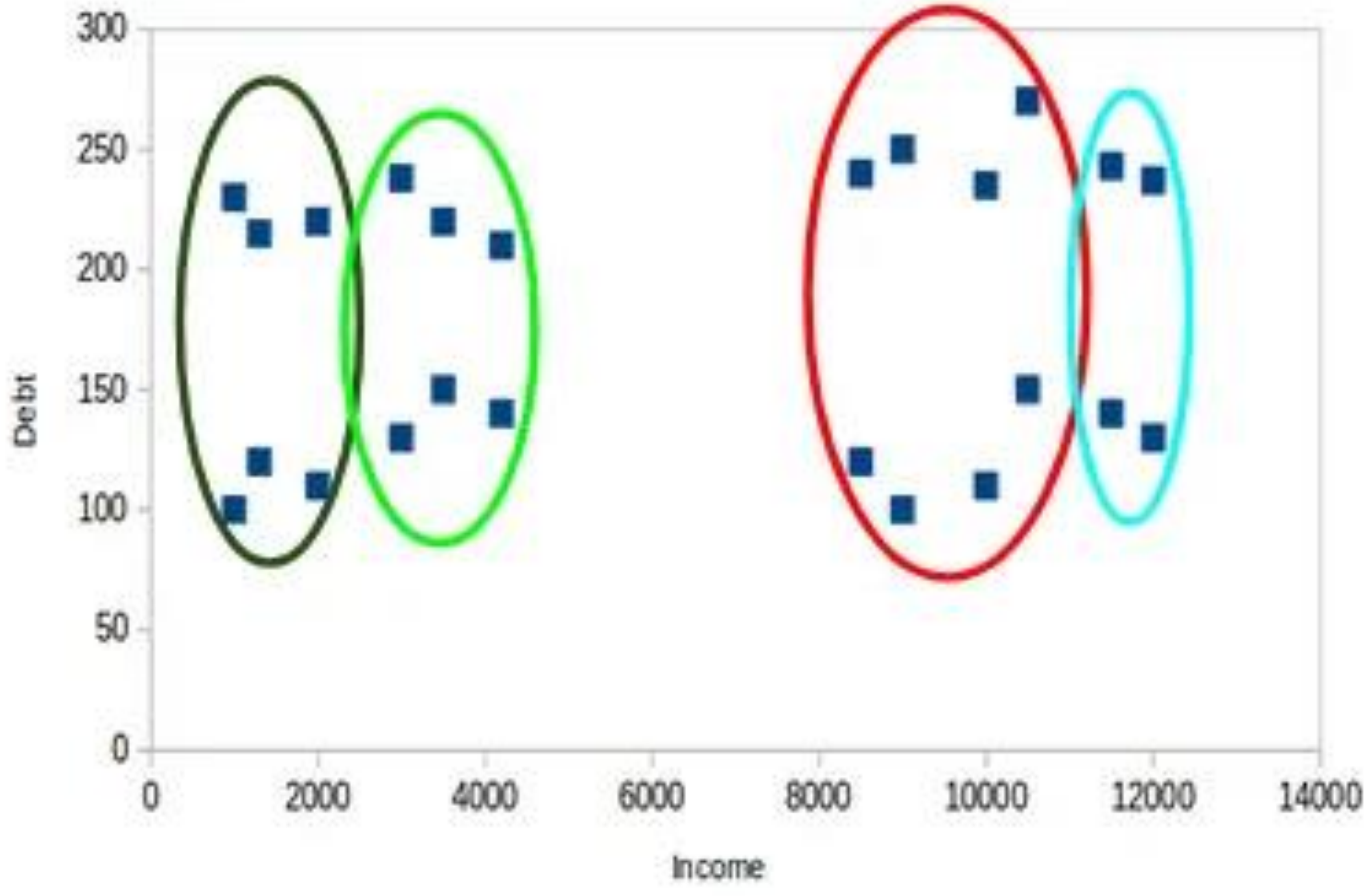
Also an optional normalizing term

$$\sum_{i=1}^{k}\sum_{x\in C_i}(x-\mu_i)^2$$

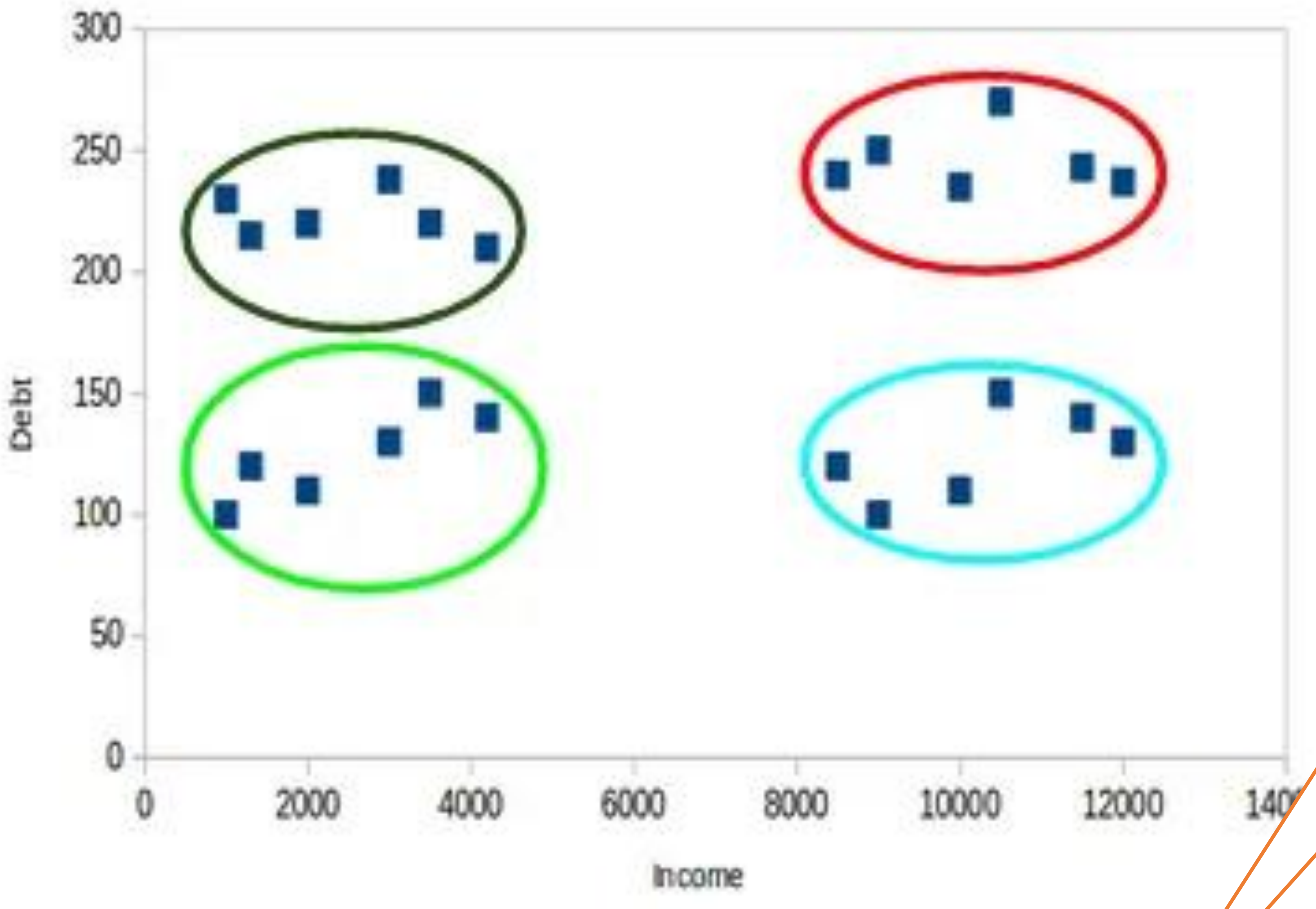
Sklearn calculates inertia without normalizing term

- Inertia also known as WCSS (Within cluster sum of squares)
- Low inertia = better cluster

Cluster metric - Recap



Case - I



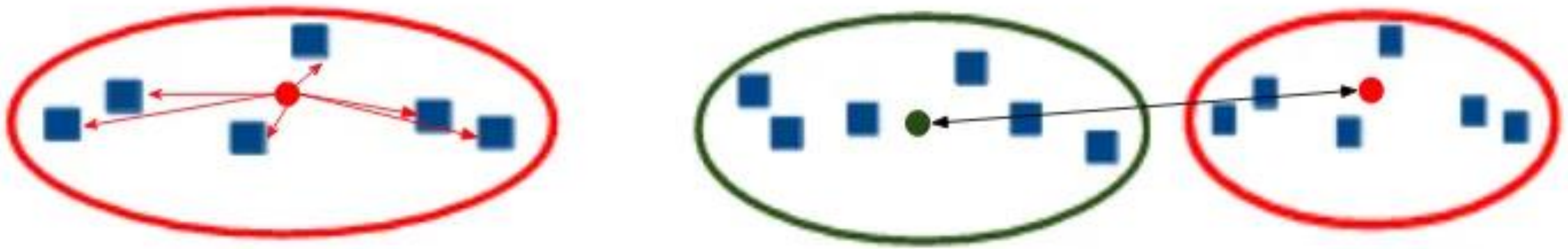
Case - II

Inertia only accounted for this

What about this?

- Data points in cluster should have as many similar properties as possible
- Data points in different clusters should be as different as possible

Cluster metric – Dunn Index

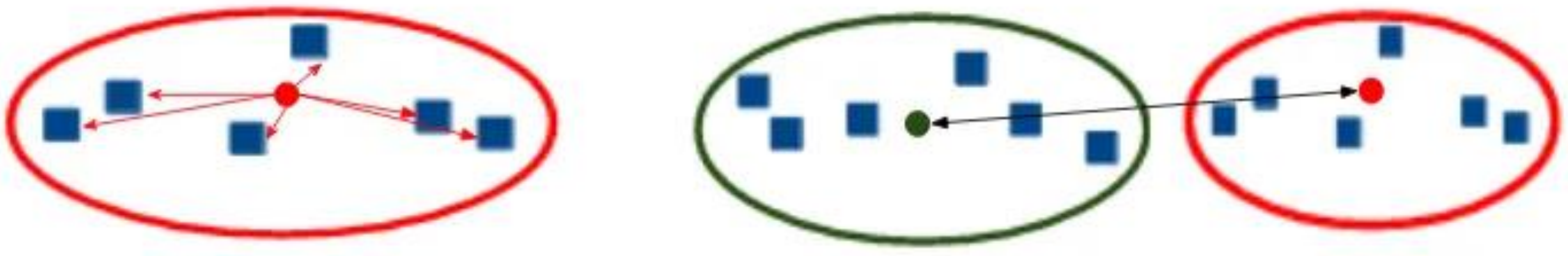


Intra cluster distance

Inter cluster distance

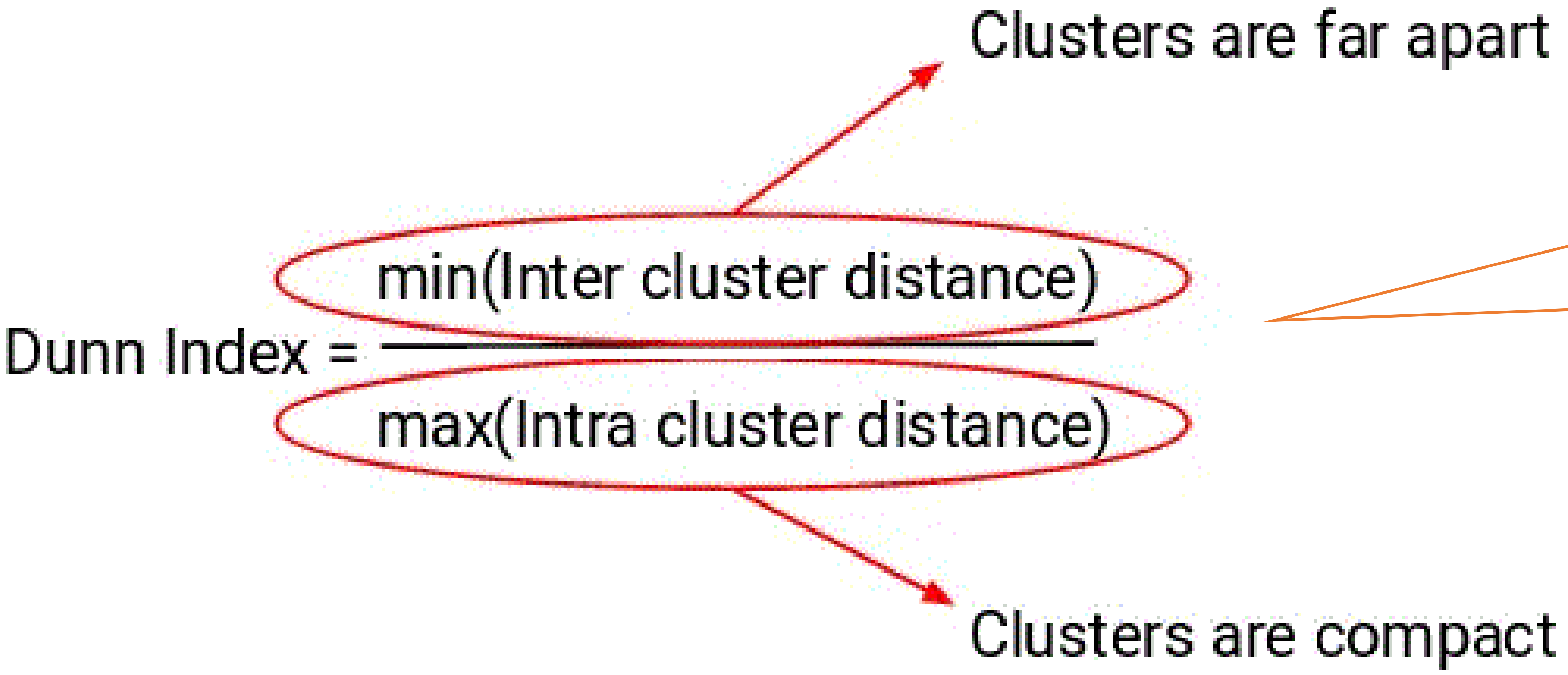
$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{Intra cluster distance})}$$

Cluster metric – Dunn Index (one of many)



Intra cluster distance

Inter cluster distance



Clusters are far apart

$$\text{Dunn Index} = \frac{\text{min(Inter cluster distance)}}{\text{max(Intra cluster distance)}}$$

Clusters are compact

Dunn Index considers worst case of Davis Bouldin index

Cluster metrics recap

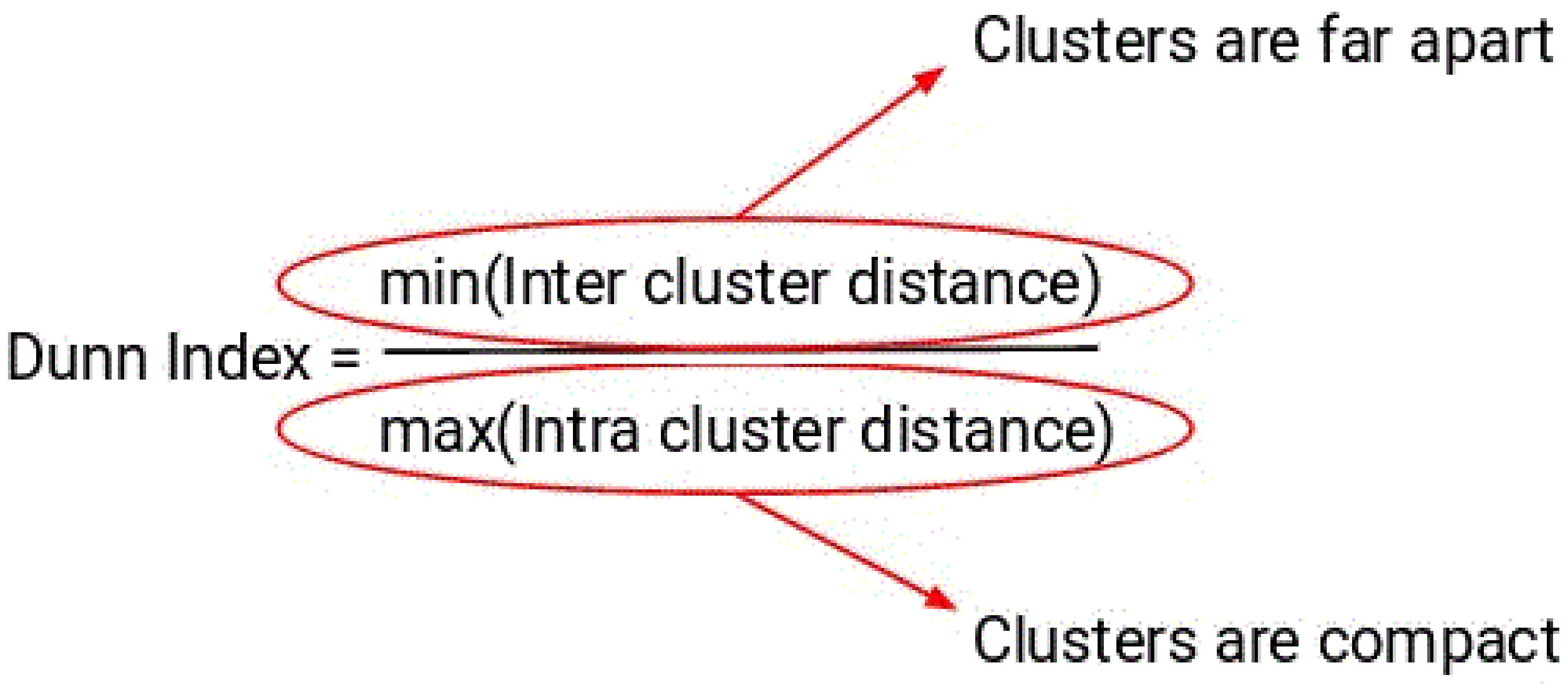
- Inertia (WCSS) – Lower the better

$$Inertia = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{|C_i|}\sum_{x\in C_i}(x-\mu_i)^2$$

Used for choosing K

- Dunn Index – Higher the better

Used for cluster evaluation



$$\text{Dunn Index} = \frac{\text{min(Inter cluster distance)}}{\text{max(Intra cluster distance)}}$$

Clusters are far apart

Clusters are compact

Selecting the best K for clustering

- Selecting by Cross validation with one of:
  - Elbow method
  - Silhouette score
  - Gap Statistic (not part of syllabus)
- Davis-Bouldin Index, Dunn Index not used
- Cluster Validation Indices (CVI)
  - Internal Validation Indices
  - External Validation Indices
  - Relative Validation Indices

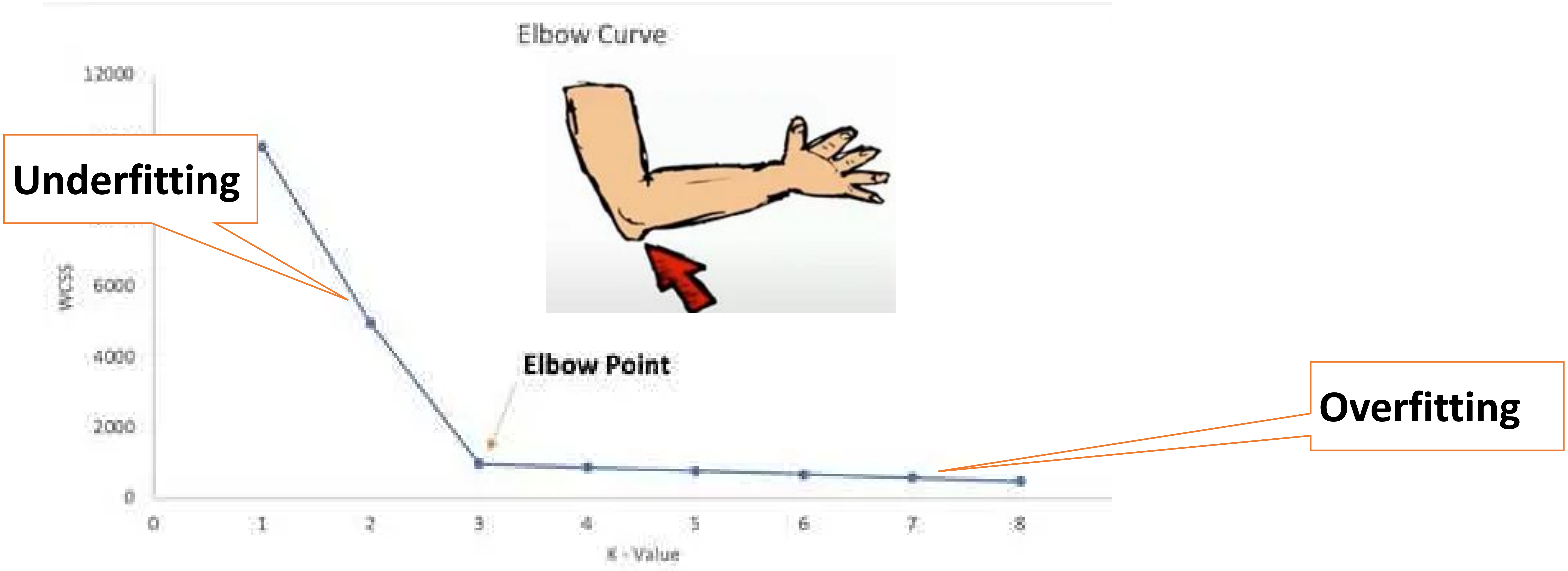
Relies on cluster goodness labels to measure clustering efficacy

Relies on labels to measure clustering efficacy

For choosing K, Internal validation index should also function as a good relative validation index

Selecting the best k with elbow method

Elbow plot

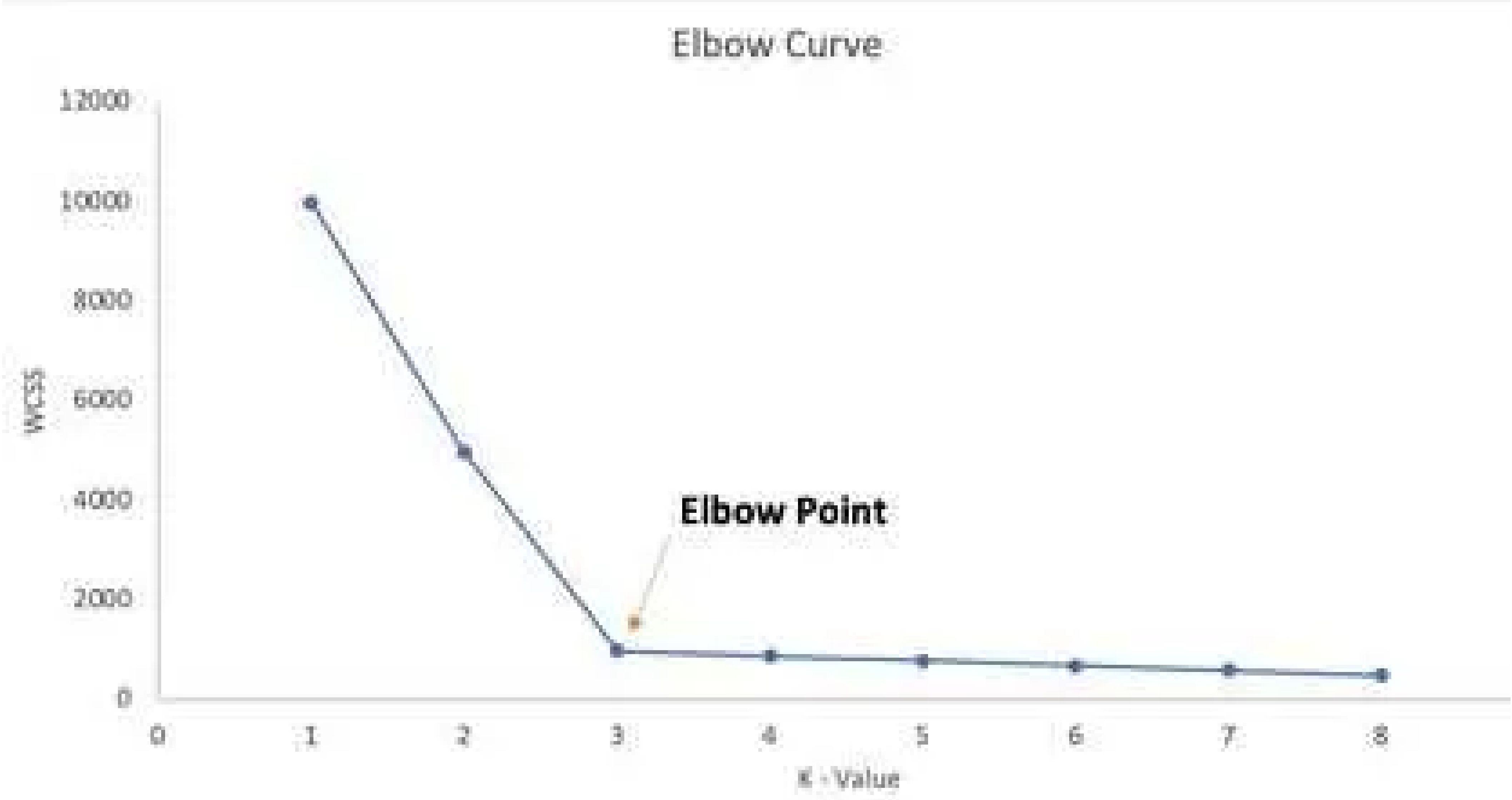


Underfitting

Overfitting

- Increasing k decreases inertia. But is it worth it?
- Cost of tuning parameter is no longer worth the benefit

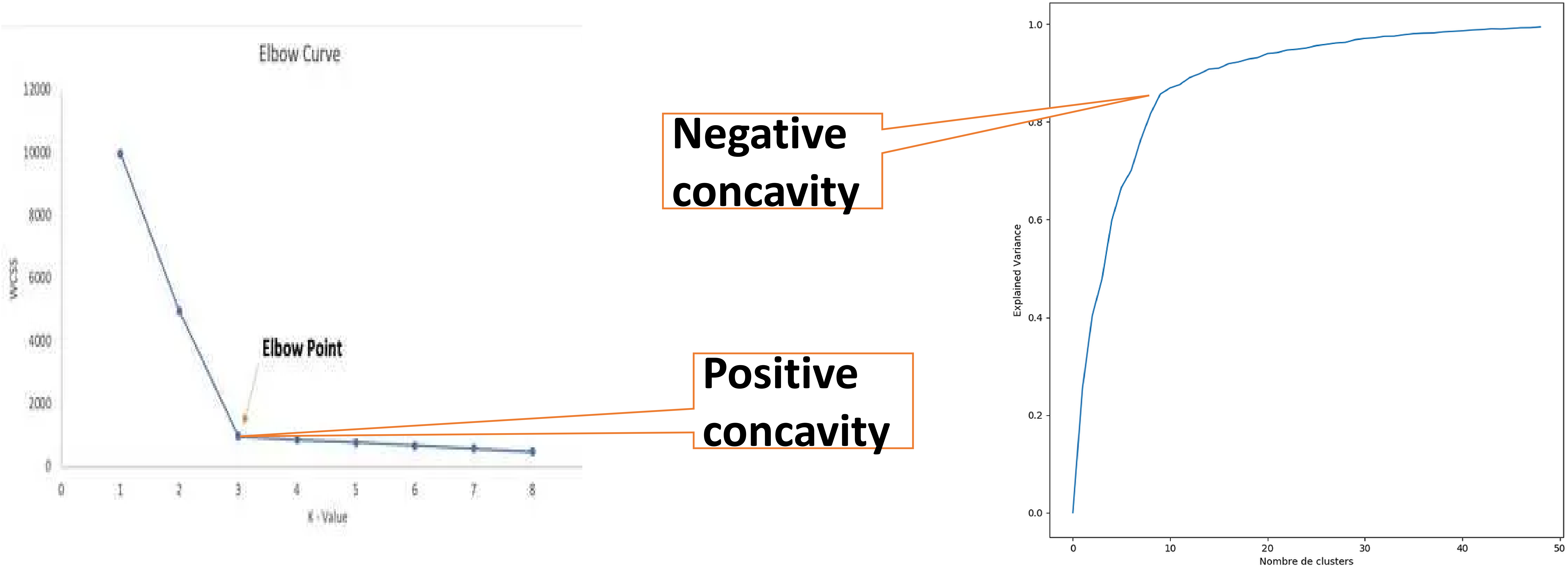
Elbow plot: Cons



- Needs visual examination
- Not possible in automated environments

Knees & Elbows

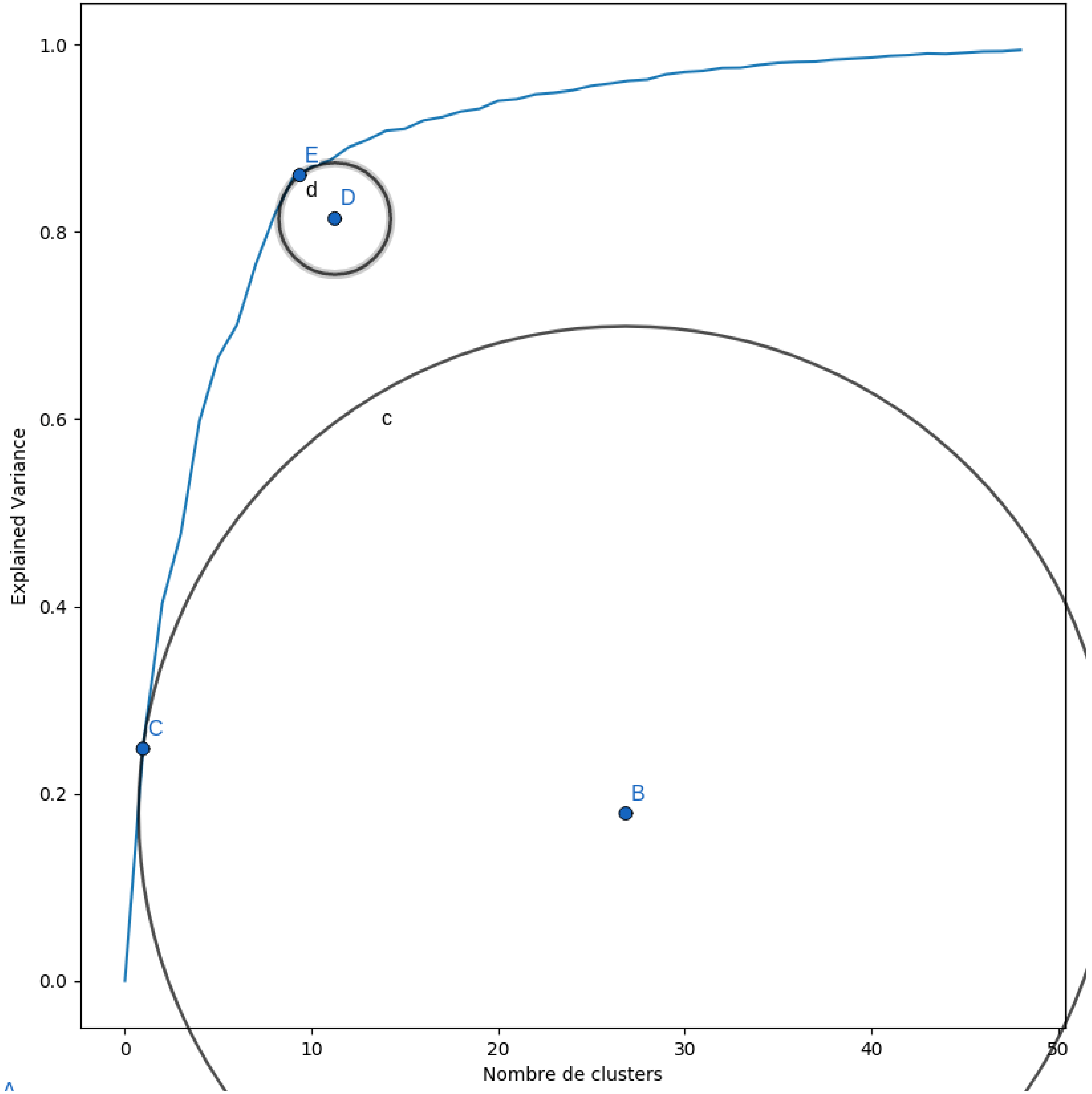
- Automated detection based on curvature
- Elbow – Decrease in rate of decrease of WCSS
- Knee – Decrease in rate of increase of “explained variance”



Negative concavity

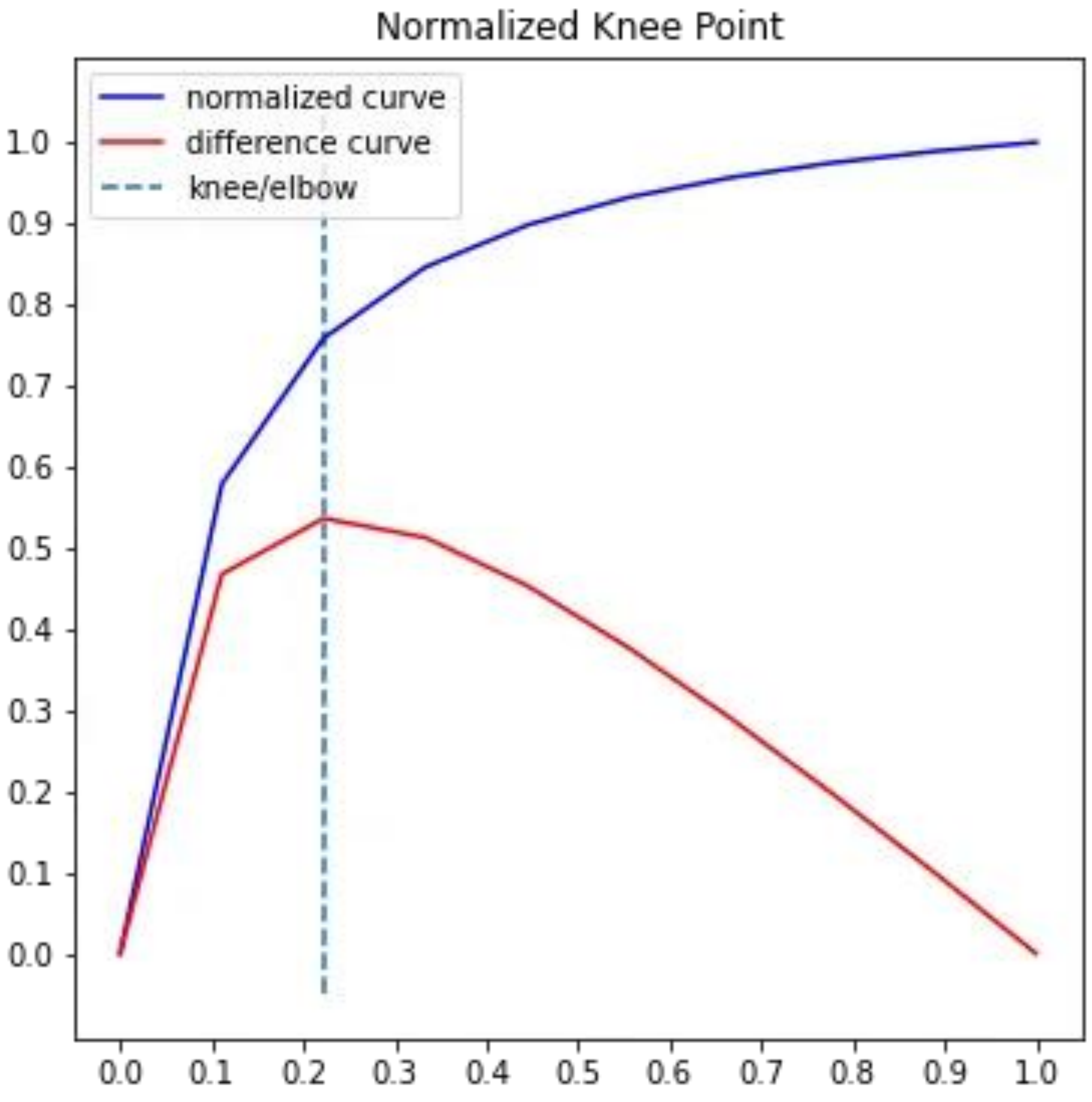
Positive concavity

Curvature at Knees & Elbows



- Curvature: Draw **largest** circle where the tangent touches the curve
    - Such that circle touches the curve only at one point
- Smaller the circle, larger the curvature
- Knees & elbows – Region where curvature is most drastic

Curvature at Knees & Elbows conceptually

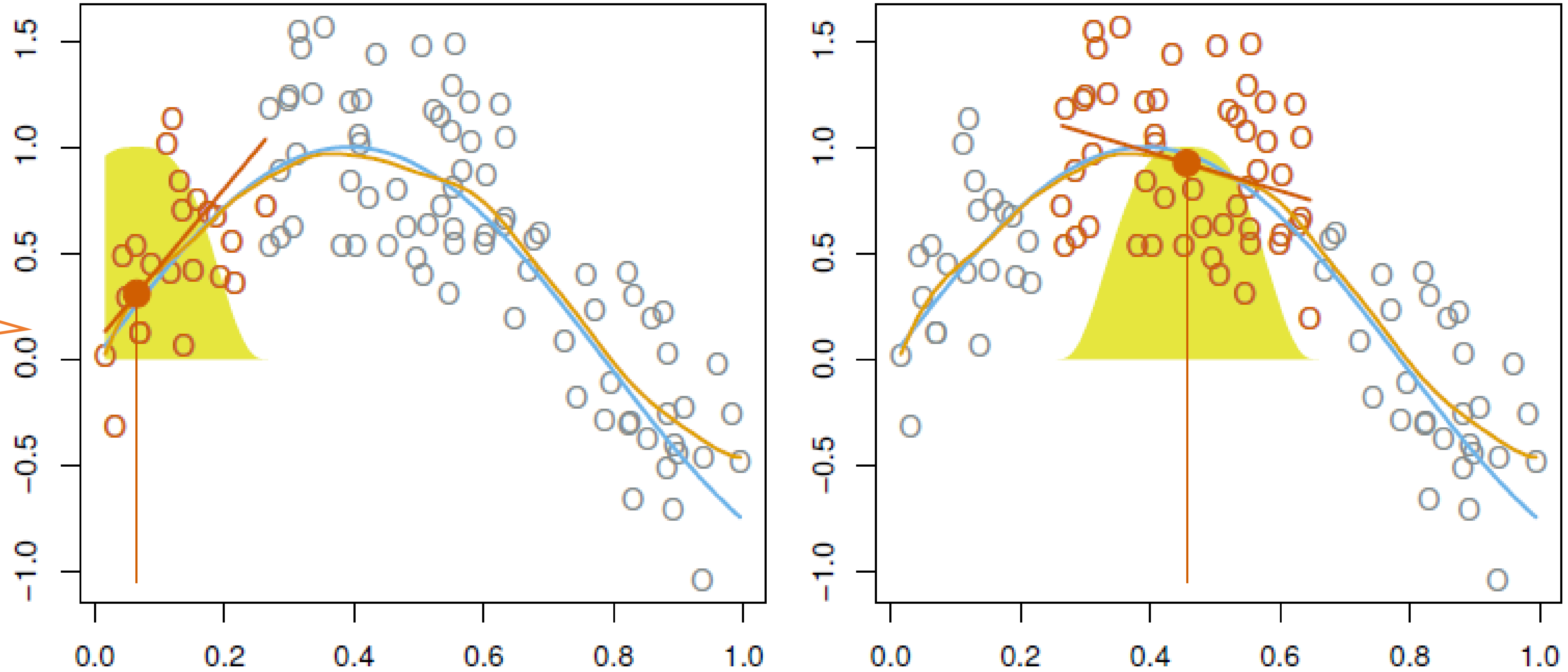


- Second derivative
- Knee – Maxima of second derivative
- Elbow – Minima of second derivative
- Kneed library available
- Based on paper IEEE
- Finding a Kneedle in haystack
- Paper is slightly more involved than simple second derivative

Problems to address in applying kneedle

- Need to find second derivative. Knots aren't differentiable
- Smoothing spline should be applied first
- Smoothing spline = Regression Spline + Regularization
- Regression Spline = Piecewise continuous polynomial regression

Learn more about spline in Intro to Statistical Learning in Gareth, Tibshirani etal.



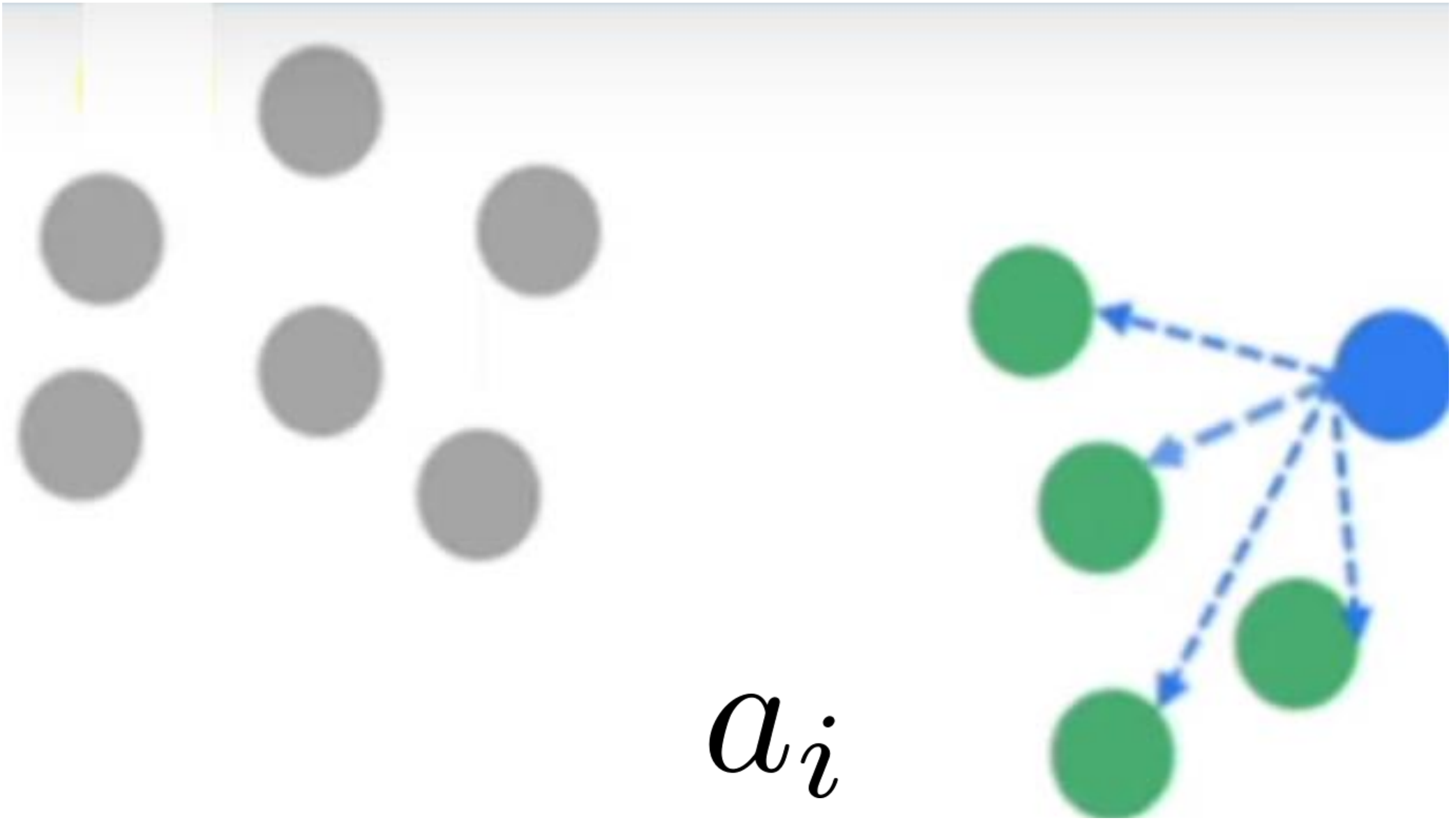
Selecting the best k with Silhouette score

Silhouette score intuition

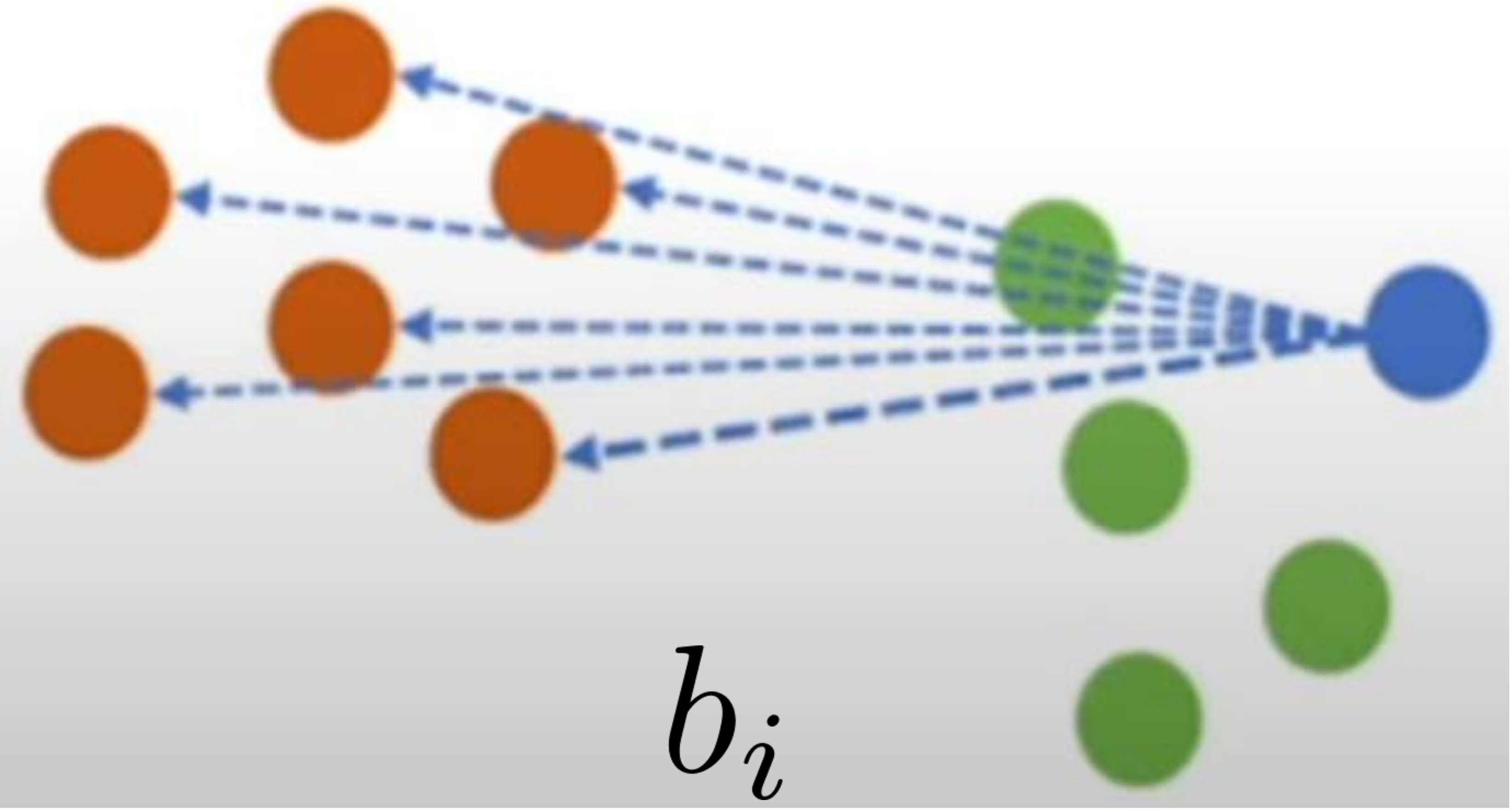
- Overall representative score of clustering using
  - Compactness of individual clusters (intra cluster distance)
  - Separation between clusters (inter cluster distance)
- Uses both intra cluster cohesion & inter cluster separation

$$s_i = \frac{b_i - a_i}{max(b_i, a_i)}$$

Silhouette score intuition



Avg distance between i & data points in same cluster

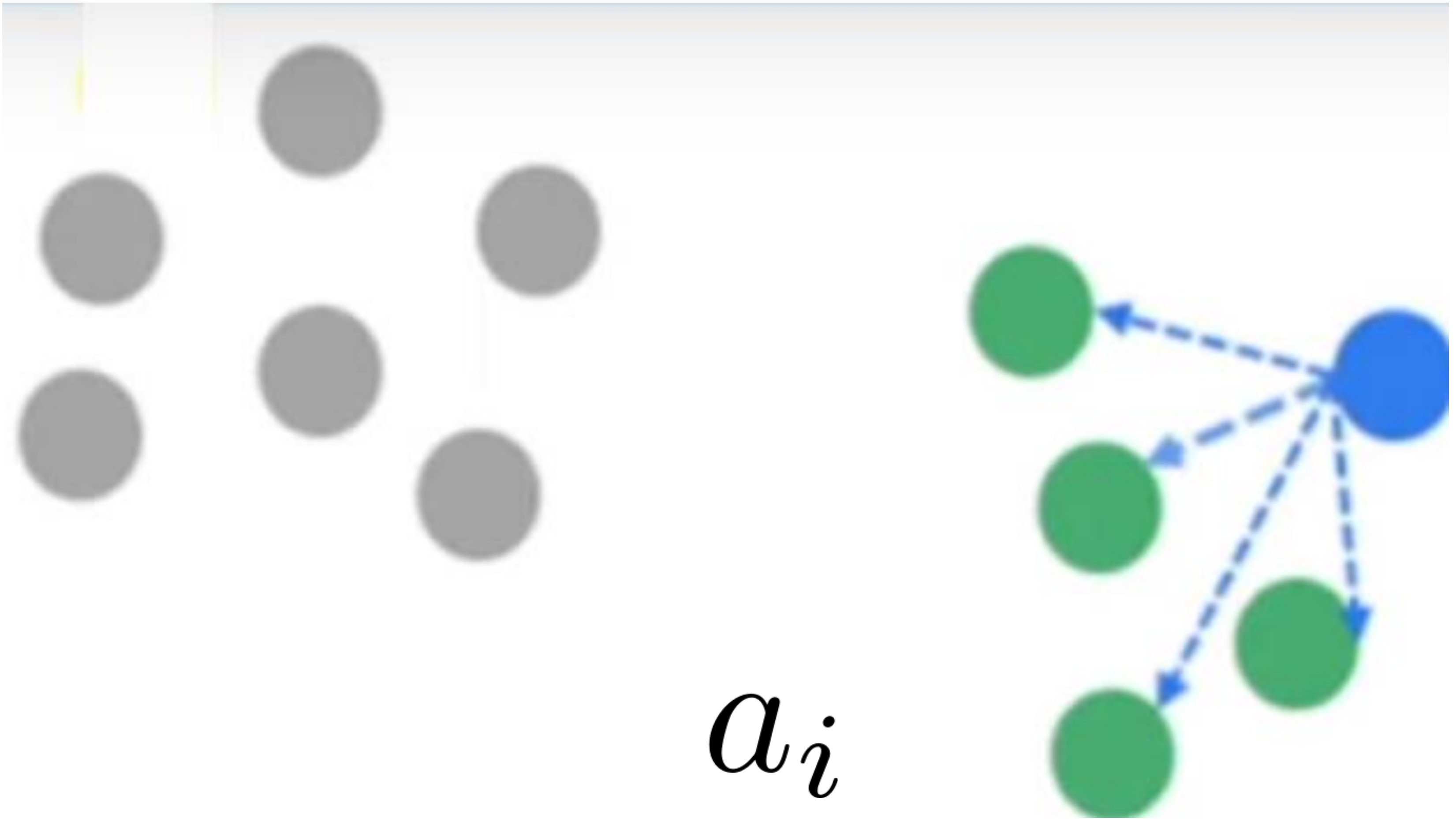


Avg distance between i & data points in other clusters

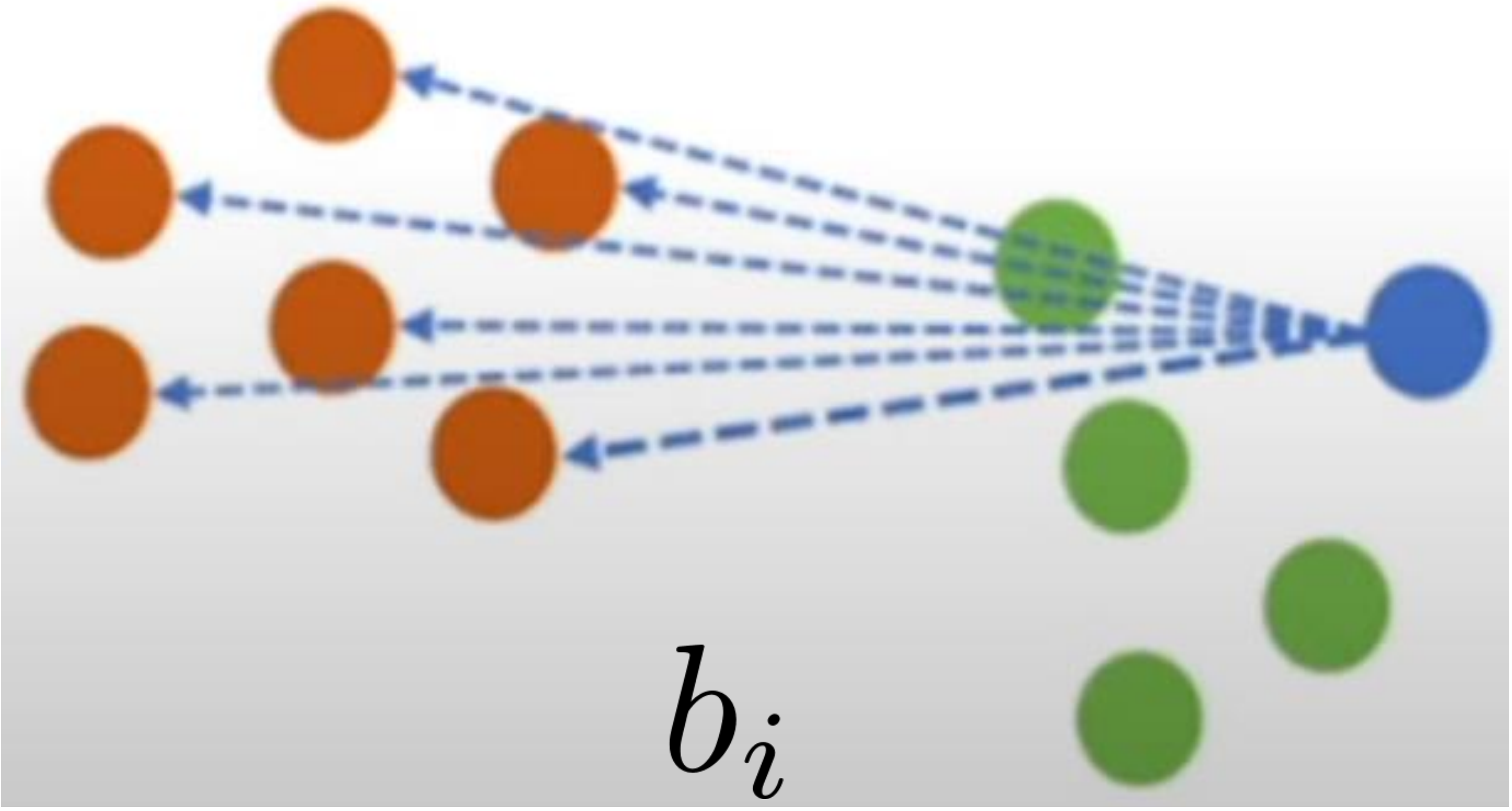
$$s_i = \frac{b_i - a_i}{max(b_i, a_i)}$$

- Measured for each data point
- Between -1 and +1

Silhouette score intuition (contd.)



Avg distance between i & data points in same cluster



Avg distance between i & data points in other clusters

$$s_i = \frac{b_i - a_i}{max(b_i, a_i)}$$

$$Silhouette - Score = \frac{1}{n}\sum_{i=1}^{n} s_i$$

Silhouette score summary

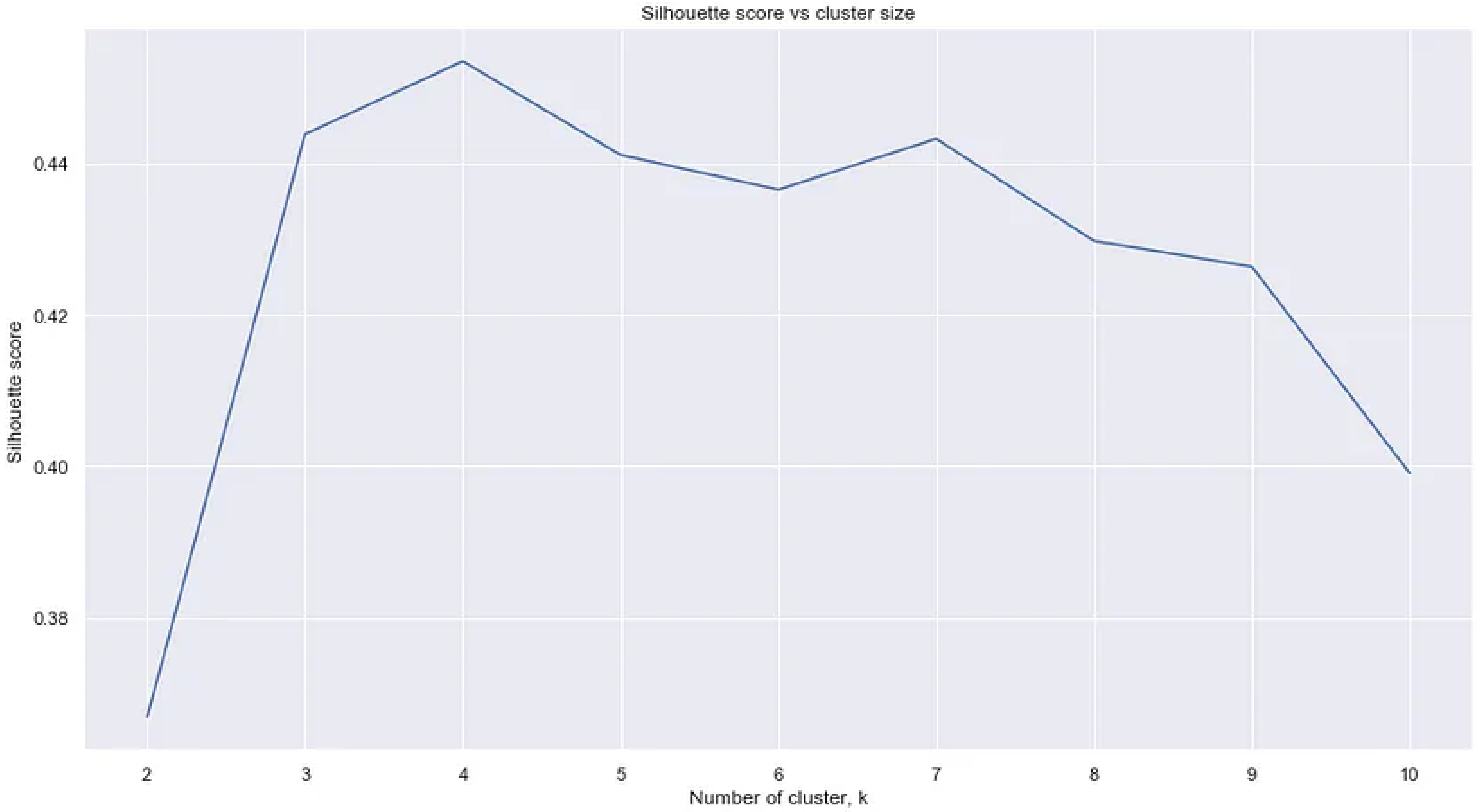
$$s_i = \frac{b_i - a_i}{max(b_i, a_i)}$$

$$a_i = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j) \qquad b_i = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

$$Silhouette - Score = \frac{1}{n} \sum_{i=1}^{n} s_i$$

- Implementation available in sklearn

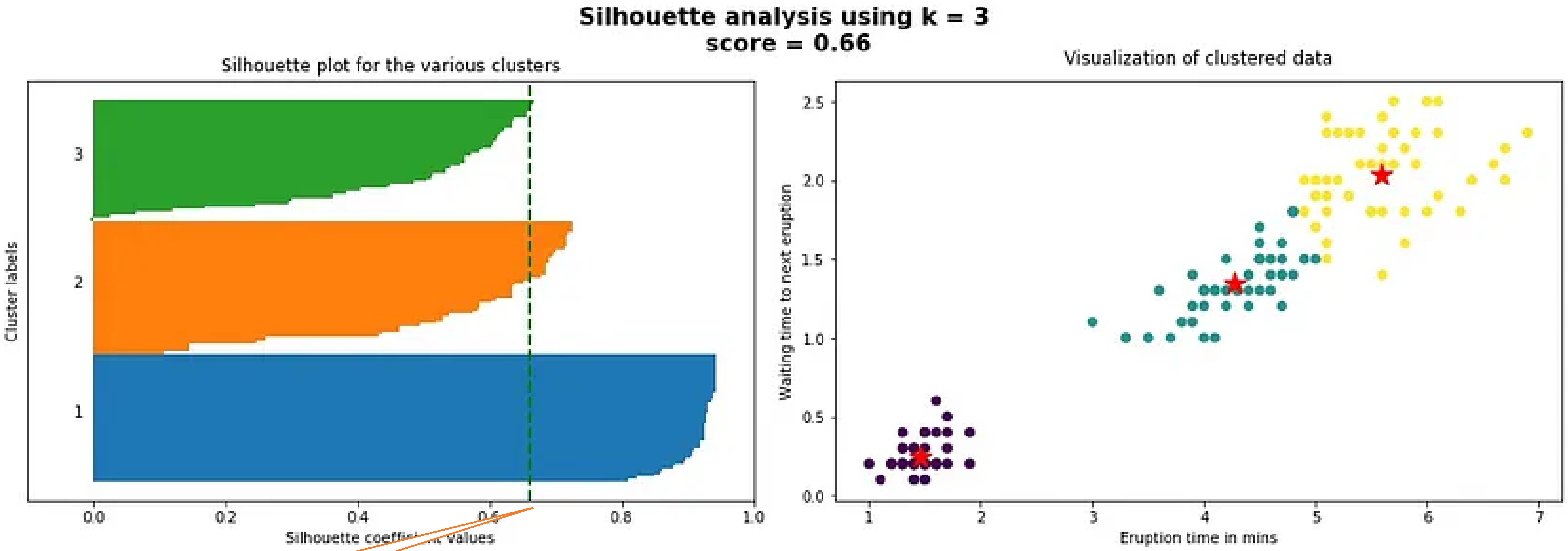
Silhouette score plot



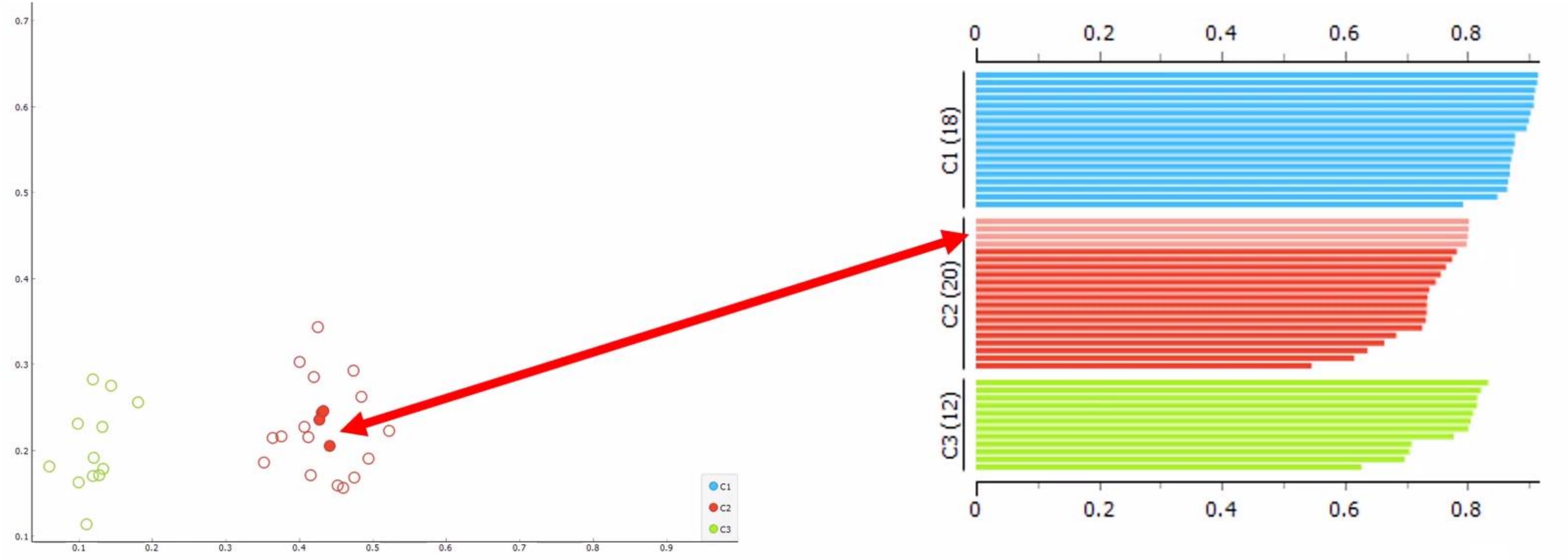
- Choose k with largest Silhouette score

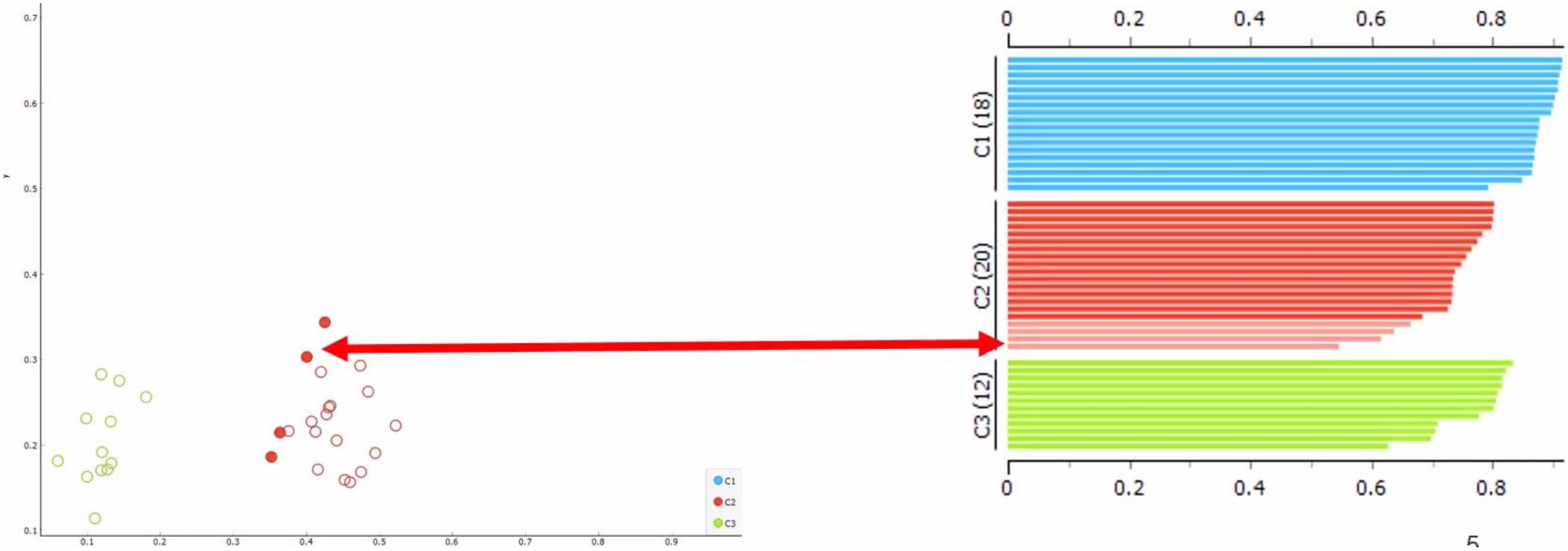
Silhouette analysis with silhouette plot

- How to read a silhouette plot?

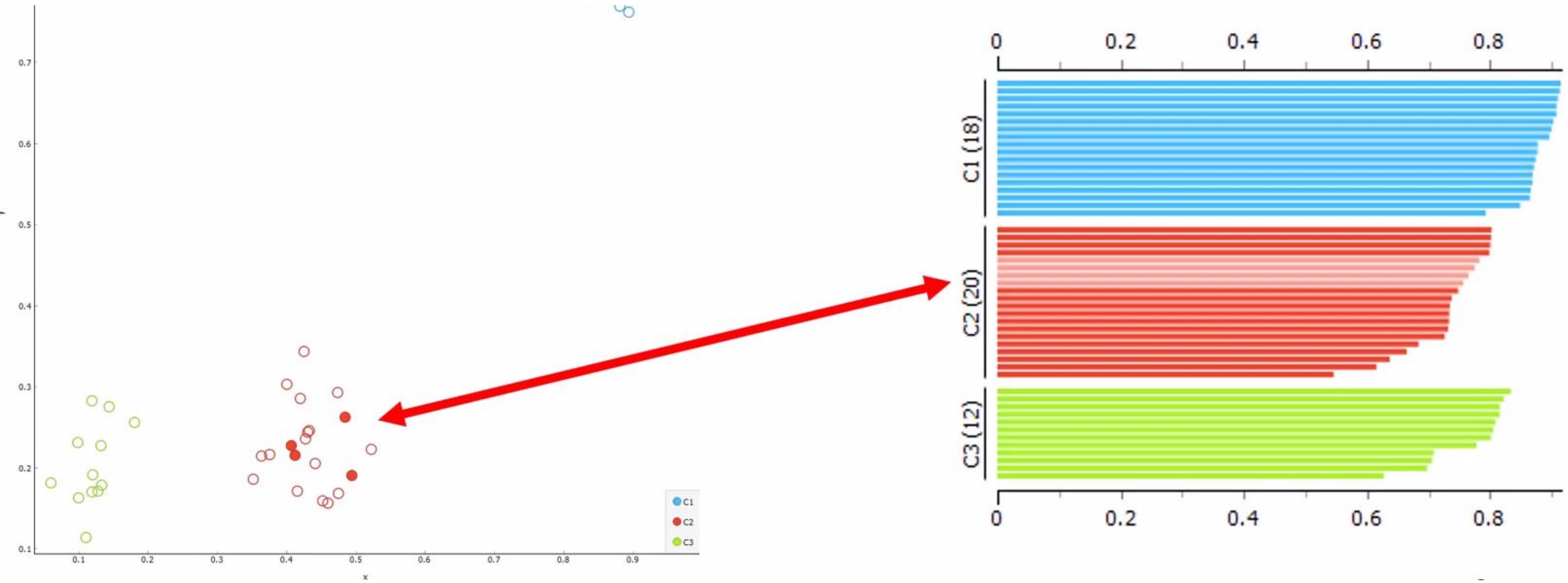


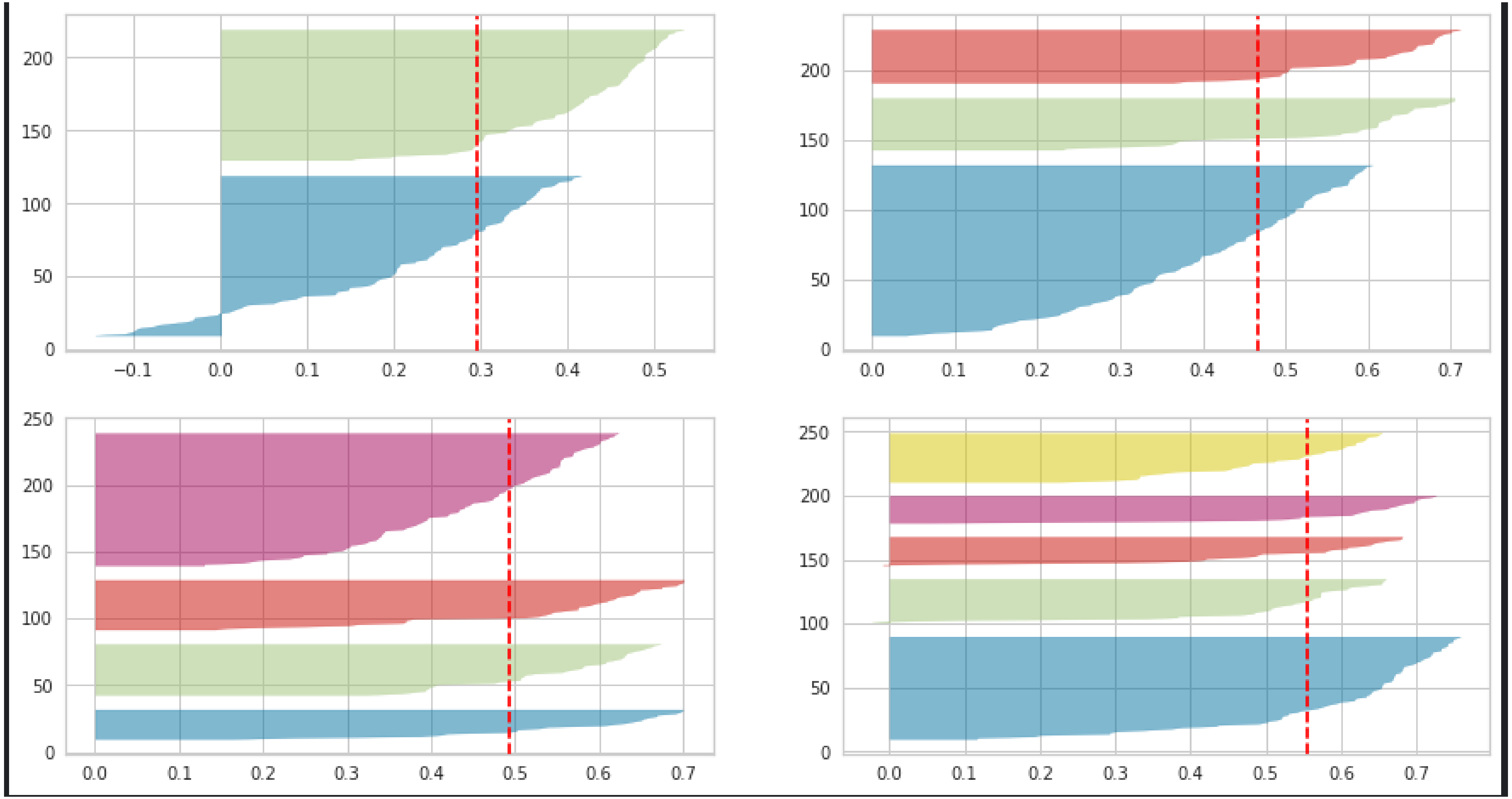
Silhouette score (avg)


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
C1
C2
C3
0
0.2
0.4
0.6
0.8
C1 (18)
C2 (20)
C3 (12)
0
0.2
0.4
0.6
0.8



y
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
C1
C2
C3
0
0.2
0.4
0.6
0.8
C1 (18)
C2 (20)
C3 (12)
0
0.2
0.4
0.6
0.8
5


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
x
C1
C2
C3
0
0.2
0.4
0.6
0.8
C1 (18)
C2 (20)
C3 (12)
0
0.2
0.4
0.6
0.8

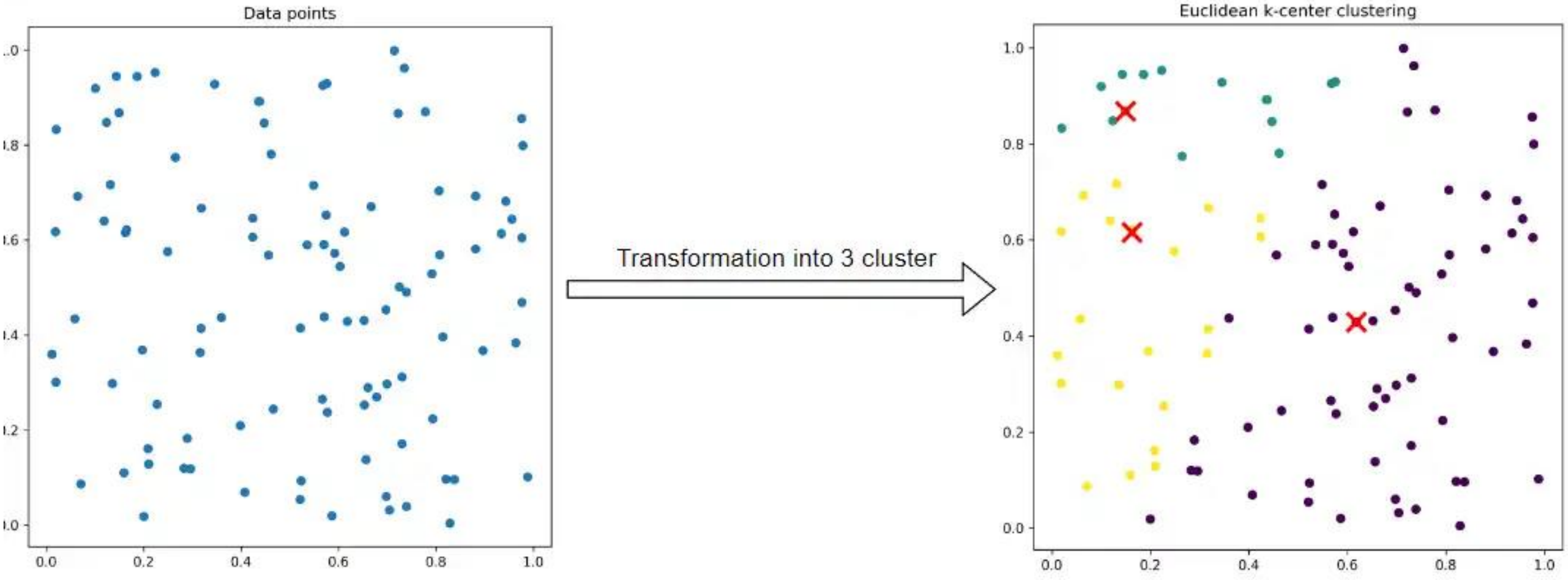


- Sub optimal cluster will show
  - Clusters below avg. score
  - Wide fluctuation in plot

P, NP, NP-hard, NP-complete refresher & clustering complexity

K-means demo

What is the cost of K-Center Clustering?

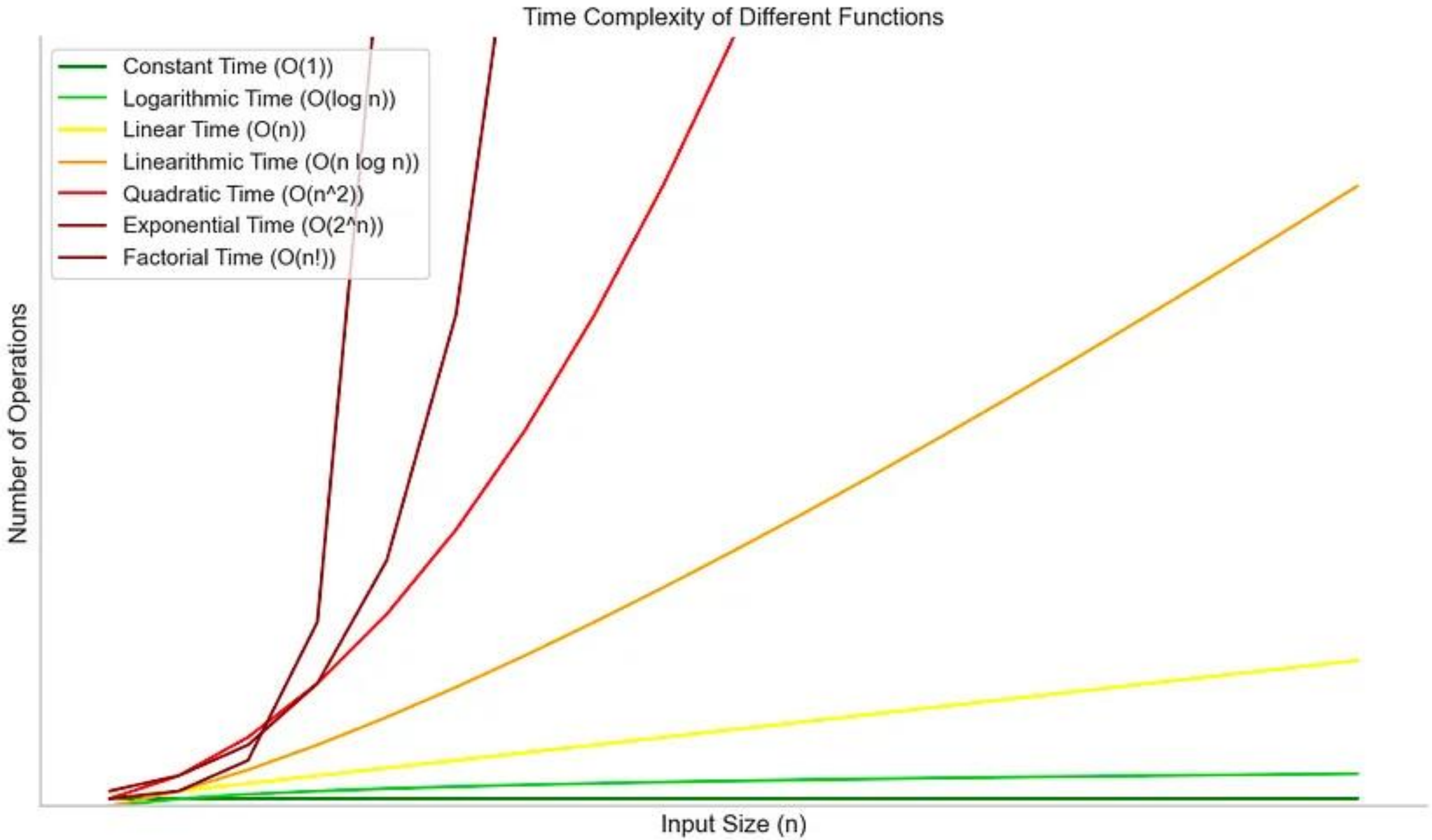


Transformation into 3 cluster

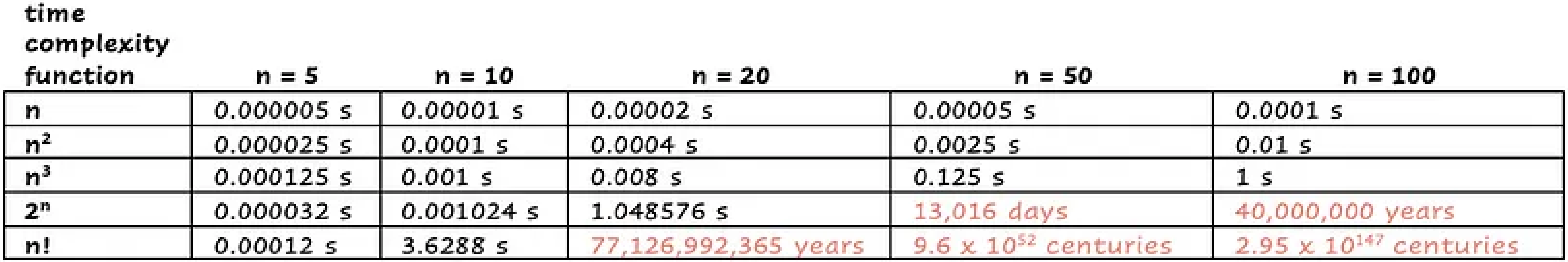
$$\binom{100}{3} = \frac{100!}{3!\ \ 97!}$$

$$\binom{n}{k} = \frac{n!}{k!\ \ (n-k)!}$$

Analyzing time complexity



Analyzing time complexity of K-Centers



time complexity function	n = 5	n = 10	n = 20	n = 50	n = 100
n	0.000005 s	0.00001 s	0.00002 s	0.00005 s	0.0001 s
n^2	0.000025 s	0.0001 s	0.0004 s	0.0025 s	0.01 s
n^3	0.000125 s	0.001 s	0.008 s	0.125 s	1 s
2^n	0.000032 s	0.001024 s	1.048576 s	13,016 days	40,000,000 years
n!	0.00012 s	3.6288 s	77,126,992,365 years	9.6×10^{52} centuries	2.95×10^{147} centuries

- It didn’t take us centuries to execute K-Means. Why?
- We used non-deterministic algorithm
  - Expectation-Maximization (EM)
  - K-centers is NP-Complete problem with deterministic approach
  - We used heuristic (initial centroid selection)

P versus NP decision problem

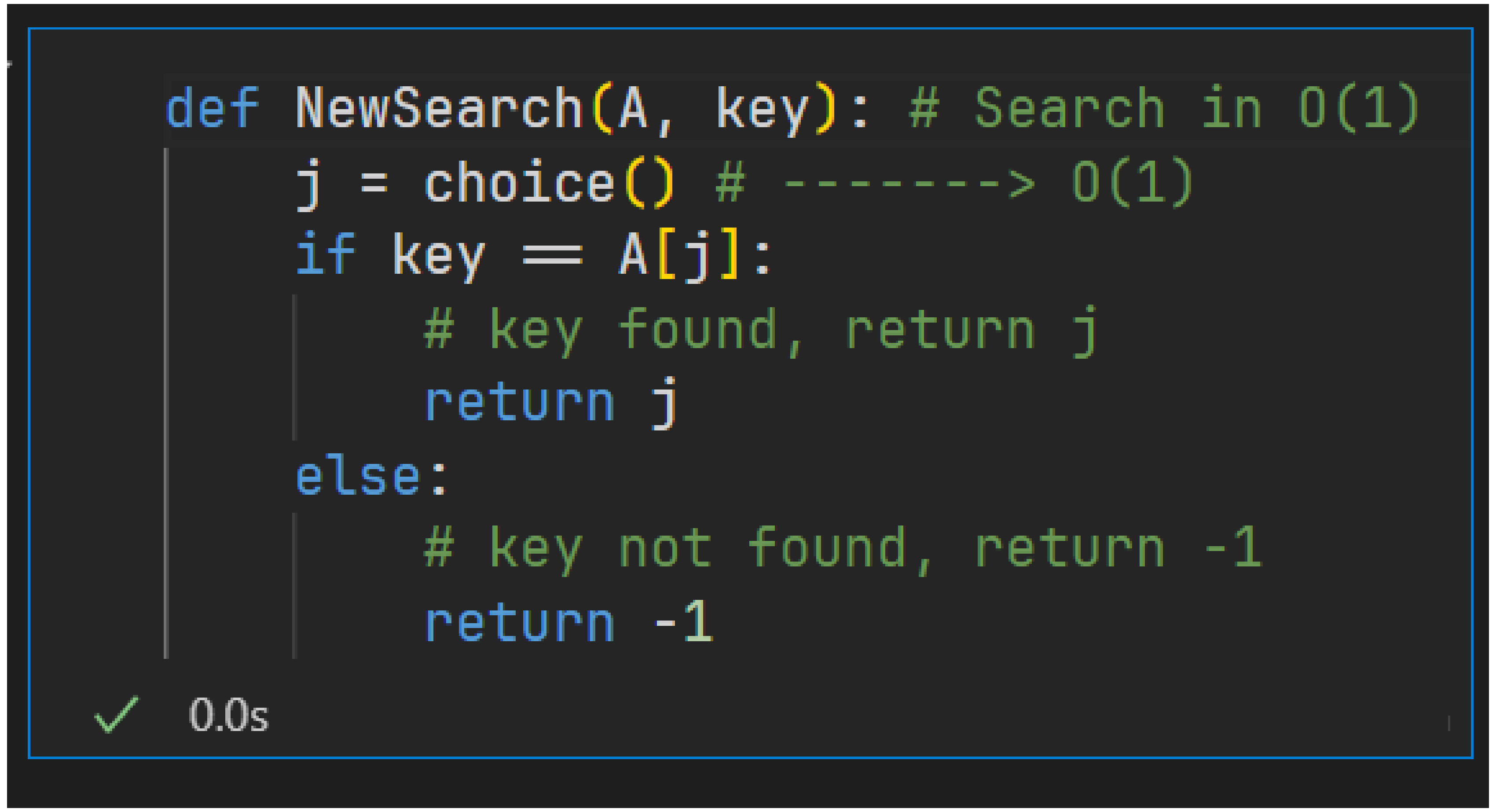
P	NP
Solvable in polynomial time	Solvable in exponential time
Linear search – n Binary search – log n Bubble sort – n^2 Merge sort – n logn Matrix multiplication – n^3	0/1 Knapsack Travelling salesman Hamiltonian cycle CNF
Verified in polynomial time	Verified in polynomial time
Deterministic in polynomial time	Non-deterministic in polynomial time

- What is non-deterministic?

Non-deterministic

- Deterministic: We clearly know how every statement we code behaves
- Non Deterministic: Most of the statements in algorithm may be deterministic. Some may be non deterministic

- A = [10, 15, 20, 2, 4, 6]
- key = 20
- Returns j = 2 in O(1)
- How?
    - We don’t know (yet)

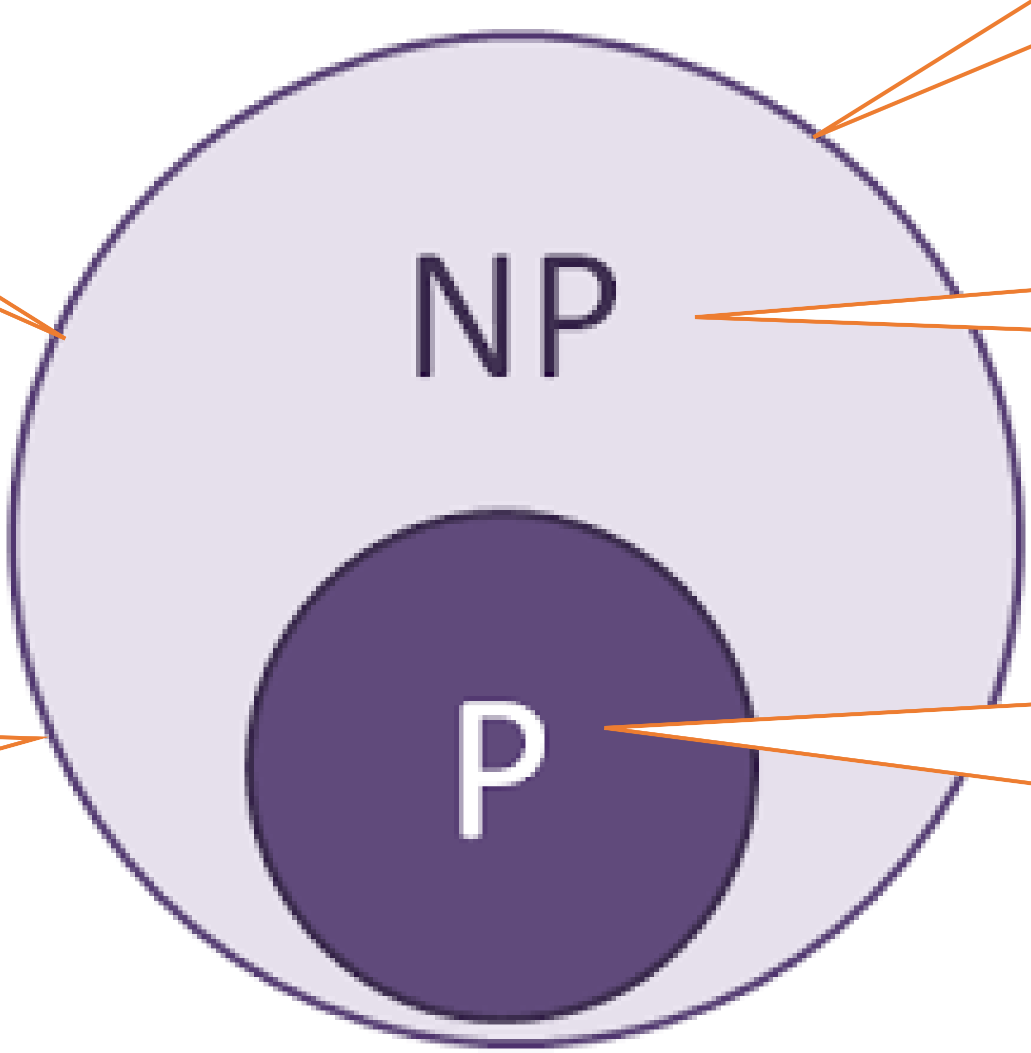


```
def NewSearch(A, key): # Search in O(1)
    j = choice() # -------> O(1)
    if key == A[j]:
        # key found, return j
        return j
    else:
        # key not found, return -1
        return -1
```

✓ 0.0s

NP becoming P

- Before the advent of Binary search
  - Search was O(n)



Why is P subset of NP?

Both verifiable in polynomial time

Sudoku (NP) can be verified in P time

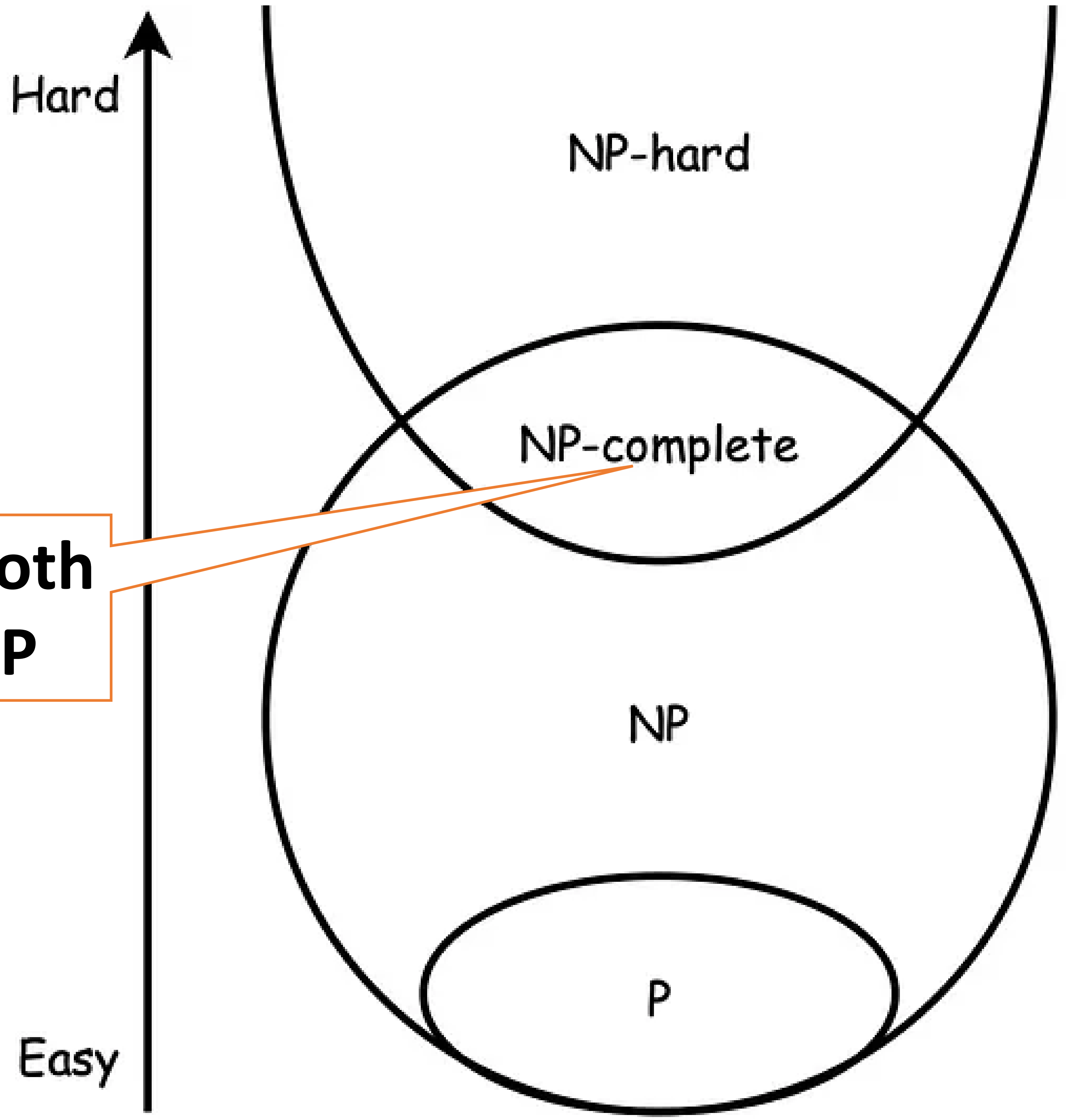
Non deterministic v/s deterministic in polynomial time

Sort (P) can be verified in P time (in addition to solving in P time)

NP-hard NP-complete

- NP-hard problem: Any problem that is at least as hard as the hardest problem in NP
- NP-complete problem is NP-hard that can be verified in polynomial time.
  - E.g. Sudoku, K-centers
- How to determine NP-hard v/s NP-complete
- Reducing NP-hard problems
  - Take a base problem. E.g. CNF

NP-complete is both in NP-hard and NP

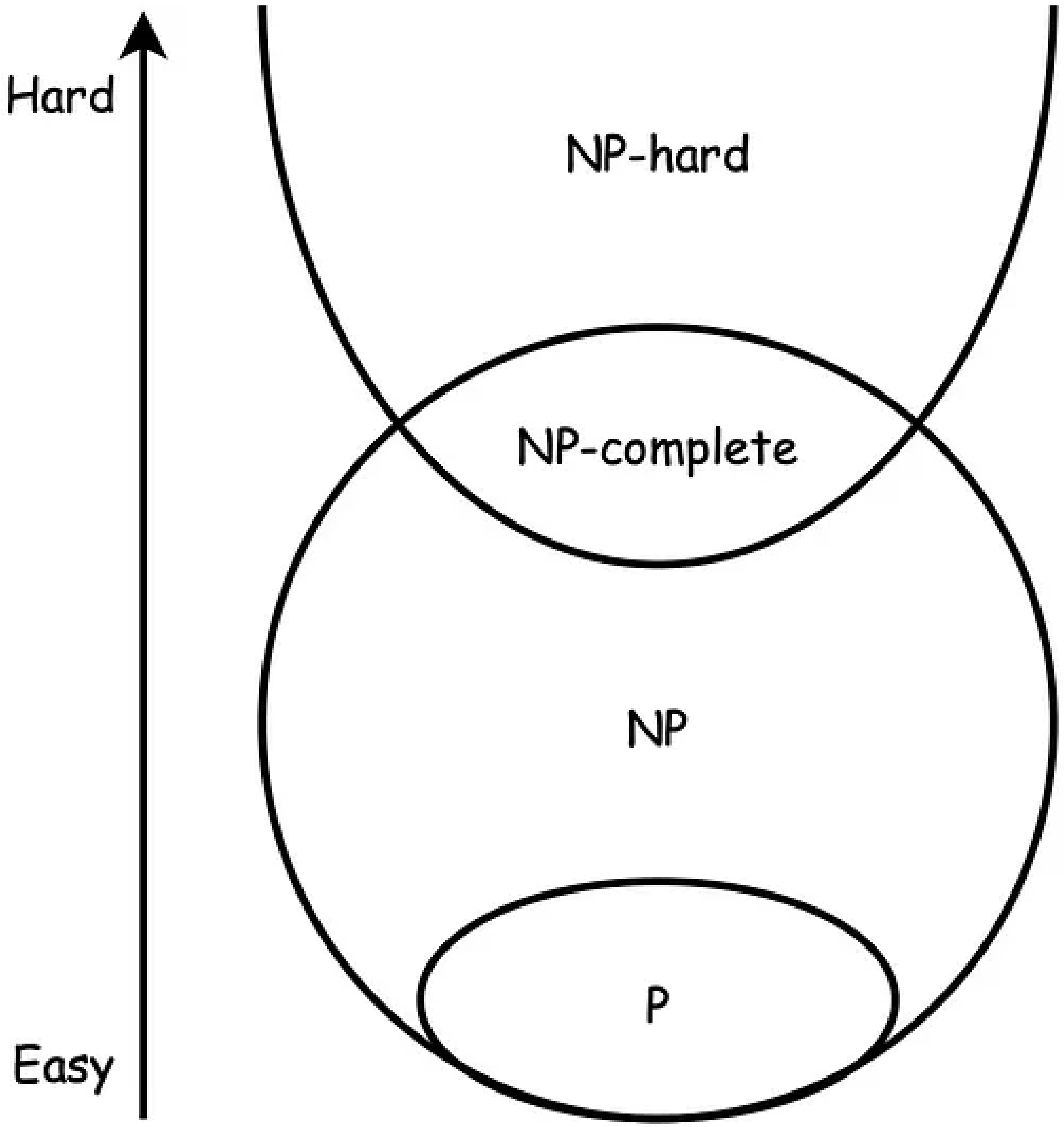


NP-hard NP-complete (contd.)

- CNF – Conjugate Normal Form
- Express logical operations using propositional logic

$$(\neg x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3 \vee \neg x_4)$$

- What values of x1, x2, x3, x4 satisfy this?
- Solve in 2^4 time. Verify in O(n)
- NP-complete problem
- K-centers can be reduced to planar 3-SAT



QUESTIONS