https://github.com/pvateekul/ieat2026

กนอ.
การนิคมอุตสาหกรรมแห่งประเทศไทย

Clustering

Prof. Peerapon Vateekul, Ph.D.
Peerapon.v@chula.ac.th

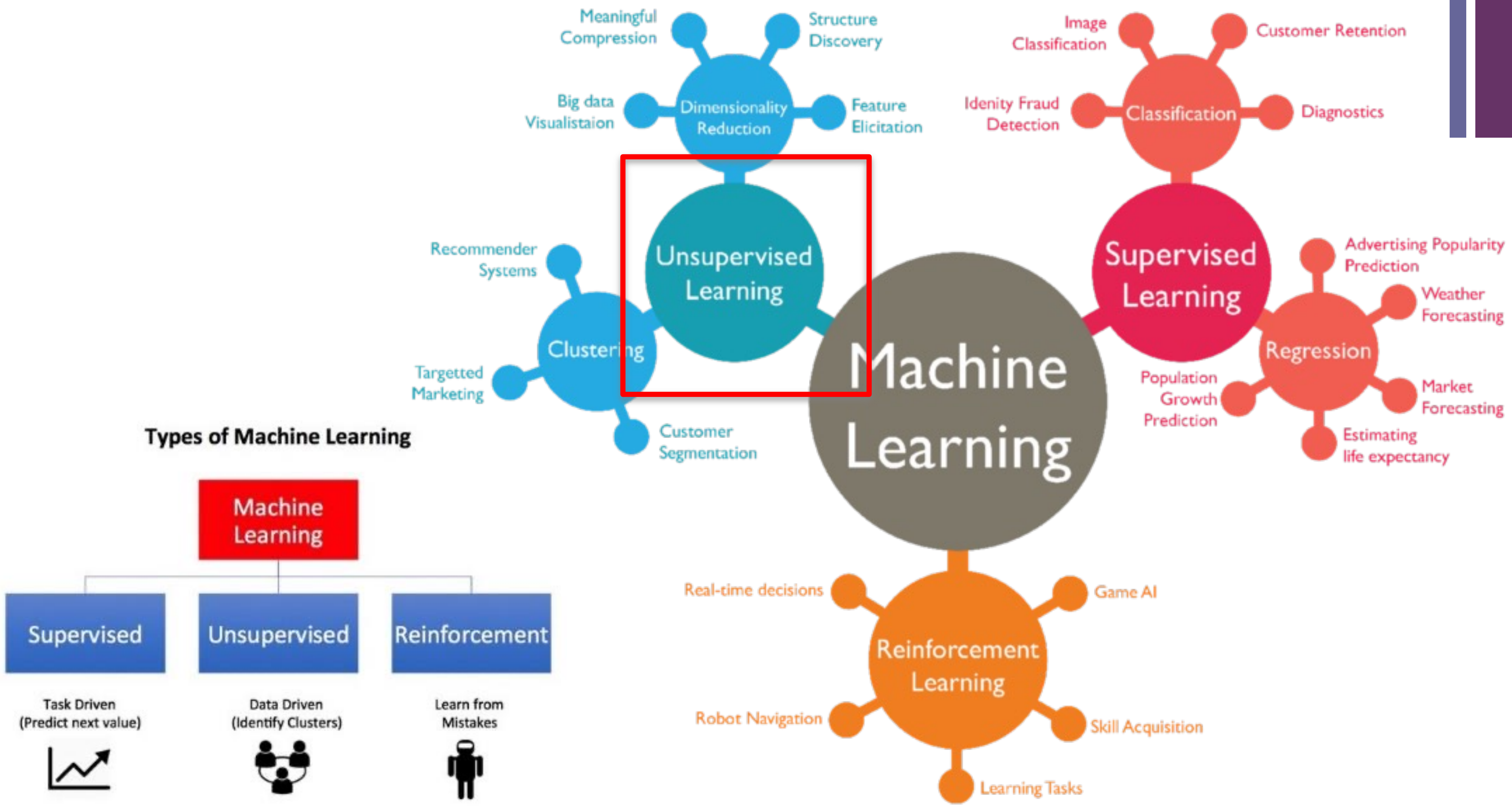
Outlines

- Clustering
- LAB 4: Clustering

+

Unsupervised Learning

+ Machine Learning (cont.)



Task2: Unsupervised learning (descriptive task)

Training Data

inputs

target

Age	Income	Gender	Province	rchase
25	25,000	Female	Bangkok	Ye
35	50,000	Female	Nontaburi	Yes
32	35,000	Male	Bangkok	

Testing Data

Age	Income	r	Province	Purchase
25	25,000	e	Bangkok	?

+ Clustering

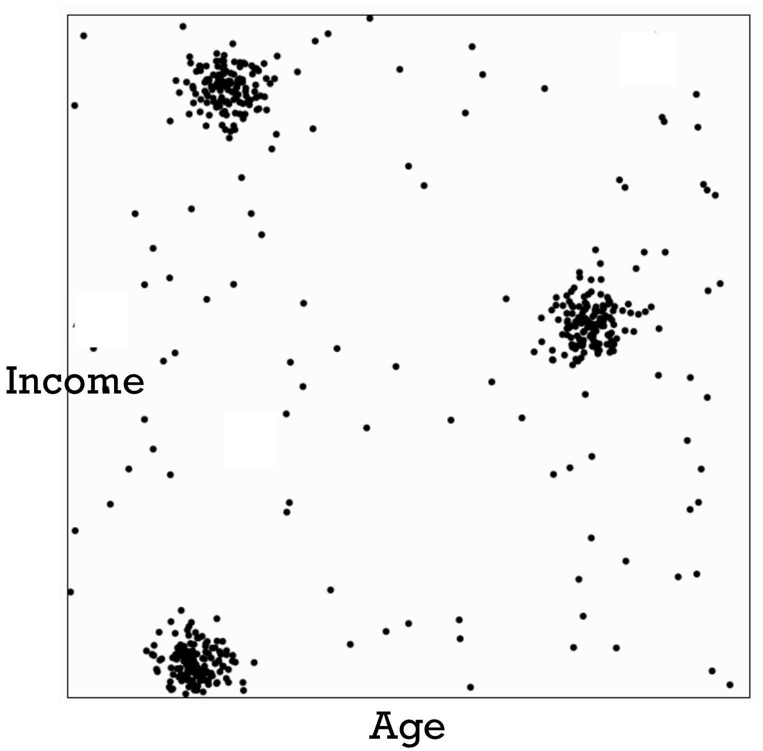
- In our class, there are many participants. Should we teach them using the same method?

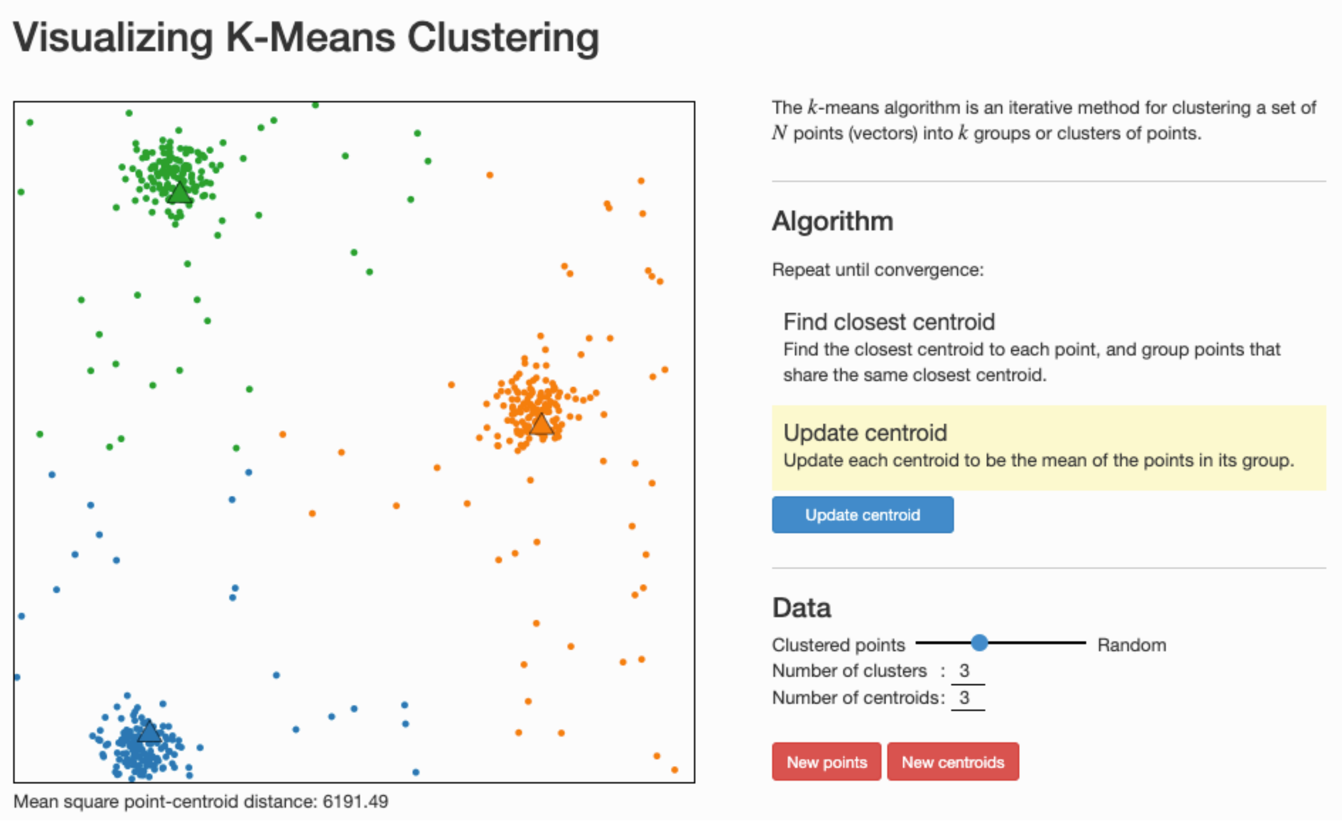
- May be not! Since they may have different learning behaviors and backgrounds.

- Inputs
  - Education field
  - Level of English communication
  - Level of computer skills
  - Age range
  - Gender

K-means Clustering

- http://web.stanford.edu/class/ee103/visualizations/kmeans/kmeans.html





Visualizing K-Means Clustering

Mean square point-centroid distance: 6191.49

The k-means algorithm is an iterative method for clustering a set of N points (vectors) into k groups or clusters of points.

Algorithm

Repeat until convergence:

Find closest centroid
Find the closest centroid to each point, and group points that share the same closest centroid.

Update centroid
Update each centroid to be the mean of the points in its group.

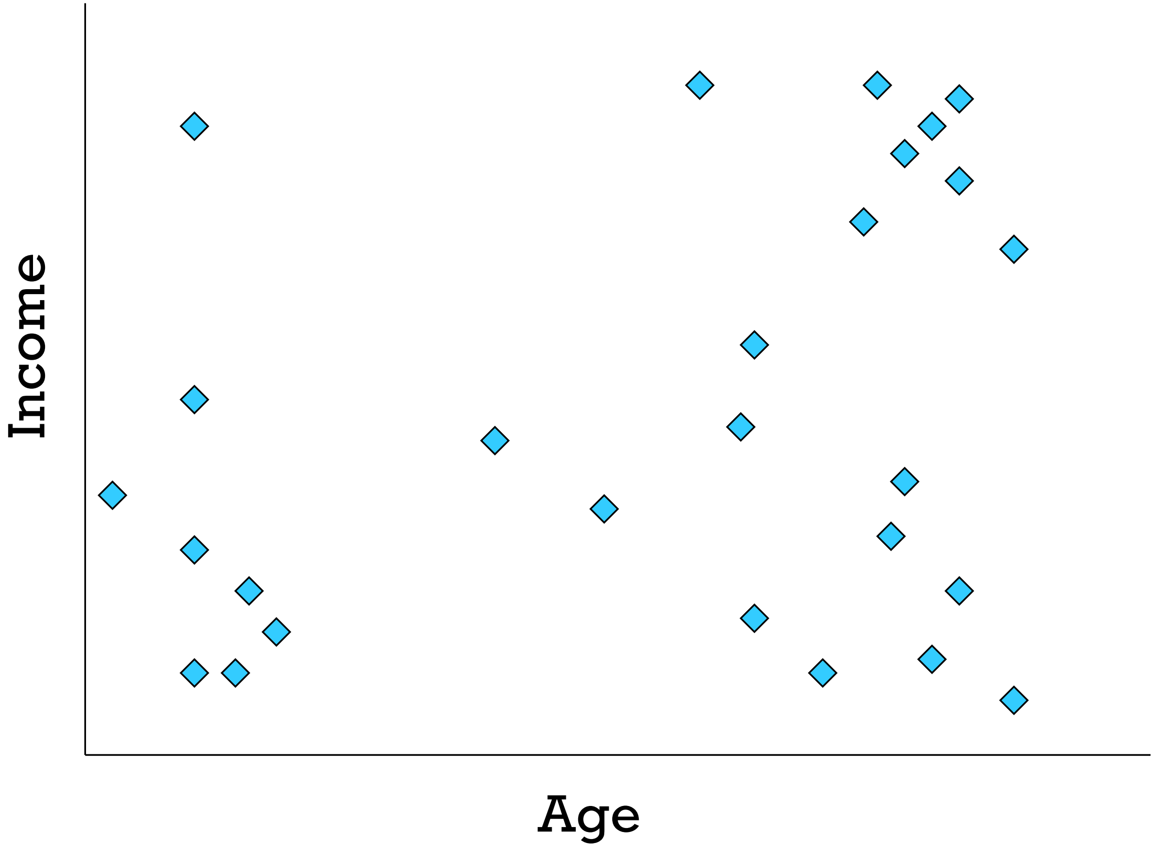
Data

Clustered points — Random

Number of clusters : 3

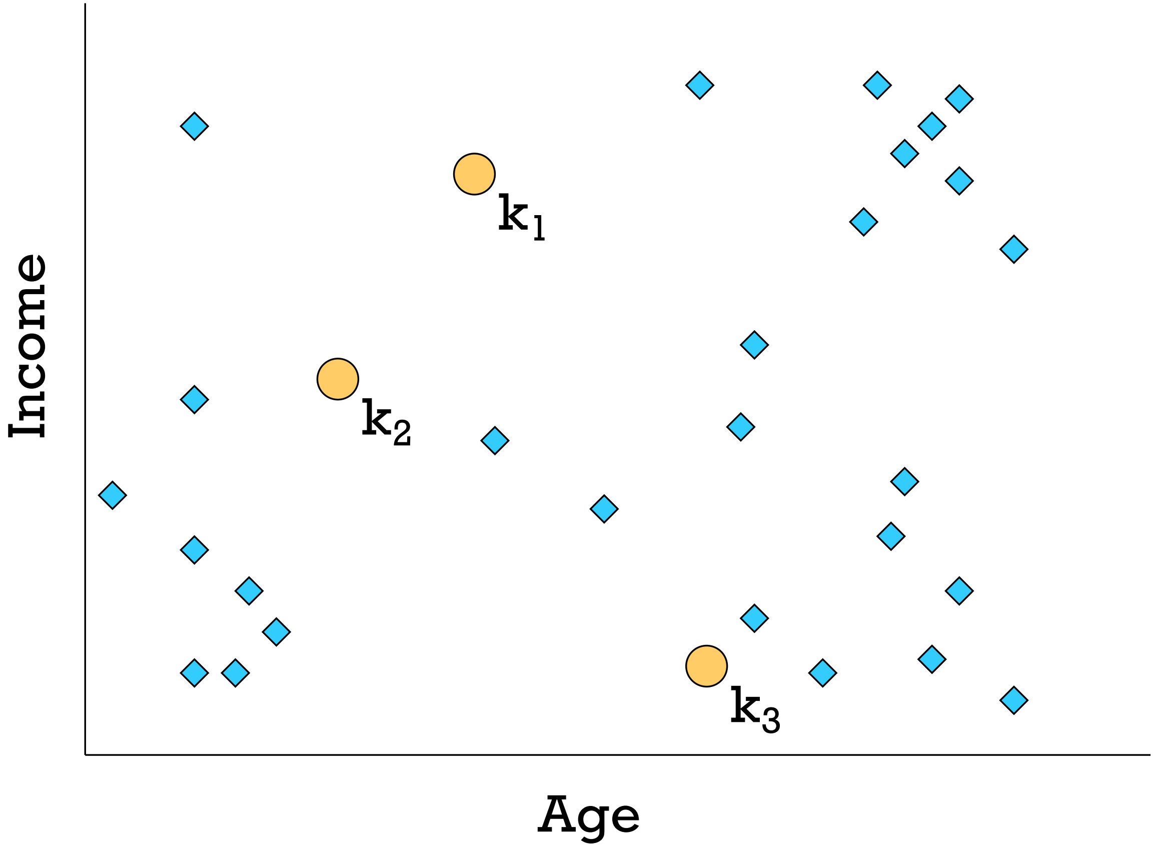
Number of centroids: 3

K-means: Step0



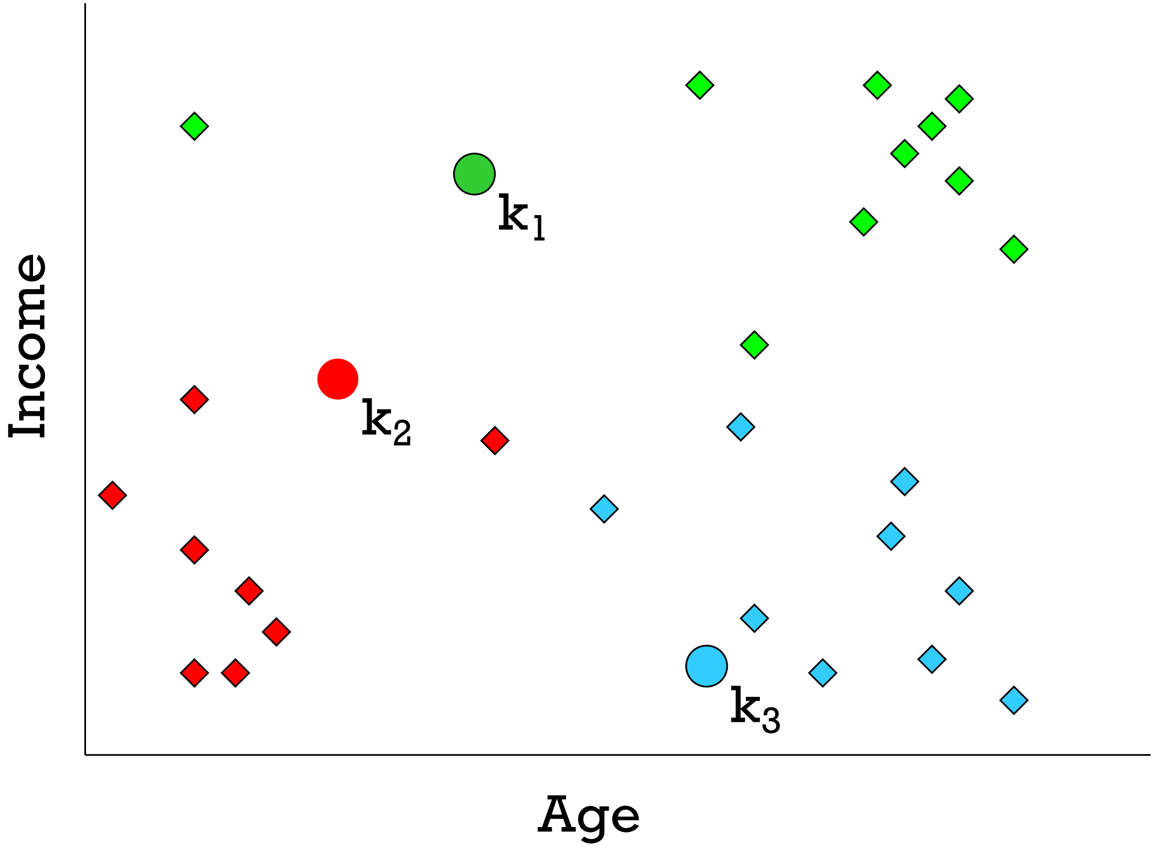
K-means: Step 1

Pick 3 initial cluster centers (randomly)



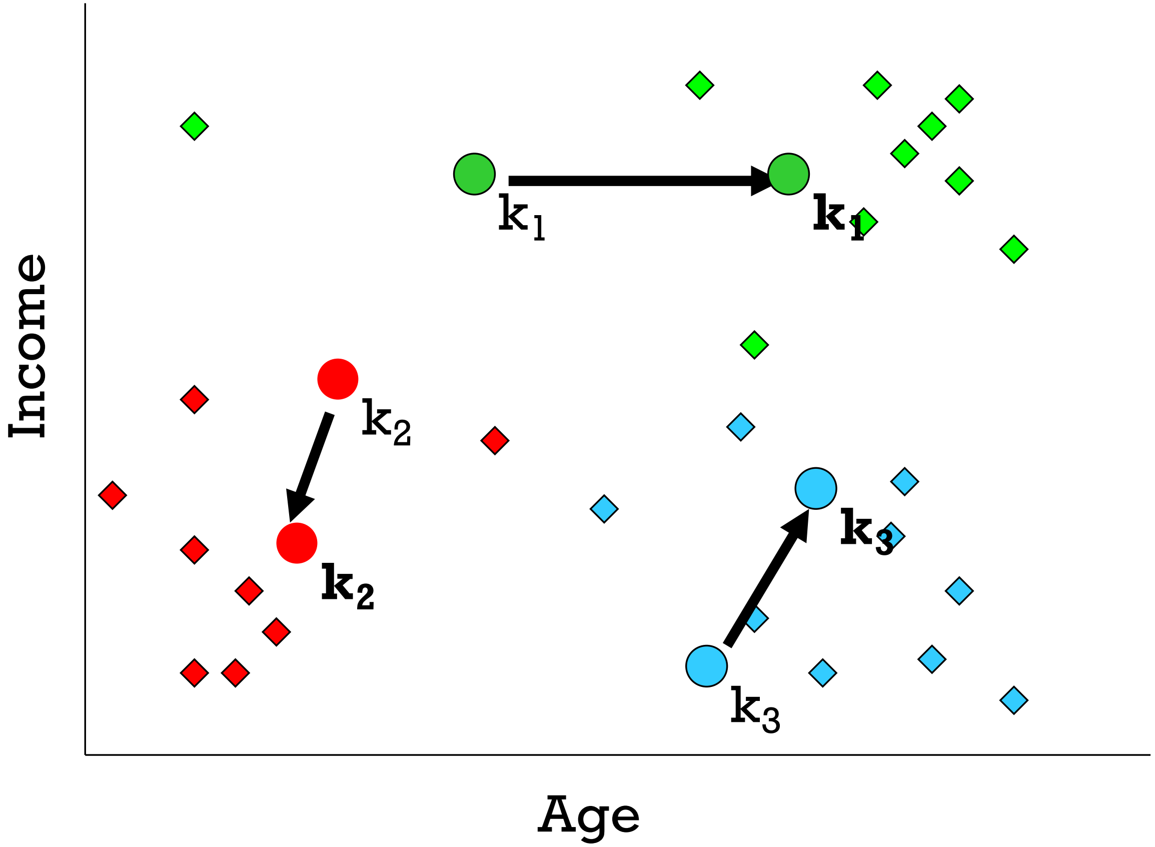
+ K-means: Step2

Assign each point to the closest cluster center



K-means: Step3

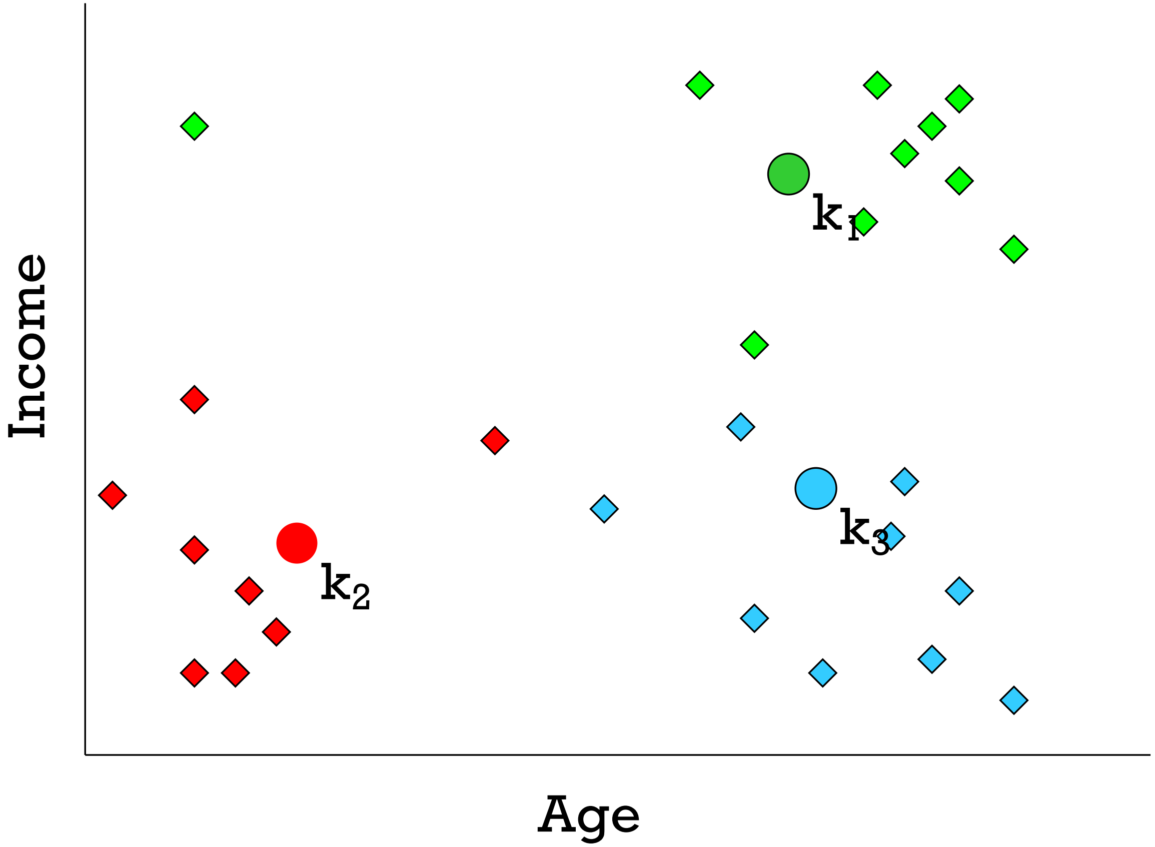
Move
each cluster
center
to the mean
of each cluster



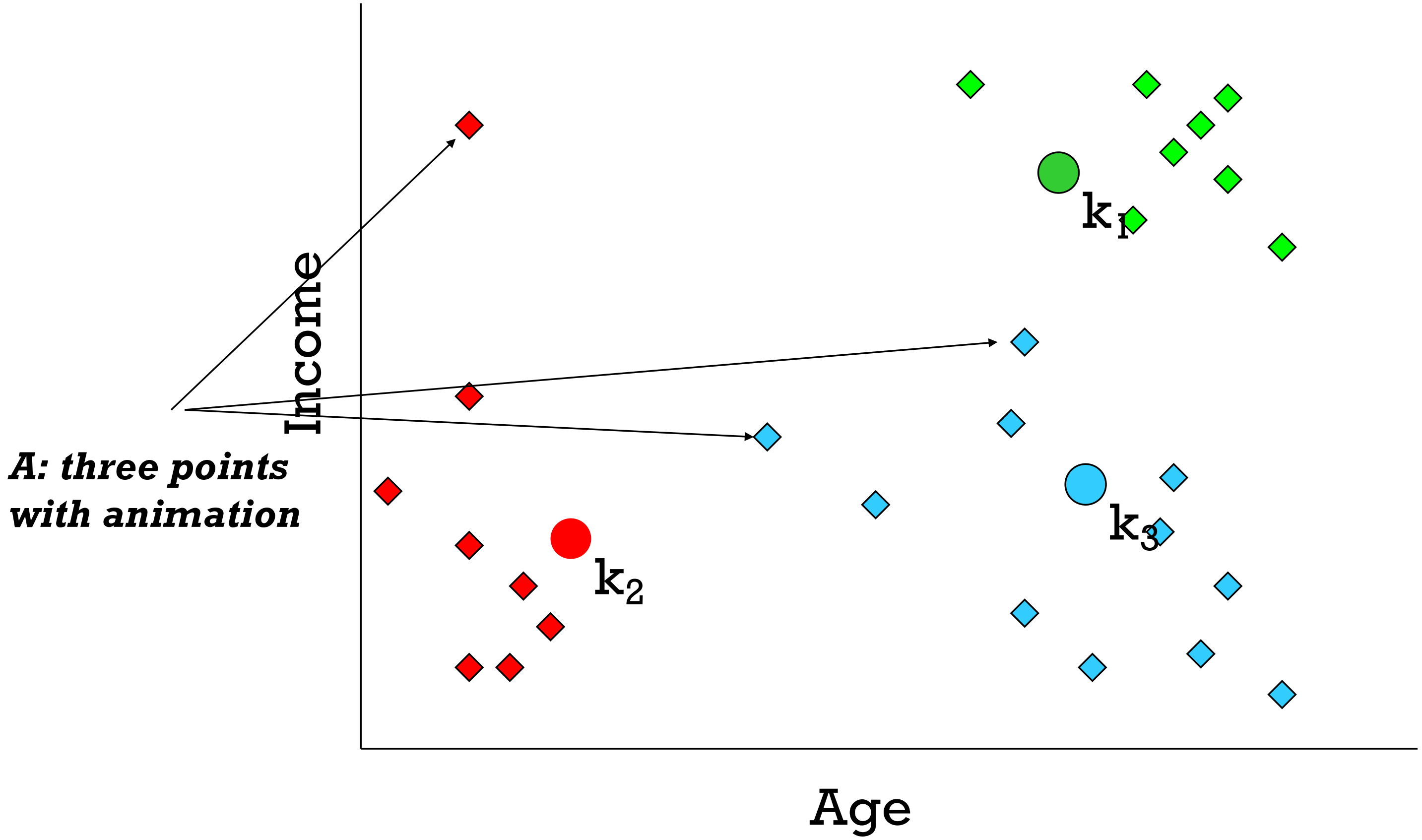
K-means: Step4

Reassign points closest to a different new cluster center

Q: Which points are reassigned?



+ K-means: Step4(a)

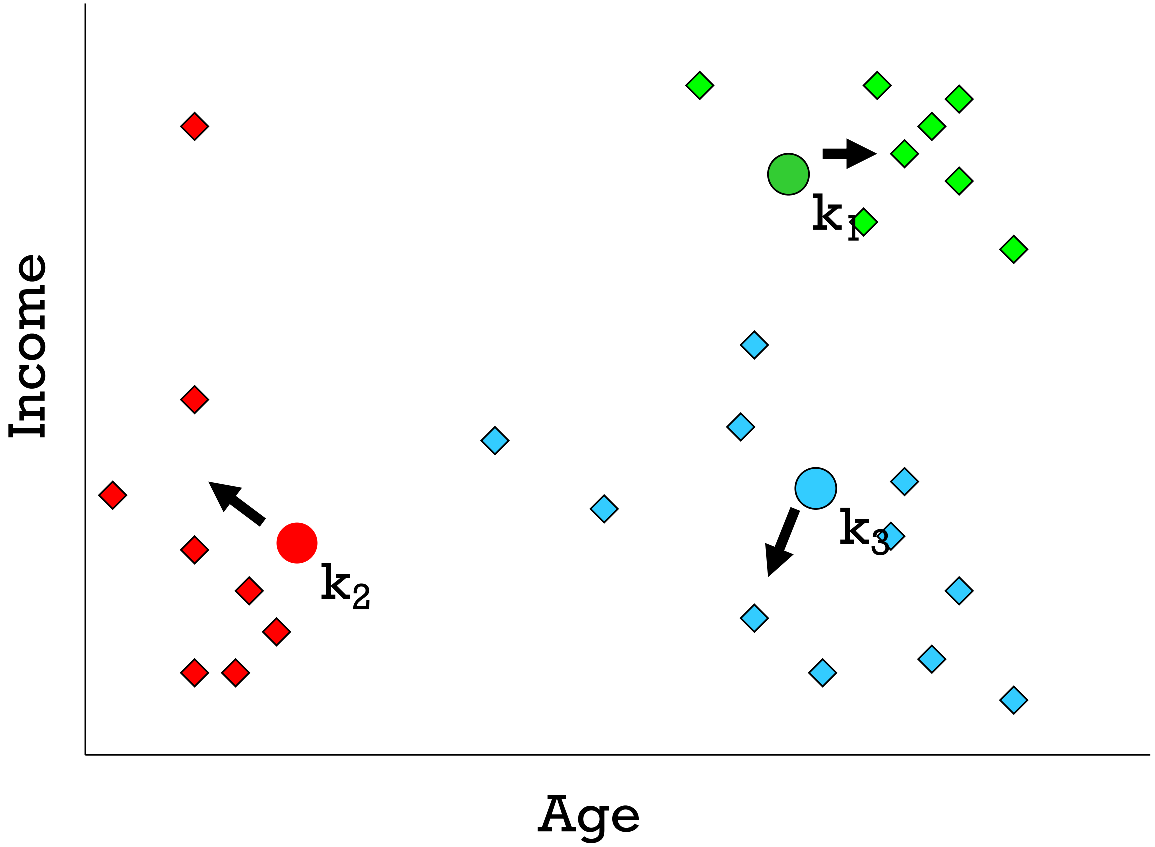


A: three points with animation

Chula Data Science

K-means: Step5

re-compute cluster means



Chula Data Science

K-means: Step5(a)

Cautions:

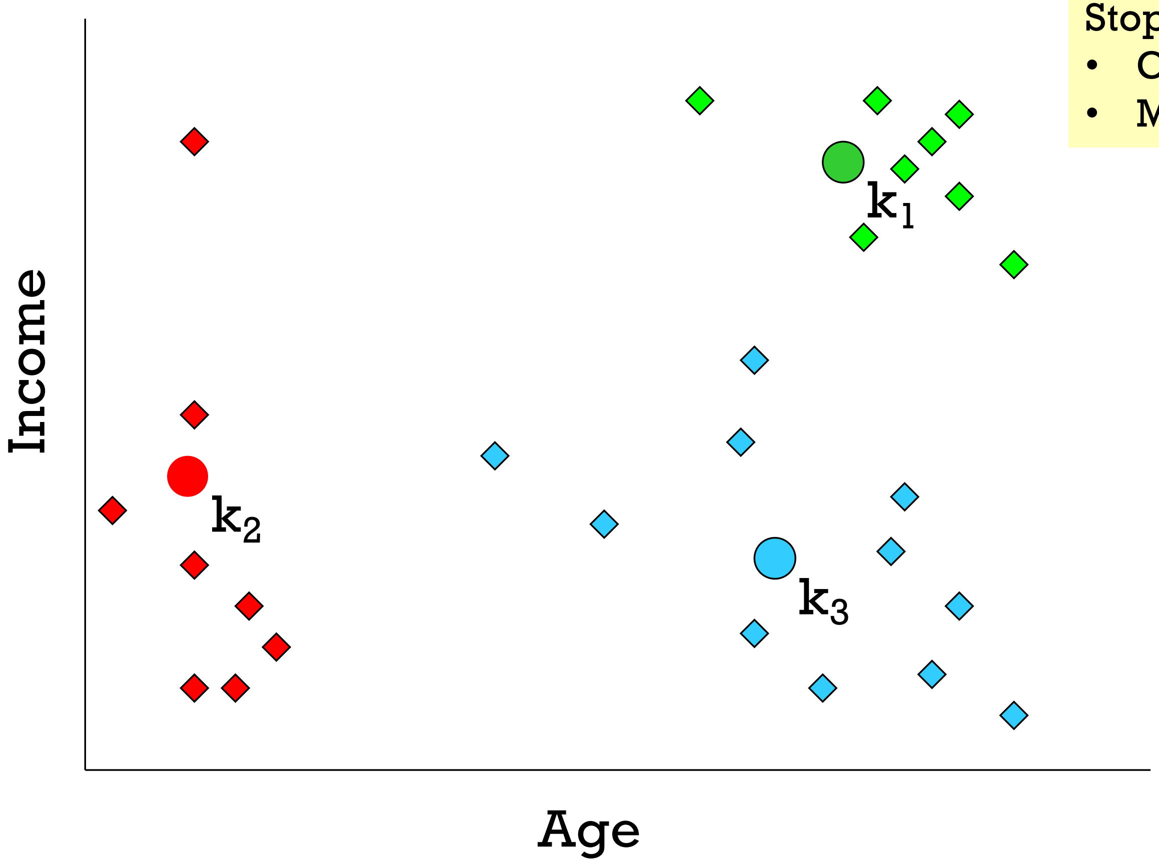
- Support only numerical variables
- Need to adjust variable range

Important Params:

- k, Distance function
- Maximum epochs

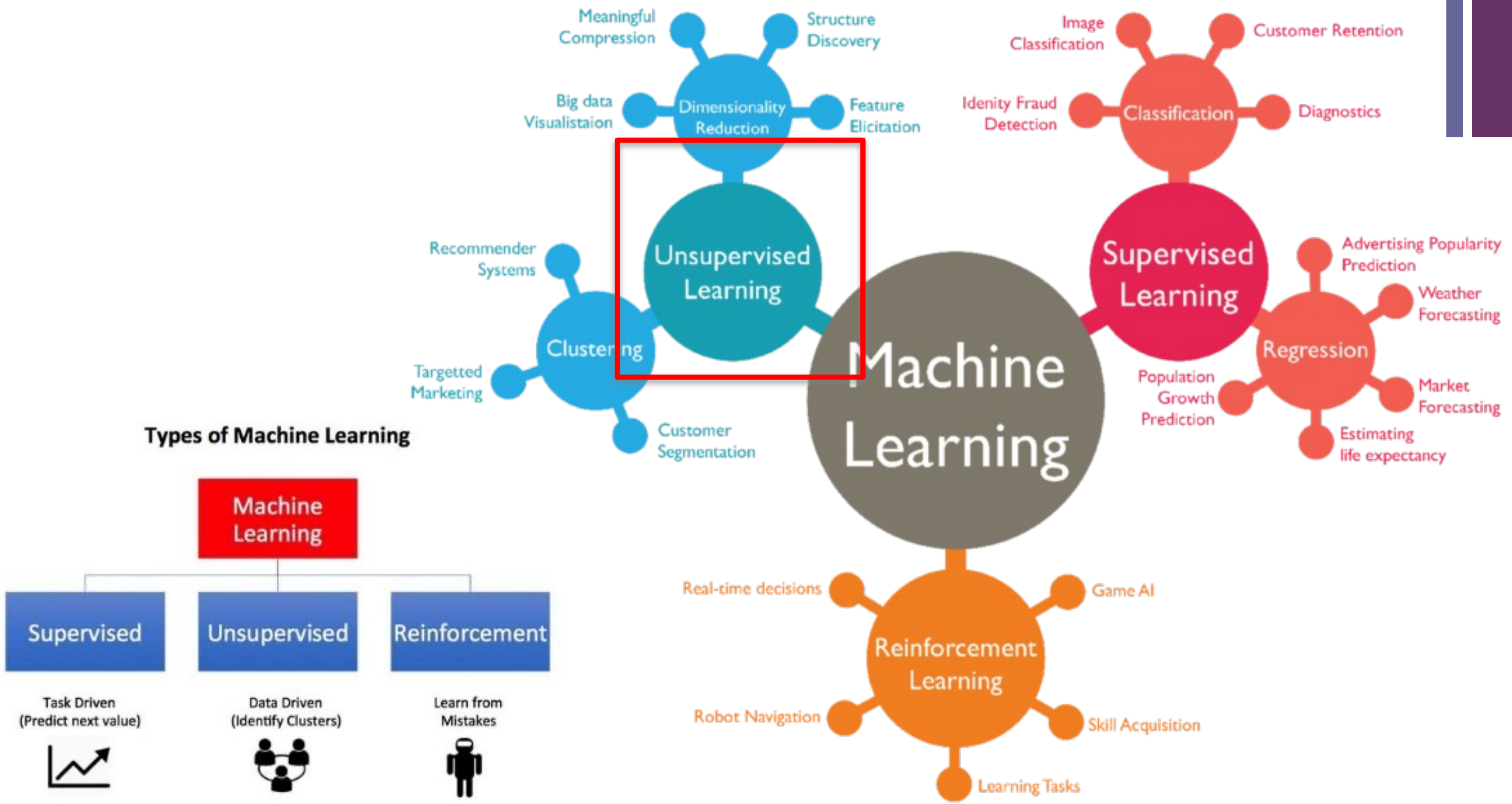
Stop

- Converge (no change)
- Maximum epochs

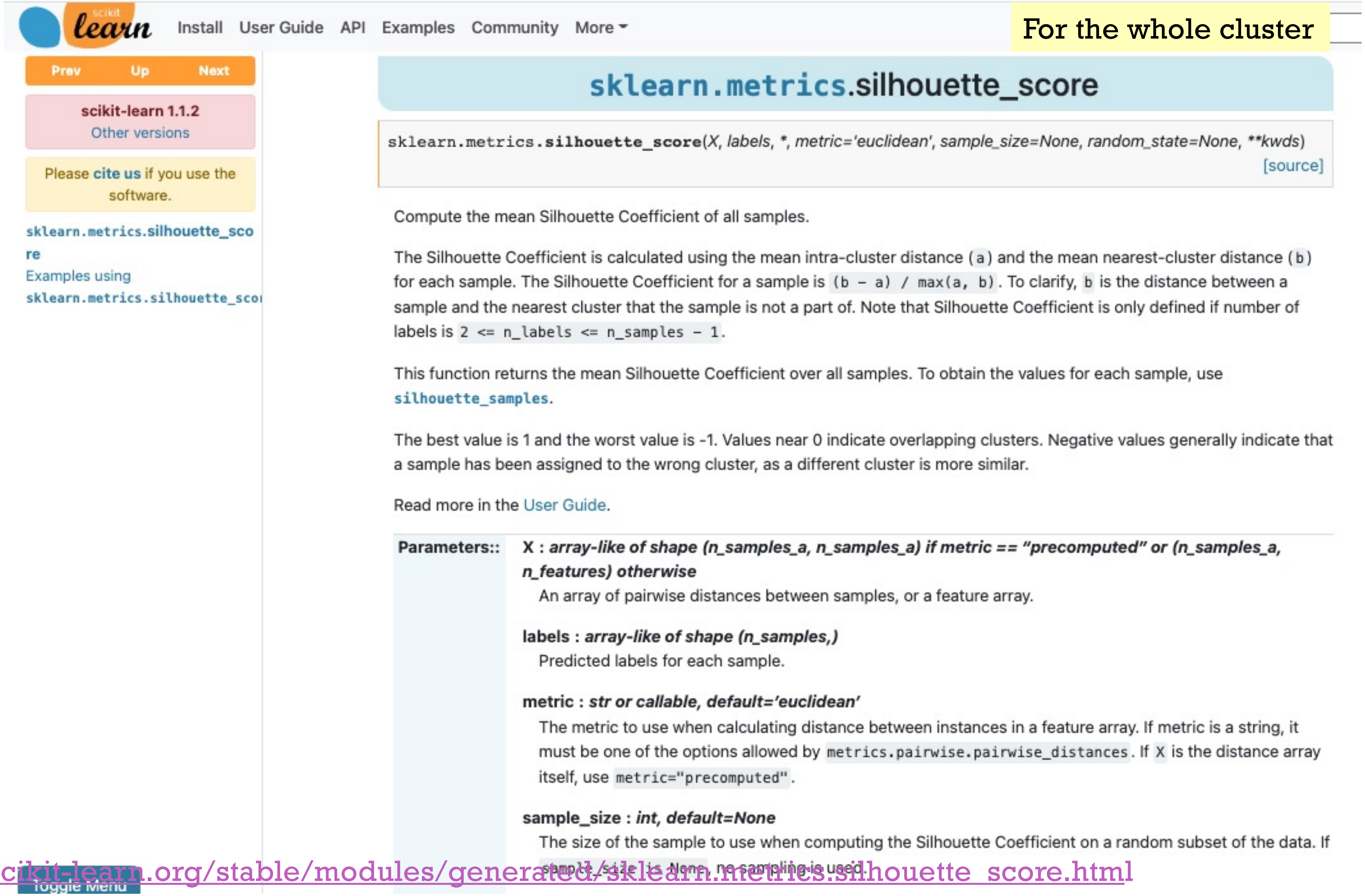


move cluster centers to cluster means

+ Machine Learning



For the whole cluster


learn
Install
User Guide
API
Examples
Community
More
Prev
Up
Next
scikit-learn 1.1.2
Other versions
Please cite us if you use the software.
sklearn.metrics.silhouette_score
Examples using sklearn.metrics.silhouette_score
Toggle Menu

sklearn.metrics.silhouette_score

`sklearn.metrics.silhouette_score(X, labels, *, metric='euclidean', sample_size=None, random_state=None, **kwds)` [source]

Compute the mean Silhouette Coefficient of all samples.

The Silhouette Coefficient is calculated using the mean intra-cluster distance (`a`) and the mean nearest-cluster distance (`b`) for each sample. The Silhouette Coefficient for a sample is `(b - a) / max(a, b)`. To clarify, `b` is the distance between a sample and the nearest cluster that the sample is not a part of. Note that Silhouette Coefficient is only defined if number of labels is `2 <= n_labels <= n_samples - 1`.

This function returns the mean Silhouette Coefficient over all samples. To obtain the values for each sample, use `silhouette_samples`.

The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar.

Read more in the User Guide.

Parameters::

X : *array-like of shape (n_samples_a, n_samples_a) if metric == "precomputed" or (n_samples_a, n_features) otherwise*
An array of pairwise distances between samples, or a feature array.

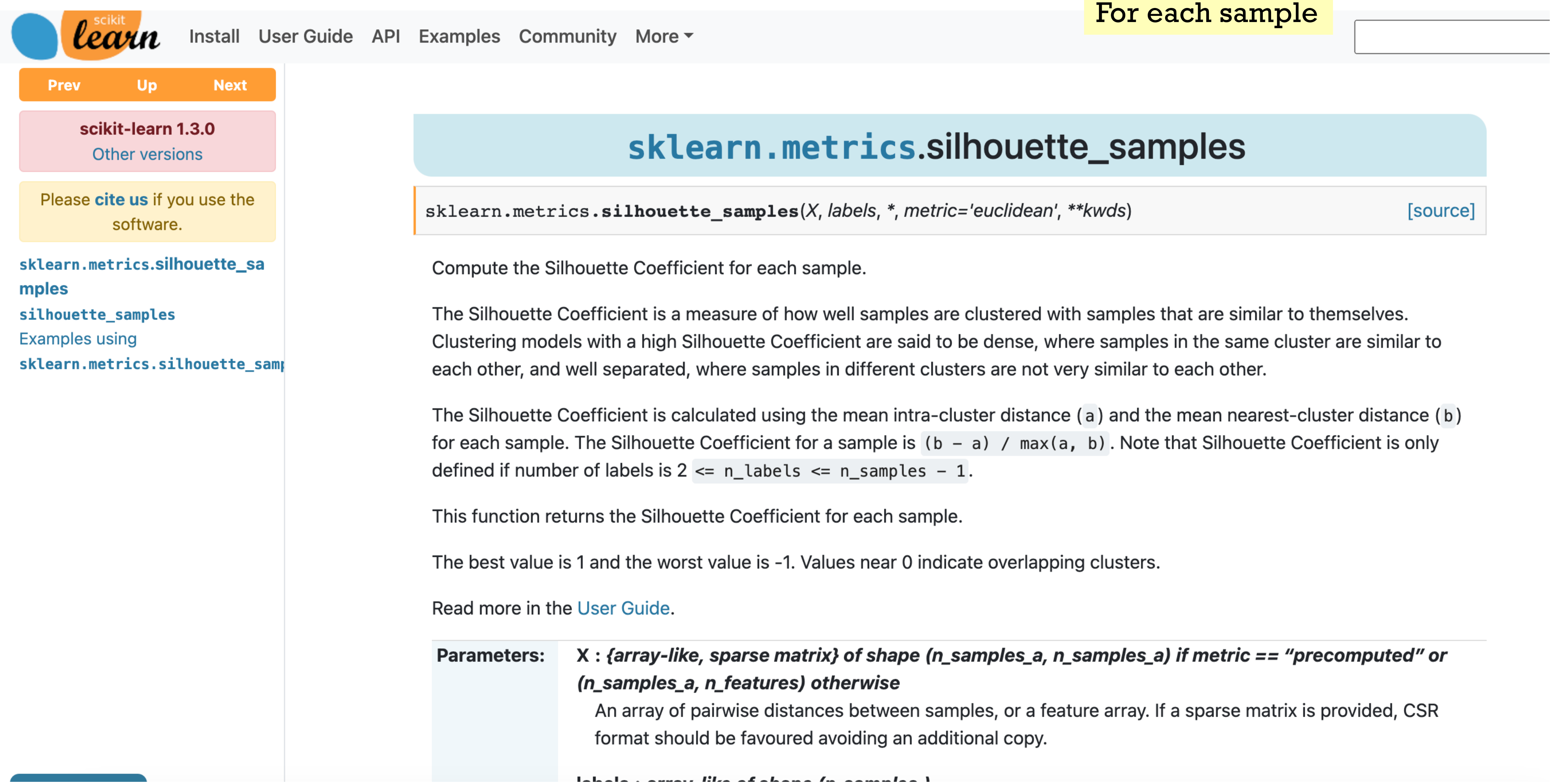
labels : *array-like of shape (n_samples,)*
Predicted labels for each sample.

metric : *str or callable, default='euclidean'*
The metric to use when calculating distance between instances in a feature array. If metric is a string, it must be one of the options allowed by `metrics.pairwise.pairwise_distances`. If `X` is the distance array itself, use `metric="precomputed"`.

sample_size : *int, default=None*
The size of the sample to use when computing the Silhouette Coefficient on a random subset of the data. If `sample_size` is `None`, no sampling is used.

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

For each sample



scikit learn
Install
User Guide
API
Examples
Community
More
Prev
Up
Next
scikit-learn 1.3.0
Other versions
Please cite us if you use the software.
sklearn.metrics.silhouette_samples
silhouette_samples
Examples using sklearn.metrics.silhouette_samp

sklearn.metrics.silhouette_samples

`sklearn.metrics.silhouette_samples`(*X*, *labels*, ***, *metric='euclidean'*, ***kwds*) [source]

Compute the Silhouette Coefficient for each sample.

The Silhouette Coefficient is a measure of how well samples are clustered with samples that are similar to themselves. Clustering models with a high Silhouette Coefficient are said to be dense, where samples in the same cluster are similar to each other, and well separated, where samples in different clusters are not very similar to each other.

The Silhouette Coefficient is calculated using the mean intra-cluster distance (`a`) and the mean nearest-cluster distance (`b`) for each sample. The Silhouette Coefficient for a sample is `(b - a) / max(a, b)`. Note that Silhouette Coefficient is only defined if number of labels is 2 `<= n_labels <= n_samples - 1`.

This function returns the Silhouette Coefficient for each sample.

The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters.

Read more in the User Guide.

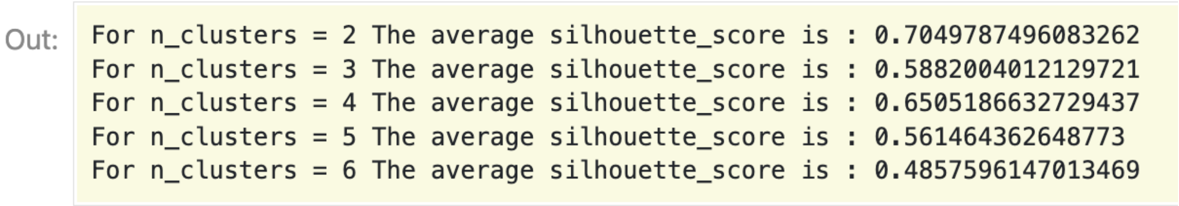
Parameters:

X : {array-like, sparse matrix} of shape (n_samples_a, n_samples_a) if metric == "precomputed" or (n_samples_a, n_features) otherwise

An array of pairwise distances between samples, or a feature array. If a sparse matrix is provided, CSR format should be favoured avoiding an additional copy.

labels : array-like of shape (n_samples,)

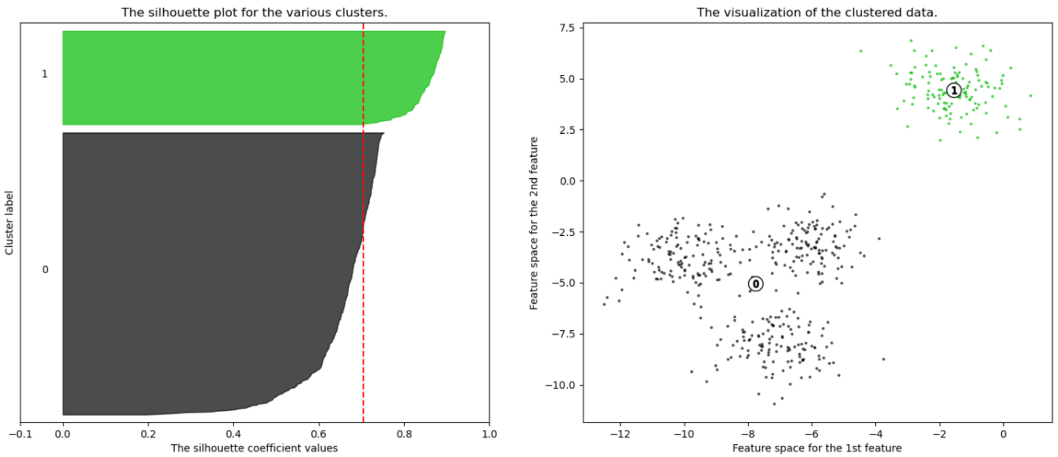
How to choose n_clusters? Chosen is 2 or 4.



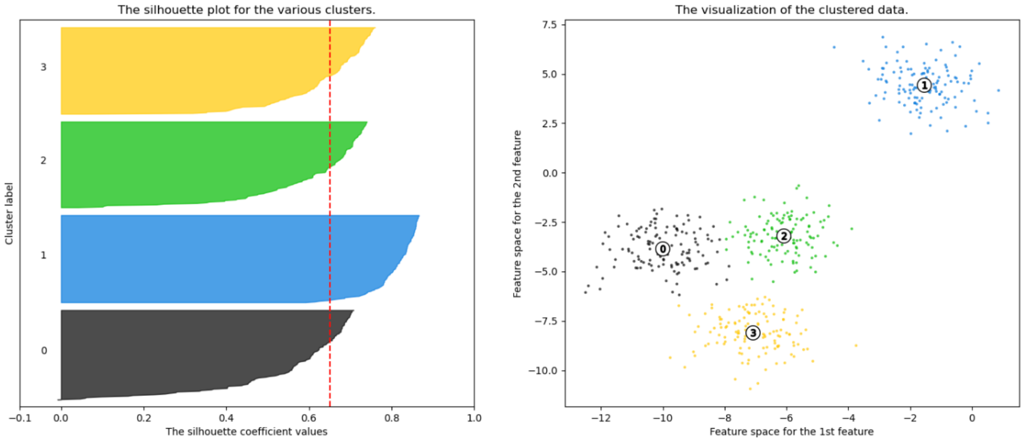
Out:

```
For n_clusters = 2 The average silhouette_score is : 0.7049787496083262
For n_clusters = 3 The average silhouette_score is : 0.5882004012129721
For n_clusters = 4 The average silhouette_score is : 0.6505186632729437
For n_clusters = 5 The average silhouette_score is : 0.561464362648773
For n_clusters = 6 The average silhouette_score is : 0.4857596147013469
```

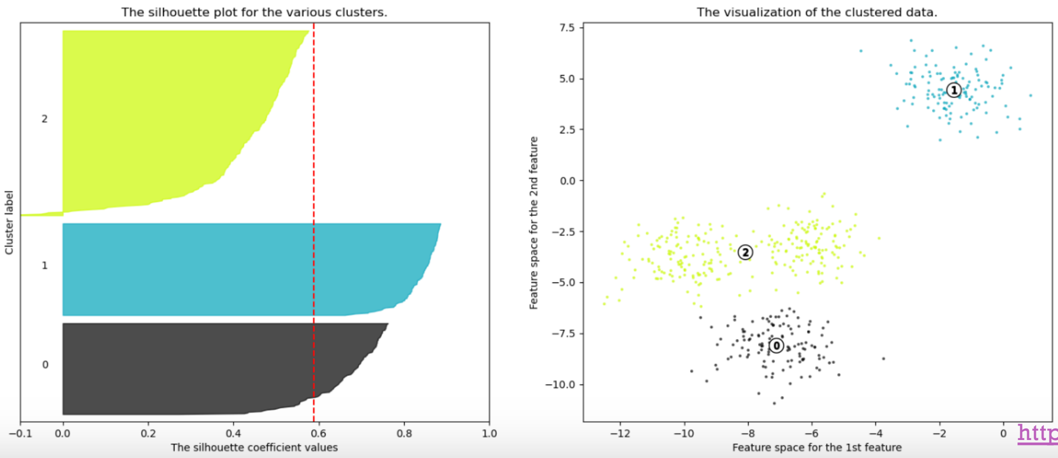
Silhouette analysis for KMeans clustering on sample data with n_clusters = 2



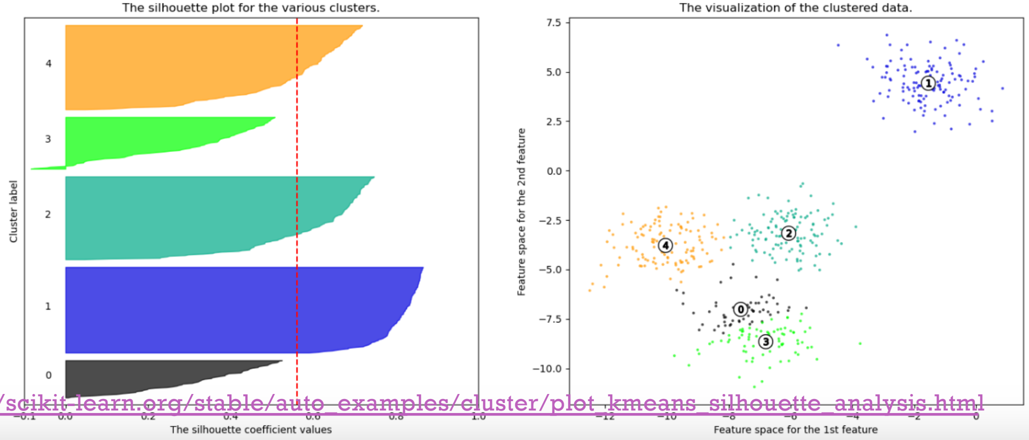
Silhouette analysis for KMeans clustering on sample data with n_clusters = 4



Silhouette analysis for KMeans clustering on sample data with n_clusters = 3



Silhouette analysis for KMeans clustering on sample data with n_clusters = 5



https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

LAB 4: Clustering

LAB 4: Clustering

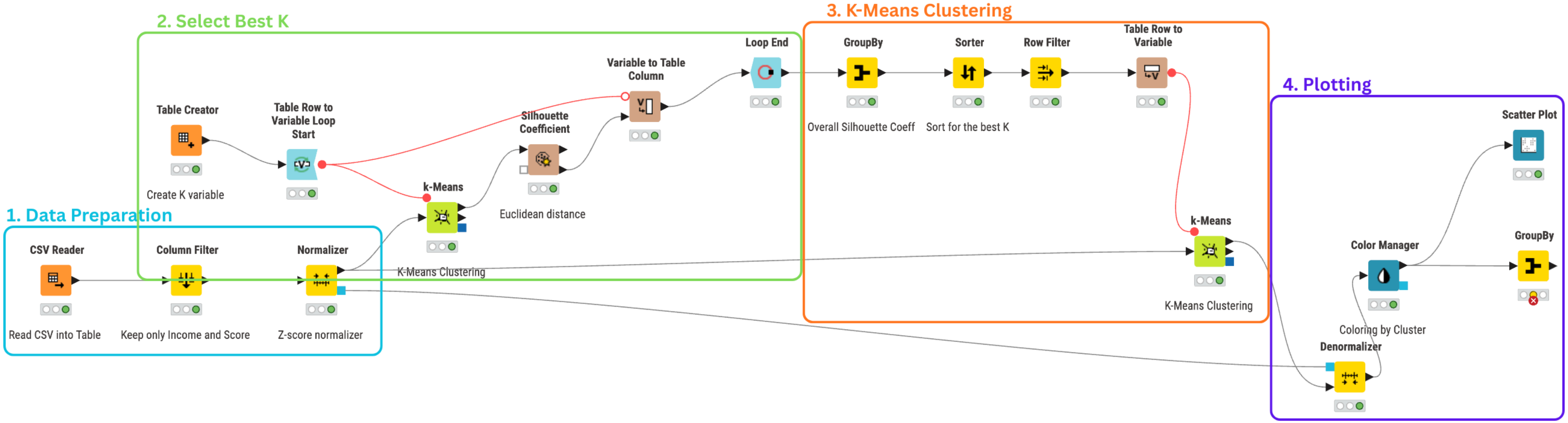
Data Description

- The Mall Customer dataset contains 200 customers. Each row is one subject. The goal is to predict the Spending Class (High or Low Spender) of the customer.
- Target Variable: High Spender: "yes" if spending score >= 60; else "no"

Input:

- CutomerID
- Gender: Male/Female
- Age: (years)
- Annual Income: (k$)
- Spending Score: Derived Metric (0-100)

LAB 4: Clustering



+

Thank you & any questions