Unsupervised Learning

Agenda

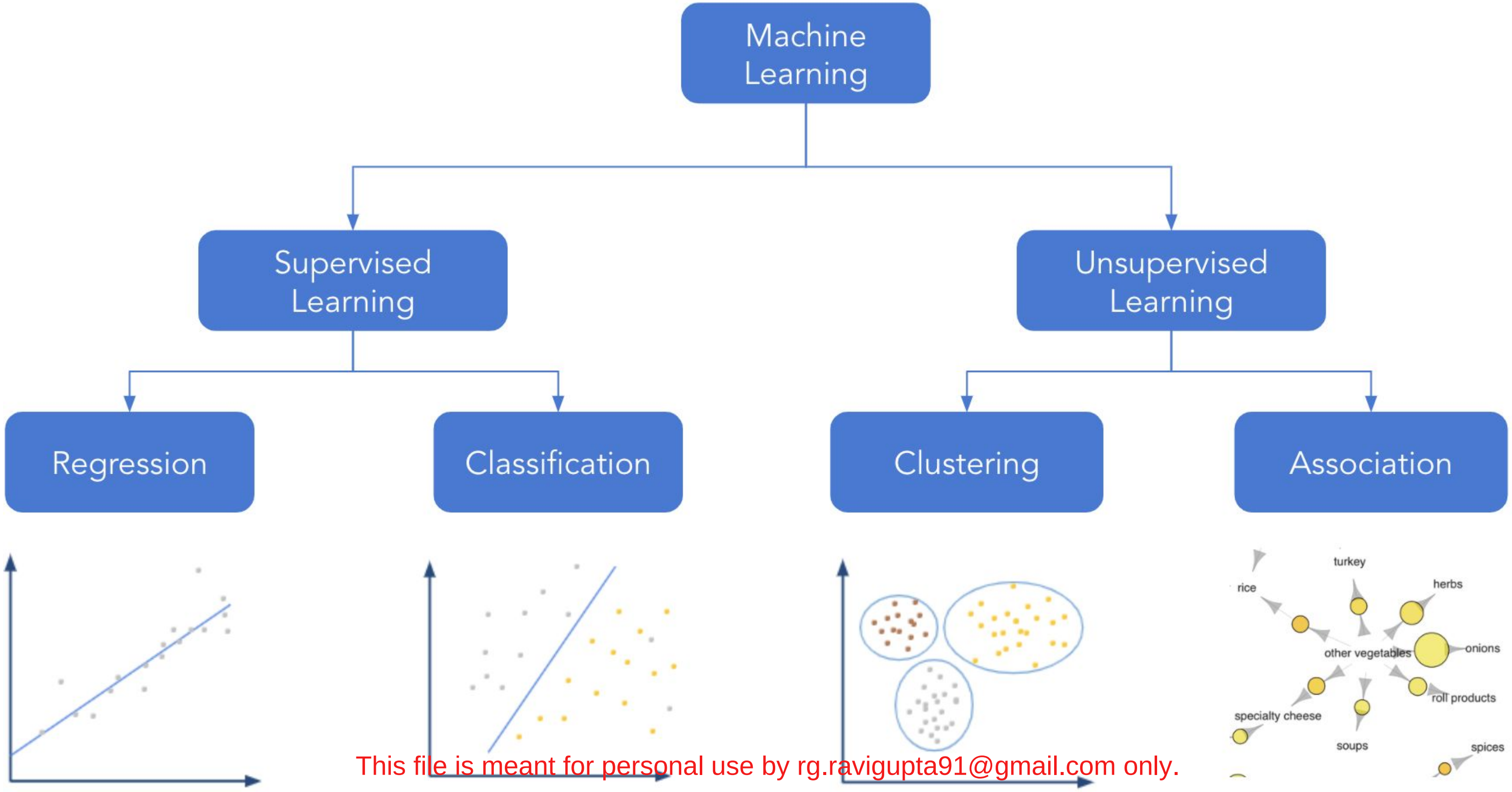
- Machine Learning
  - Supervised Learning
  - Unsupervised Learning
- Clustering
- Visiting Basics
  - Proximity Measures
  - Distance Measures

Agenda

- K-means Algorithm
  - Cluster Formation
  - Optimal Value of K
    - Elbow Plot
    - Silhouette Method

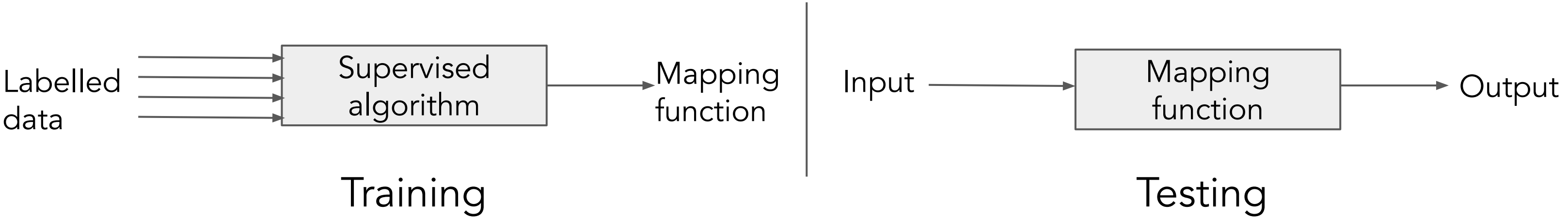
Machine Learning

Machine learning



Supervised learning

Supervised learning aims at finding a model that maps the output (target) variable to the input (predictor) variables.



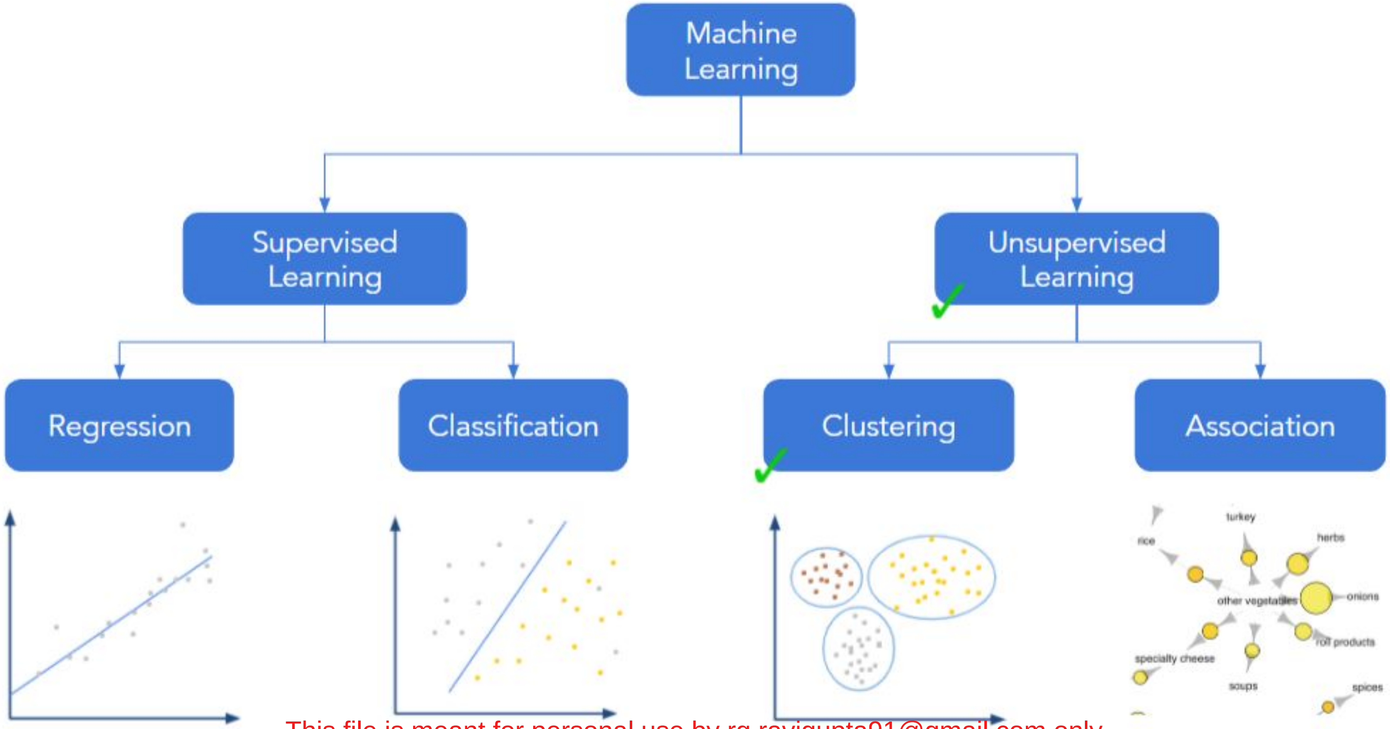
Example: Detection of phishing emails based on certain phrases like 'You have won million'. More such phrases are prespecified while training the model. So if a new email also contains a similar phrase such emails can directly be tagged as spam.

Unsupervised learning

- Unsupervised learning aims to learn more about the given data
- The data used for unsupervised learning has no labels, i.e. there is no desired outcome or correct answer given

Example: Consider a dataset with information about flowers. We know the data has
records of flowers and their different characteristics.
Using unsupervised learning, we group flowers with similar characteristics and try to find if
they belong to a particular species.

In this session, we shall cover clustering



Business example: group the data

Consider the data of flower's petal length and petal width in millimeters for different flowers.

Petal Length	5.5	17	16.5	4.8	11.5	5.8	10	4.6	15.5	13	5.1	12	16.2	13
Petal Width	7.5	15	15	8.4	10	8.6	9	8.8	14	12	9	11	15.7	11

Can we find the data that belongs to the same kind of flower?

Why clustering?

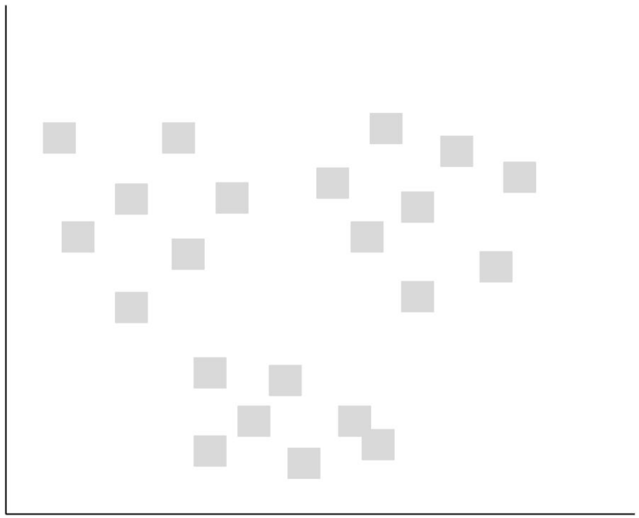
Yes, it is possible to find the data that belongs to the same kind of flower.

However, it is not possible to name the type of the flower, since no information on its ‘label’ is available.

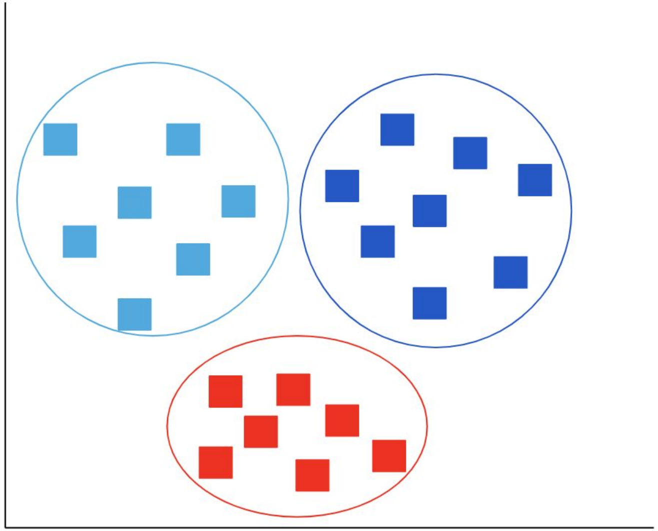
Clustering

Clustering

Clustering is a technique in which data is grouped based on the similarities in them



Raw data



Clustered data

Clustering

- Clustering partitions the data into natural groups such that
  - Points in the cluster are as similar as possible
  - Points across the clusters are as dissimilar as possible

Types of clustering

PES UNIVERSITY

- Density based: Clusters are formed based on the density of observations (DBSCAN)
- Hierarchical based: Hierarchy of clusters are formed based on the distances between the observations (Hierarchical clustering)
- Graph based: Clusters are formed either by dividing a set of graphs or dividing the nodes of a graph (K-spanning tree)
- Partition based: Observations are partitioned into predetermined number of clusters based on distance (K-means clustering)
- Model based: Assumes that the data is a mixture of distributions and tries to fit a model such that each distribution represents a cluster (Gaussian Mixture Model)

In this course...

- Density based: Clusters are formed based on the density of observations (DBSCAN)
- Hierarchical based: Hierarchy of clusters are formed based on the distances between the observations (Hierarchical clustering)
- Graph based: Clusters are formed either by dividing a set of graphs or dividing the nodes of a graph (K-spanning tree)
- Partition based: Observations are partitioned into predetermined number of clusters based on distance (K-means clustering)
- Model based: Assumes that the data is a mixture of distributions and tries to fit a model such that each distribution represents a cluster (Gaussian Mixture Model)

Clustering approach

Each point is assigned to only one cluster

eg. K-means clustering

The probability of each cluster for each of the points is obtained

eg. Gaussian Mixture

PLEASE!! NOTE

Classification and clustering

- In the classification problem, the target variable is known and is categorical
- A model is trained on this information and new data is classified accordingly
- In clustering, groups of data are formed based on similarity in observations

Visiting Basics

Proximity measures

- The proximity measures find the distance between two instances
- Proximity measures include
    - Similarity measures
    - Dissimilarity measures
- Depending upon the data types, we choose the proximity measure

Similarity and dissimilarity measures

A similarity measure for two objects return the value between 0 and 1. For the completely similar objects the similarity measure is 1.

$$
\begin{array}{c c}
 & \begin{array}{cccc} x_1 & x_2 & & x_n \end{array} \\
\begin{array}{c} x_1 \\ x_2 \\ \\ x_n \end{array} &
\begin{pmatrix}
1 & & & \\
d_{12} & 1 & & \\
 & & \ddots & \\
d_{1n} & d_{2n} & \dots & 1
\end{pmatrix}
\end{array}
$$

A dissimilarity measure for two objects return the value between 0 and 1. For the completely similar objects the dissimilarity measure is 0.

$$
\begin{array}{c c}
 & \begin{array}{cccc} x_1 & x_2 & & x_n \end{array} \\
\begin{array}{c} x_1 \\ x_2 \\ \\ x \end{array} &
\begin{pmatrix}
0 & & & \\
d_{12} & 0 & & \\
 & & \ddots & \\
d_{1n} & d_{2n} & \dots & 0
\end{pmatrix}
\end{array}
$$

Distance measures for numeric data

The x_i and y_i are the values taken by variables X and Y respectively

Distance Measure	Formula
Euclidean measure	$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
Squared Euclidean distance	$\sum_{i=1}^{n}(x_i - y_i)^2$
Manhattan distance	$\sum_{i=1}^{n}\lvert x_i - y_i \rvert$
Minkowski distance	$\sqrt[p]{\sum_{i=1}^{n}\lvert x_i - y_i \rvert^p}$
Chebyshev's distance	$\max_{i=1}^{n}\lvert x_i - y_i \rvert$

K - means Algorithm

K - means Algorithm

Specifies the number of clusters

Specifies average (centroid) of a cluster

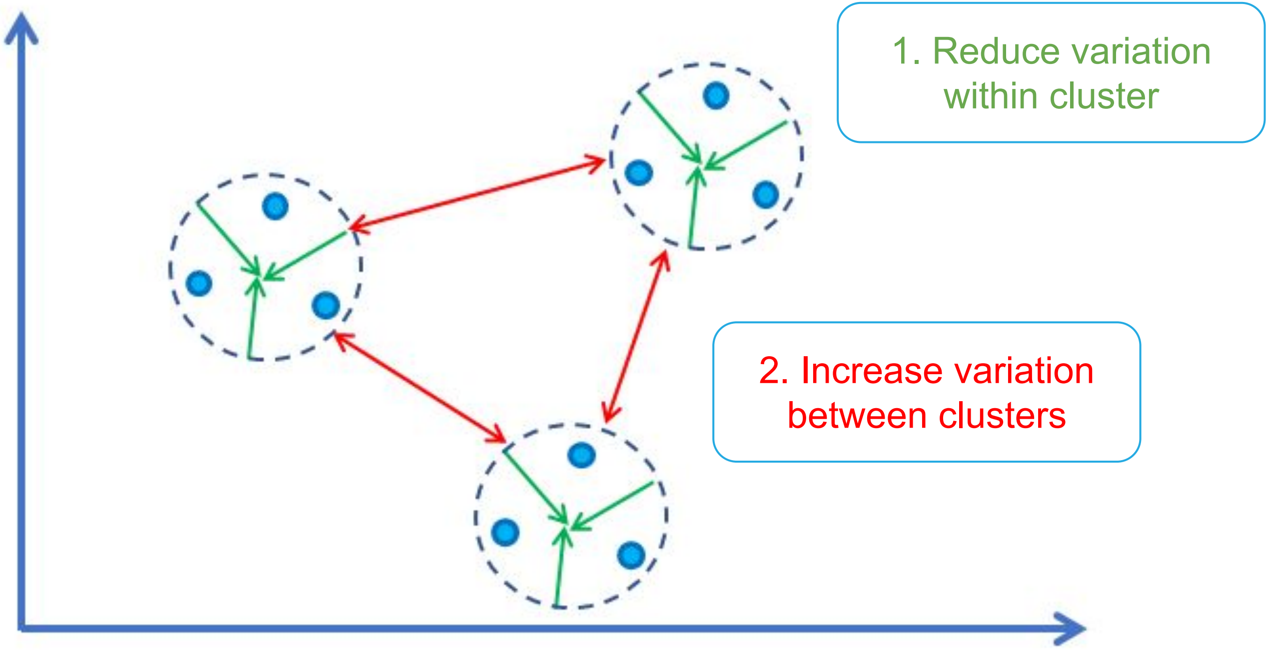
K-means algorithm

- Used when data is numeric
- Recursive technique
- Can not train a model
- Based on proximity measures

K-means algorithm

- Greedy algorithm
- Minimizes the squared error of points in its cluster
- Non-deterministic algorithm

Objective of clustering



K-means algorithm - procedure

Choose a distance measure and value of K

↓

Randomly choose K points as cluster centroids

↓

Assign the nearest cluster centroid to each data point

↓

Compute the means of these clusters

↓

Reassign the cluster based on the means

↓

Repeat the steps 4 and 5, until the cluster means do not change

PLEASE!! NOTE

Python’s sklearn library provides the KMeans() to cluster the data in pre-specified number of clusters.

```
# import the function
from sklearn.cluster import KMeans

# build a K-Means model for a specific value of K
# 'random_state' preserves the cluster labels over multiple code runs
model = KMeans(n_clusters= K, random_state = 10)

# fit and predict the cluster labels
model.fit_predict(data)
```

Data scaling

Consider a data with 3 features, of which 2 features have a small range (say between 0 to 25), and the third feature ranges from -100 to 2000. The clustering would be majorly based on the feature with a high range. Since it's contribution to the distance measure would be high with very little or no effect of the other variables; thus, we scale the data such that each feature will have equal weight.

Cluster Formation

Create the clusters

Consider the data of flowers petal length and petal width in millimeters.

Petal Length	5.5	17	16.5	4.8	11.5	5.8	10	4.6	15.5	13	5.1	12	16.2	13
Petal Width	7.5	15	15	8.4	10	8.6	9	8.8	14	12	9	11	15.7	11

Can we cluster the flower petals into distinct groups?

Example:

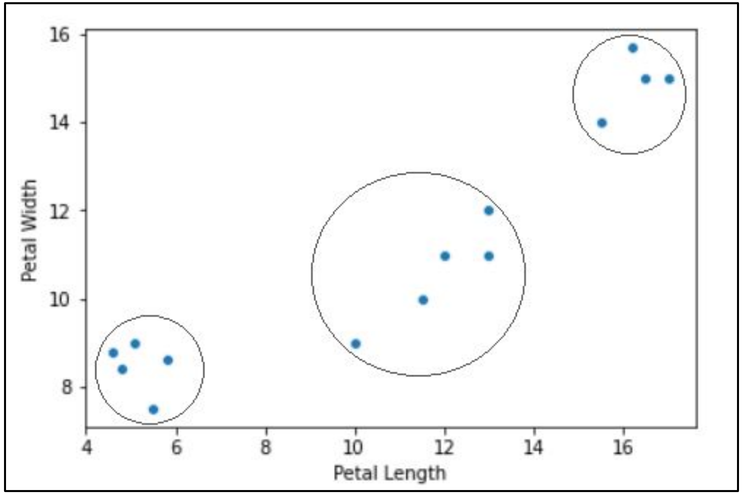
Create a dataframe of the given data.

Petal Length	Petal Width
5.5	7.5
17.0	15.0
16.5	15.0
4.8	8.4
11.5	10.0
5.8	8.6
10.0	9.0
4.6	8.8
15.5	14.0
13.0	12.0
5.1	9.0
12.0	11.0
16.2	15.7
13.0	11.0

Example: step 1

Let us first plot the data.

From the plot, we can see that the dataset is divided into 3 groups.



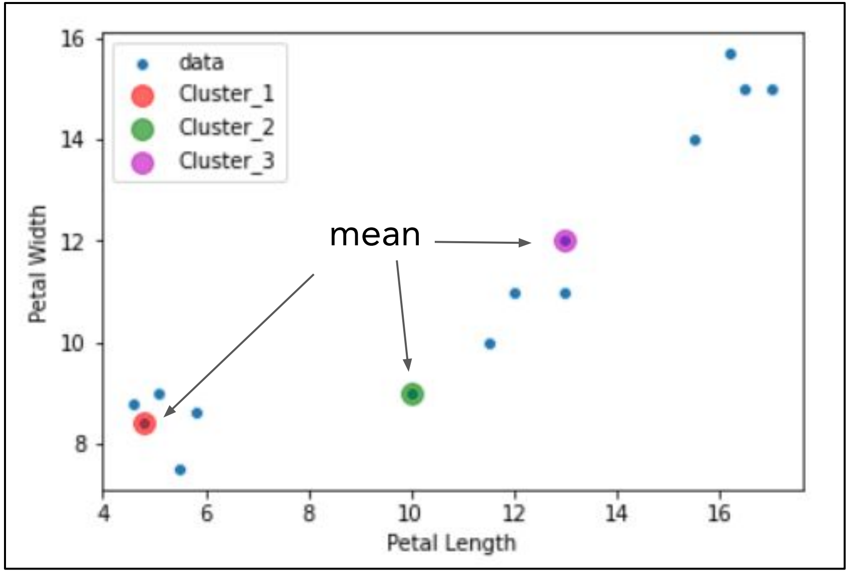
Now, use K-means clustering to check whether the algorithm can form such three clusters (K = 3) from the given data. Consider the Euclidean distance as a distance

Example: step 2

Initial assignment:

Randomly choose 3 points as cluster centroids.
Consider the below observations as initial centroids.

Petal Length	Petal Width
4.8	8.4
10.0	9.0
13.0	12.0



did you know?

Initial centroid assignment methods

- Two methods for initializing cluster centroids: Forgy, Random Partition
- Forgy method assigns K random observations as cluster centroids for K clusters
- In Random Partition method, a cluster is randomly assigned to each data point, and the mean of data in each cluster is considered as cluster centroid

Example: step 3

Calculate the Euclidean distance of each point from cluster centroids.

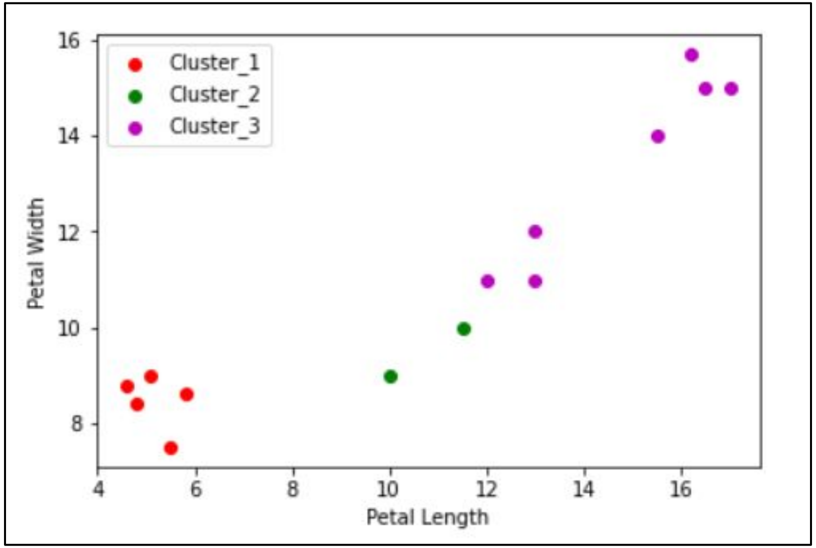
The table shows that the 1st point is closest to the first cluster, the 2nd point is closest to the third cluster and so on.

Euclidean Distance		
Dist_1	Dist_2	Dist_3
1.140175425	4.74341649	8.746427842
13.87083271	9.219544457	5
13.43316791	8.845903006	4.609772229
0	5.234500931	8.955445271
6.88839604	1.802775638	2.5
1.019803903	4.219004622	7.962411695
5.234500931	0	4.242640687
0.447213595	5.403702434	8.988882022
12.07683733	7.433034374	3.201562119
8.955445271	4.242640687	0
0.670820393	4.9	8.450443775
7.655063684	2.828427125	1.414213562
13.53698637	9.128526716	4.891829923
8.602325267	3.605551275	1

Example: step 3

Assign the data to the nearest cluster.

The plot shows that 1st cluster contains 5 points, the 2nd cluster contains 2 points, and the 3rd cluster contains 7 points.

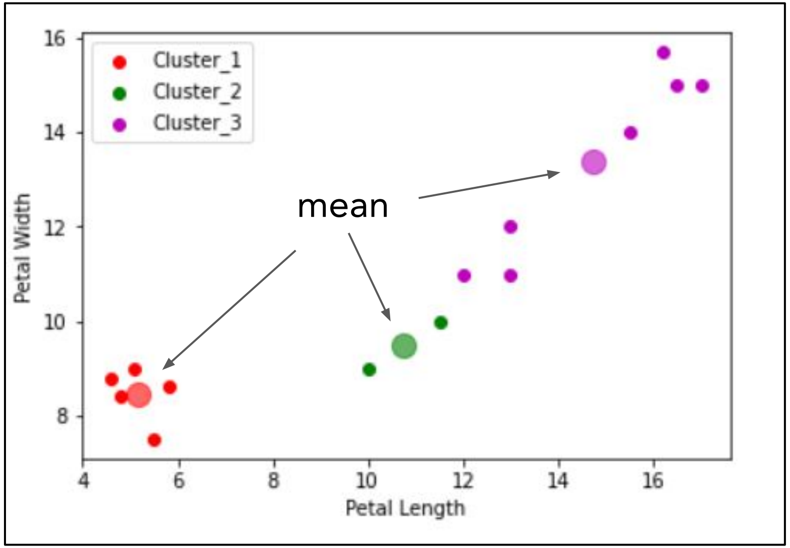


Example: step 4

1st iteration:

In step 3, we have obtained 3 clusters based on the initial centroid assignment. Now calculate the means of these clu

Cluster	Mean Petal Length	Mean Petal Width
1	5.16	8.46
2	10.75	9.50
3	14.74	13.39



Example: step 5

Now again calculate the Euclidean distance of each point from the newly obtained cluster centroids.

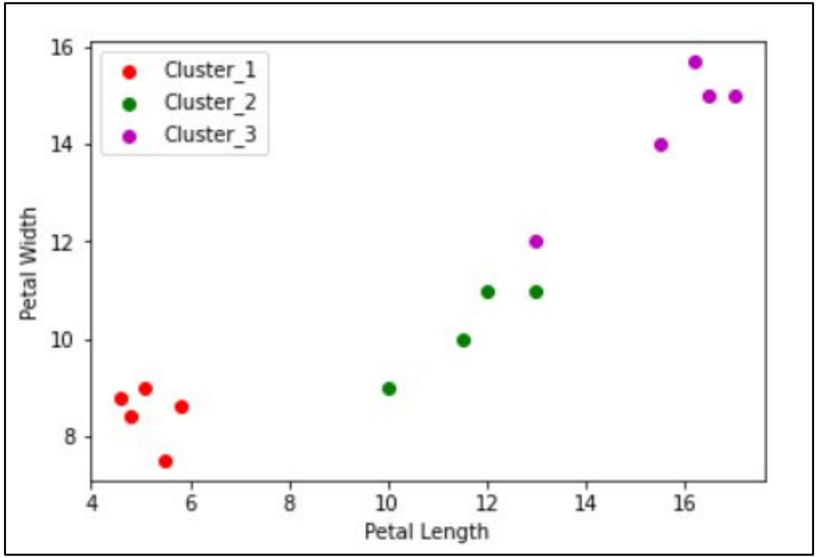
We can see that some of the data points are closer to the new centroid. Also, two data points are shifted to the 2nd cluster from the 3rd cluster.

Euclidean Distance		
Dist_1	Dist_2	Dist_3
1.018430165	5.618051264	10.95773858
13.52616723	8.325412903	2.774997984
13.09072954	7.956915231	2.386099498
0.364965752	6.05082639	11.12284808
6.524354374	0.901387819	4.688195902
0.655133574	5.031152949	10.14286694
4.870030801	0.901387819	6.459812676
0.655133574	6.189709202	11.13132162
11.73060953	6.543126164	0.97499375
8.602162519	3.363406012	2.226601404
0.543323108	5.672080747	10.59335539
7.296382665	1.952562419	3.635229816
13.20224223	8.254847061	2.734809941
8.241189235	2.704163457	2.95451888

Example: step 5

Assign the data to the nearest cluster.

The plot shows that 1st cluster contains same 5 points as before, the 2nd cluster have 2 new points, and the 3rd cluster contains 5 points.



Example: step 6

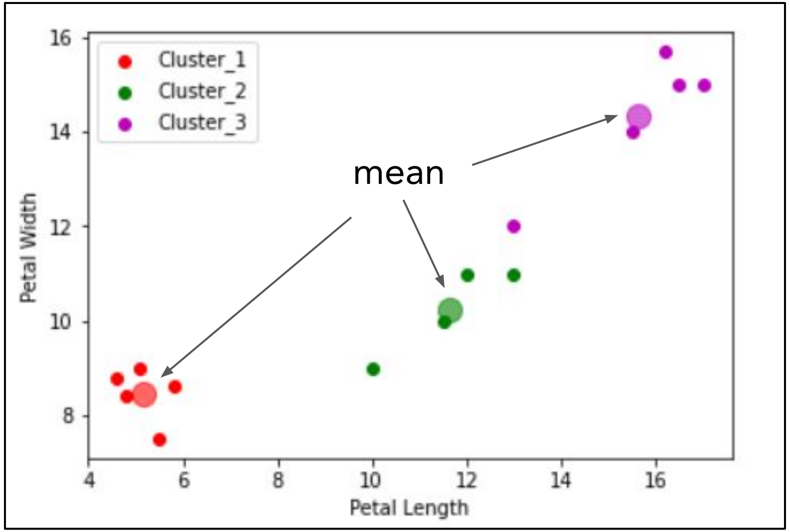
Repeat steps 4 and 5, until the cluster centroids remains same.

Example: repeat step 4

2nd iteration:

In step 5, we have obtained 3 clusters based on the centroids. Now calculate the means of the new clusters

Cluster	Mean Petal Length	Mean Petal Width
1	5.16	8.46
2	11.63	10.25
3	15.64	14.34



Note: The centroid for the cluster 1 is same as in the previous step.

Example: repeat step 5

Now again calculate the Euclidean distance of each point from the newly obtained cluster centroids.

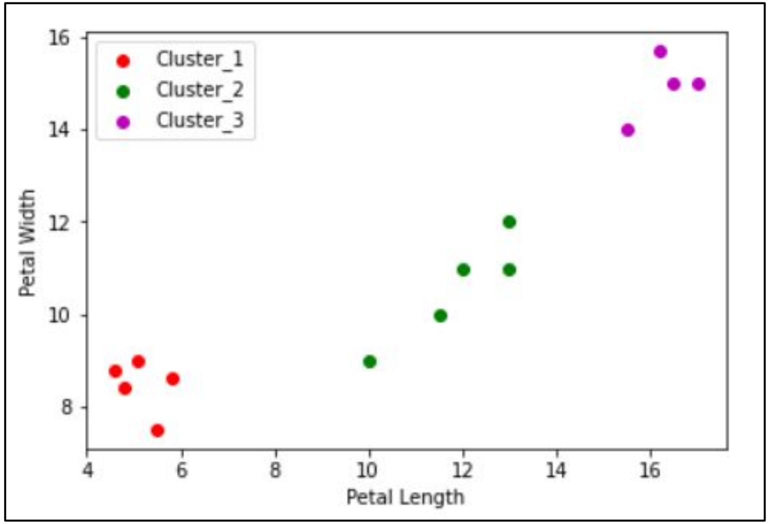
The table shows that a new point is shifted to the 2nd cluster from the 3rd cluster.

Euclidean Distance		
Dist_1	Dist_2	Dist_3
1.018430165	6.718586161	12.23132045
13.52616723	7.169337487	1.511687798
13.09072954	6.802896442	1.084066419
0.364965752	7.076114753	12.36079285
6.524354374	0.281780056	5.997932977
0.655133574	6.058993316	11.39180407
4.870030801	2.054117816	7.766929895
0.655133574	7.177980217	12.35205246
11.73060953	5.388821764	0.367695526
8.602162519	2.222476097	3.527775503
0.543323108	6.648563755	11.81554908
7.296382665	0.836301381	4.940161941
13.20224223	7.112481986	1.470782105
8.241189235	1.561857868	4.25737008

Example: repeat step 5

Assign the data to the nearest cluster.

Now the clusters are created as per our observation about the data.

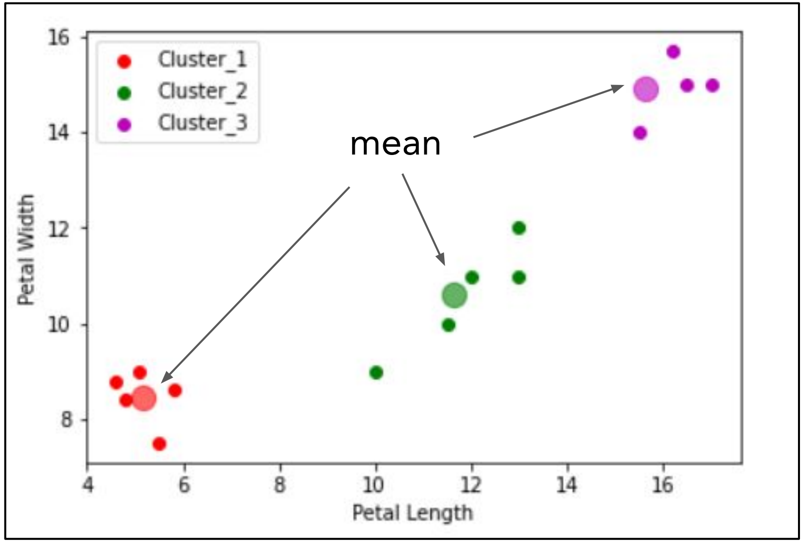


Example: repeat step 4

3rd iteration:

In previous step, we have obtained 3 clusters based on the centroids. Now calculate the means of the new clusters.

Cluster	Mean Petal Length	Mean Petal Width
1	5.16	8.46
2	11.9	10.6
3	16.3	14.93



Note: The centroid for the cluster 1 is same as in the previous step.

Example: repeat step 5

Now again calculate the Euclidean distance of each point from the newly obtained cluster centroids.

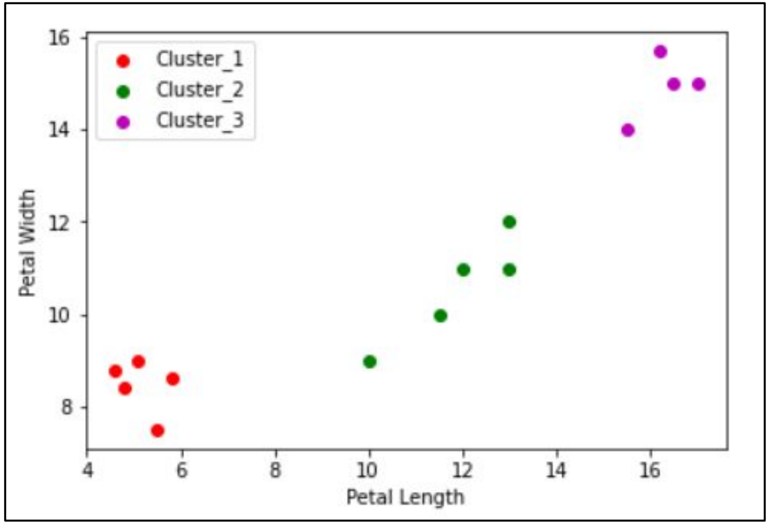
The table shows that the points belongs to the same cluster.

Euclidean Distance		
Dist_1	Dist_2	Dist_3
1.018430165	7.111258679	13.10896258
13.52616723	6.735725648	0.703491293
13.09072954	6.365532185	0.211896201
0.364965752	7.433034374	13.22463232
6.524354374	0.721110255	6.880763039
0.655133574	6.419501538	12.26046084
4.870030801	2.48394847	8.65187263
0.655133574	7.518643495	13.2085919
11.73060953	4.951767361	1.226743657
8.602162519	1.780449381	4.413037503
0.543323108	6.985699679	12.67299886
7.296382665	0.412310563	5.825366941
13.20224223	6.670832032	0.776466355
8.241189235	1.170469991	5.131754086

Example: repeat step 5

Assign the data to the nearest cluster.

There is no change in the points belonging to each cluster.

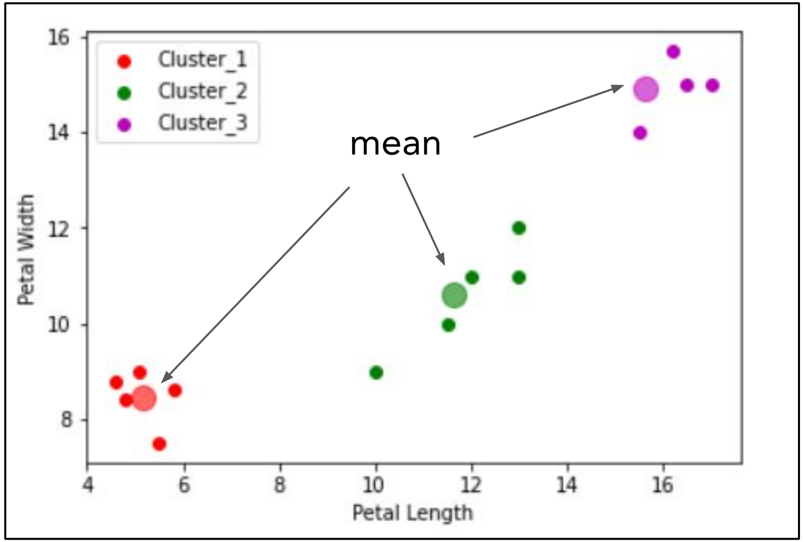


Example: repeat step 4

4th iteration:

In previous step, we have obtained 3 clusters based on the centroids. Now calculate the means of the new clusters.

Cluster	Mean Petal Length	Mean Petal Width
1	5.16	8.46
2	11.9	10.6
3	16.3	14.93



Example: repeat step 4

4th iteration:

- We can see that all the cluster centroids are the same as in the previous iteration. Thus we will stop the algorithm

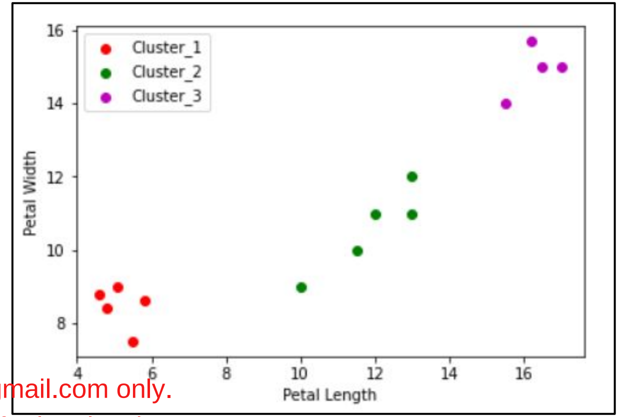
- The table shows the three clusters with the cluster centroids

Cluster	Mean Petal Length	Mean Petal Width
1	5.16	8.46
2	11.9	10.6
3	16.3	14.93

Inference

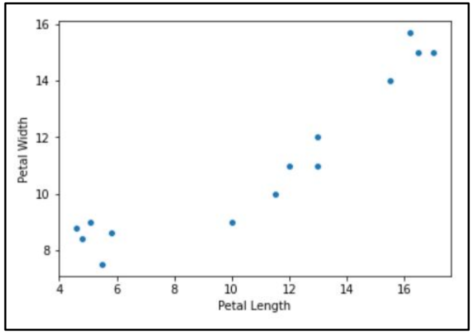
- The first cluster contains 5 flowers with the least average petal length and width. This represents the group of small-sized flowers
- The second cluster represents 5 medium-sized flowers
- The third cluster consists of 4 flowers with the highest average petal length and width
- Thus, K-means has clustered the data into 3 clusters based on the length and width of each flower petal

Cluster	Mean Petal Length	Mean Petal Width
1	5.16	8.46
2	11.9	10.6
3	16.3	14.93

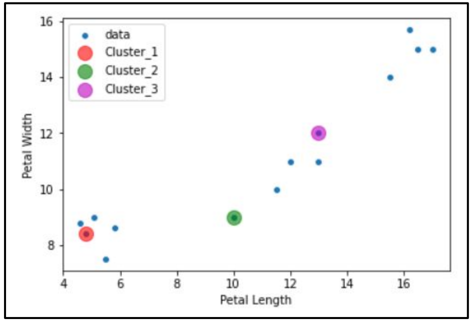


Summary

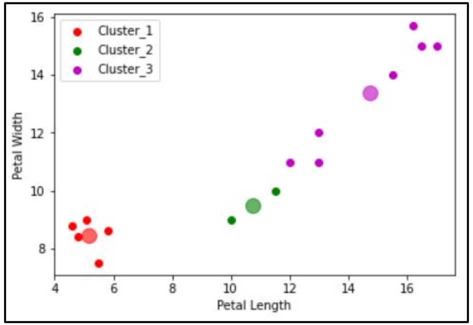
1.



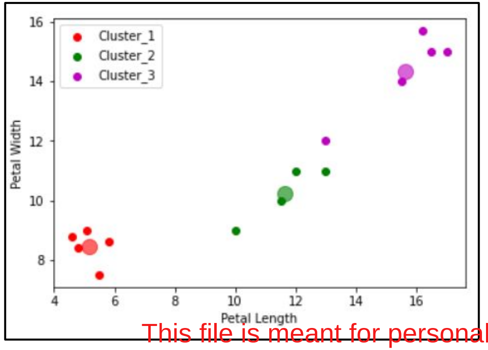
2.



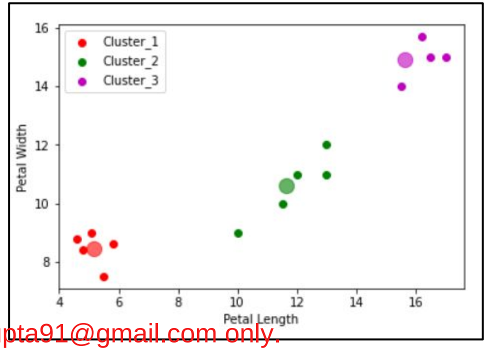
3.



4.



5.



Consider different initial centroids and check whether you can get the same clusters.

Merits and demerits

- Merits:
    - Easy to understand
    - Simple implementation

- Demerits:
    - Finding the optimal value of K can be computationally expensive
    - Initial centroid assignment affects the final output
    - Not efficient in presence of outliers

WANT TO KNOW MORE?

Clustering algorithms

- In case of categorical data use k-modes algorithm (read more: Link)

- For mixed (numerical and categorical) data type use k-prototypes algorithm (read more: Link)

Optimal Value of K

Optimal value of K

- In the previous example, it was obvious that the observations were divided into 3 groups; thus we considered K = 3

- But in general, it is not easy to decide the optimal value of K

- In this session, we study two common techniques to find the optimal value of K

- One of the methods is using the Elbow Plot and the other one is the Silhouette Method

Elbow plot

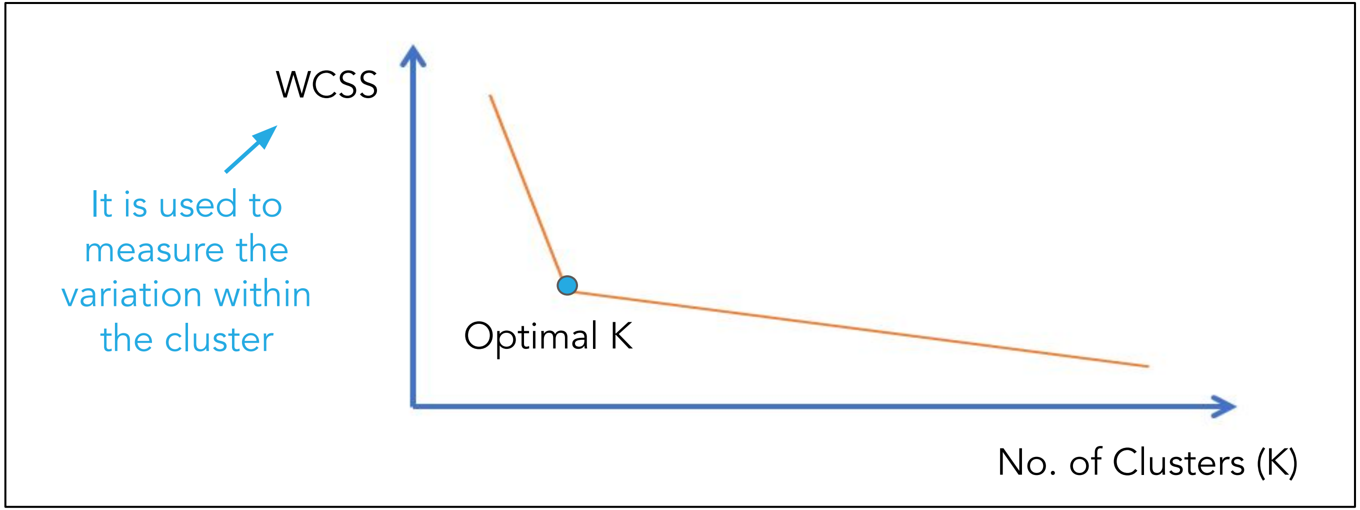
- K-means clustering aims to reduce the within-cluster variation
- The elbow (or scree) plot is used to plot the within-cluster sum of squares (WCSS) for different values of K
- Optimal K is the value corresponding to the elbow point

$$WCSS = \sum_{C_j=1}^{K} \sum_{x_i \epsilon C_j} ||x_i - \mu_j||^2$$

Where,

C_j is the j^{th} cluster and μ_j is the centroid of the j^{th} cluster

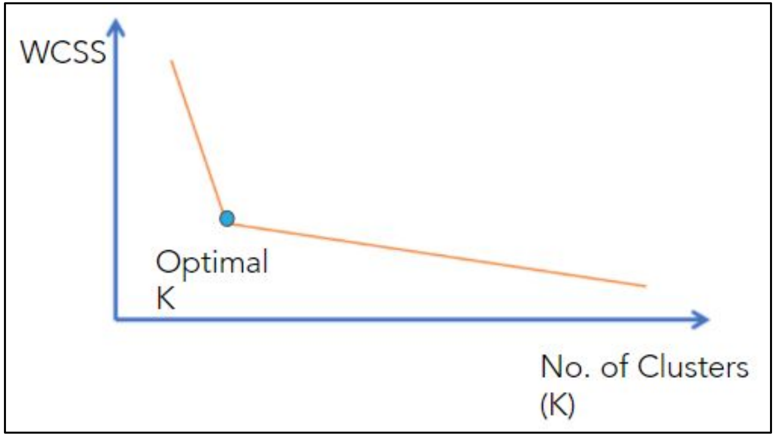
Elbow plot



Higher the value of WCSS	Lower the value of WCSS
Higher is the variation within the cluster	Lower is the variation within the cluster

Elbow plot

- The plot shows that the WCSS is decreasing rapidly for the K values less than the optimal K value
- After the elbow point, the WCSS is steadily decreasing which implies that more clusters are formed by dividing the large clusters into subgroups
- Selecting the K greater than optimal K leads to overfitting



PLEASE NOTE

In python, the attribute 'inertia_' returns the WCSS for a specific value of K. We use the different WCSS and K values to plot the elbow plot.

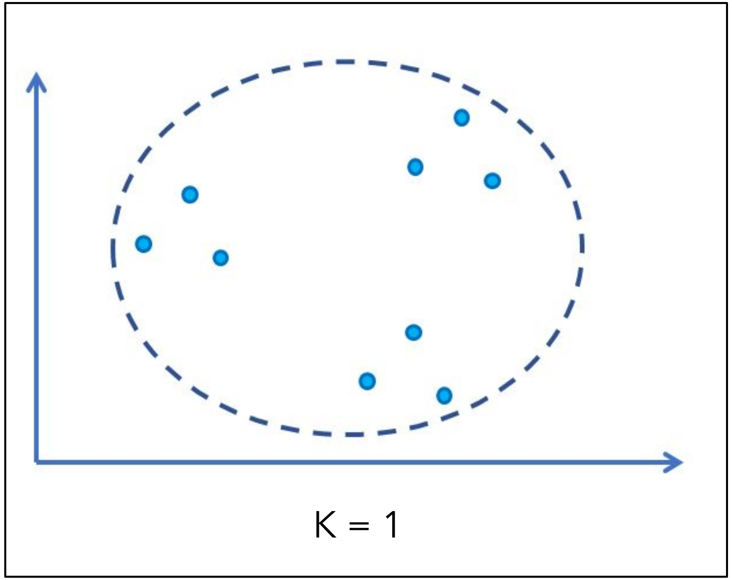
```
# consider an empty list to store WCSS for each K
wcss = []

# perform K-means with different K values
for i in list_K:
    model = KMeans(n_clusters= i, random_state = 10)
    # fit the model
    model.fit(df_data)
    # 'inertia_' returns the WCSS for a specific value of K
    wcss.append(model.inertia_)

# plot the elbow plot
plt.plot(K, wcss)
```

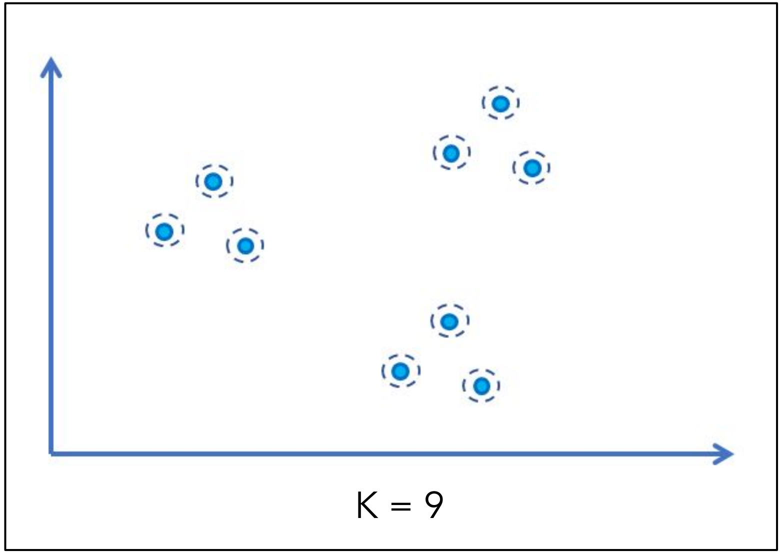
Elbow plot

- Suppose there are 9 observations
- The minimum value of K that we can have is one
- The variation within the cluster is maximum as all the observations are grouped inside a single cluster. But our aim is to reduce this variation



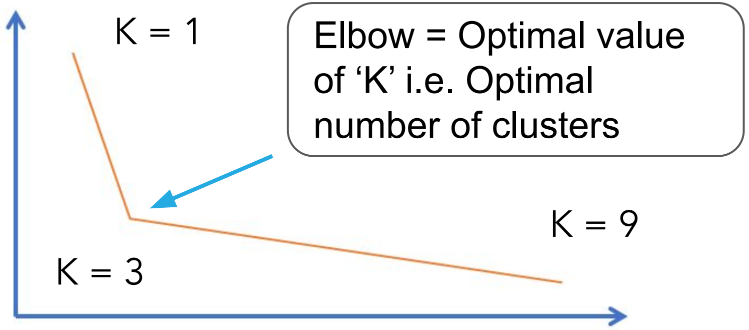
Elbow plot

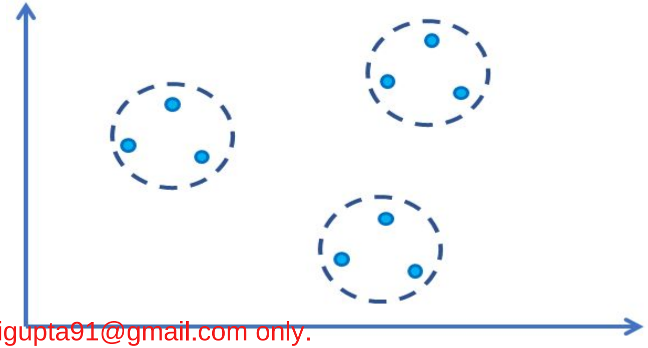
- The maximum value of K that we can have is 9 (total number of observations)
- Here there is the least variation within the cluster, as every observation is a cluster
- Thus we took care of the first objective of clustering. But the second objective; i.e. increase the variation between clusters, fails



Elbow plot

- The optimal value of K lies between 1 and 9
- The elbow plot shows that K = 3 is the optimal value
- Here the variation within the cluster is minimum and, the variation between the clusters is maximum





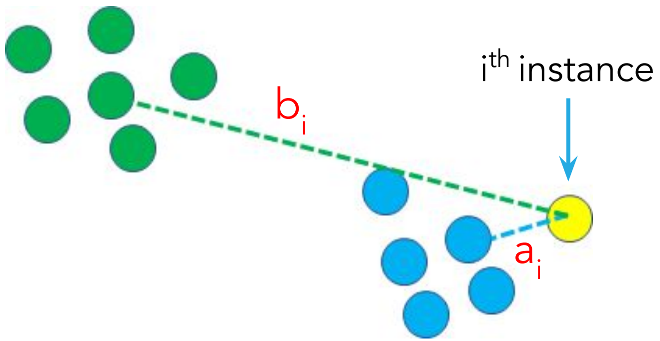
Silhouette method

- Silhouette score is used to find the optimal number of clusters
- It is the mean silhouette coefficient over all the instances
- The value of the silhouette score lies between -1 to +1
- We plot the silhouette score for different values of ‘K’ and select the K with the highest score
- It is a computationally expensive method
- It is also used to validate the quality of clusters

Silhouette coefficient

Silhouette coefficient of a single instance (observation) is given by:

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

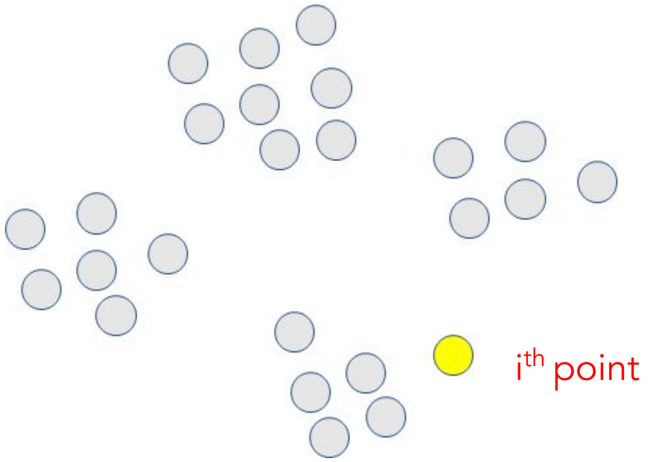


a_i = mean distance between i^{th} point and other points in the same cluster (mean intra-cluster distance)

b_i = mean distance between i^{th} point and points of the next closest cluster (mean inter cluster

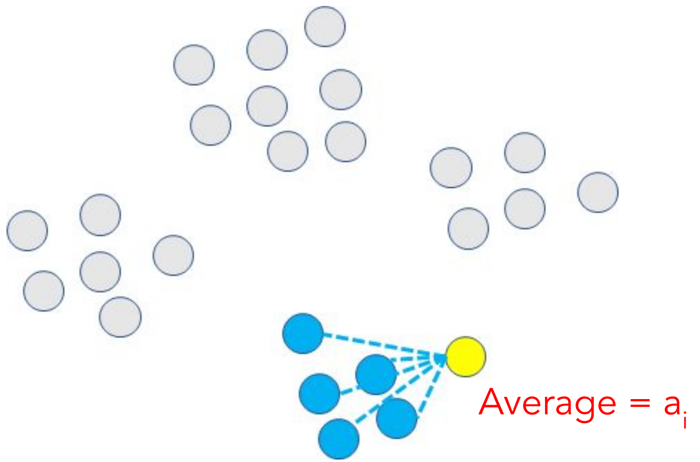
Step 1

- For each point, we have to calculate the silhouette coefficient
- Let's consider the i^{th} point for which we will calculate the silhouette coefficient



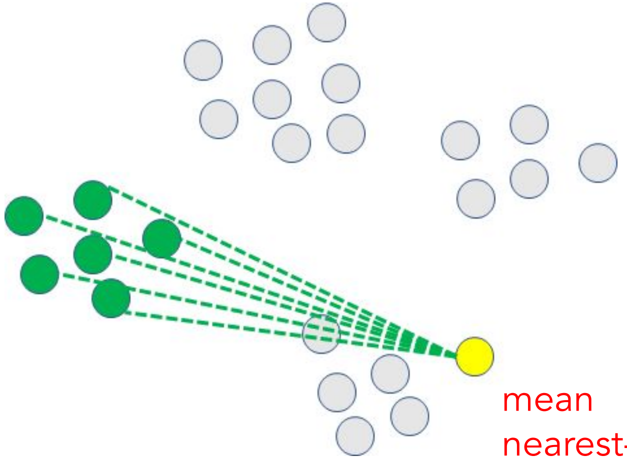
Step 2

- Calculate mean intra-cluster distance (a_i)
- It is the average distance between i^{th} point and other points in the same cluster



Step 3

- For the same point, calculate mean inter-cluster distance(b_i)
- It is the average distance between i^{th} point and points of the next closest cluster



mean
nearest-cluster
= b_i

Step 4

- Calculate s_i using the obtained values of a_i and b_i
- Similarly calculate silhouette coefficient for each observation
- The average silhouette coefficient of all the observations is the silhouette score

PLEASE!! NOTE

- As per the objectives of cluster analysis, the variation within the cluster should be minimum and the variation between clusters should be maximum
- Thus we want a_i to be much smaller than b_i, i.e. $a_i << b_i$
- Ideally we want $a_i = 0$ and b_i = infinity

Best case scenario

If $a_i = 0$ and b_i = infinity. Thus

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

$$s_i = \frac{(\infty - 0)}{max(0, \infty)}$$

$$s_i = 1$$

Silhouette coefficient near to +1 indicates that the observation is well set inside its own cluster (within cluster variation is minimum) and far from the other clusters (between clusters variation is more).

Worst case scenario

If a_i = infinity and b_i = 0. Thus

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

$$s_i = \frac{(0 - \infty)}{max(\infty, 0)}$$

$$s_i = -1$$

Silhouette coefficient close to -1 indicates that the observation has been assigned to the wrong cluster.

Silhouette method

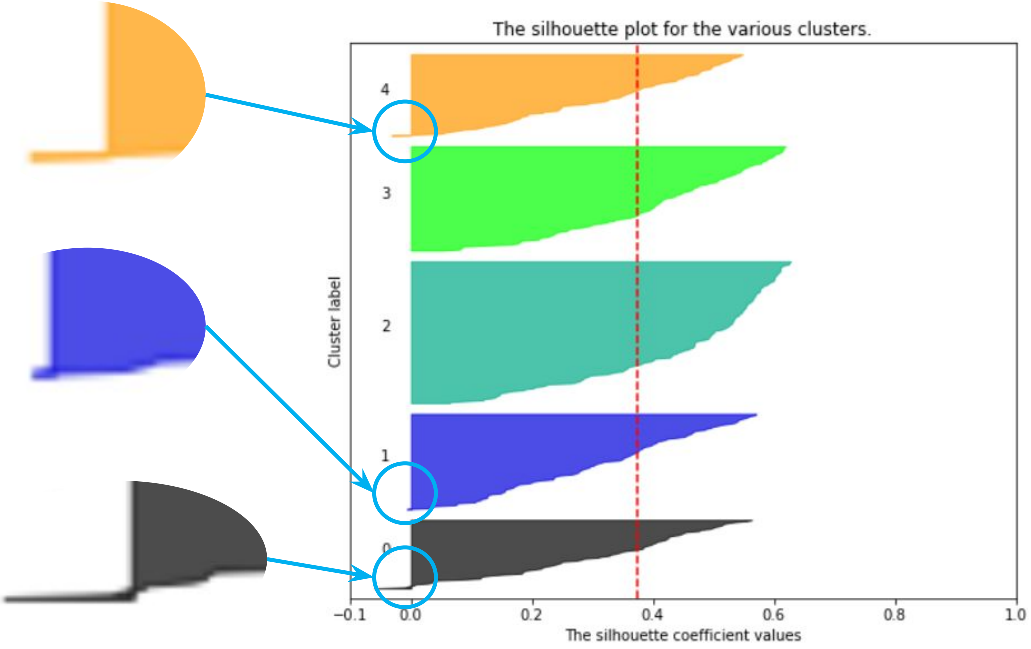
- There are several criteria to choose the optimal K using a silhouette plot
  - Select a value of K such that there are no outliers in each cluster
  - Select a value of K for which all the silhouette coefficients are greater than the average silhouette score
  - Select a value of K that has the highest average silhouette score

Identify outliers using silhouette plot

- Any silhouette coefficient that is less than 0 is considered as an outlier

- In this case there are outliers in the 4^{th}, 1^{st} and 0^{th} cluster

- Thus, we can say that the value of k = 5 is not a good choice for k

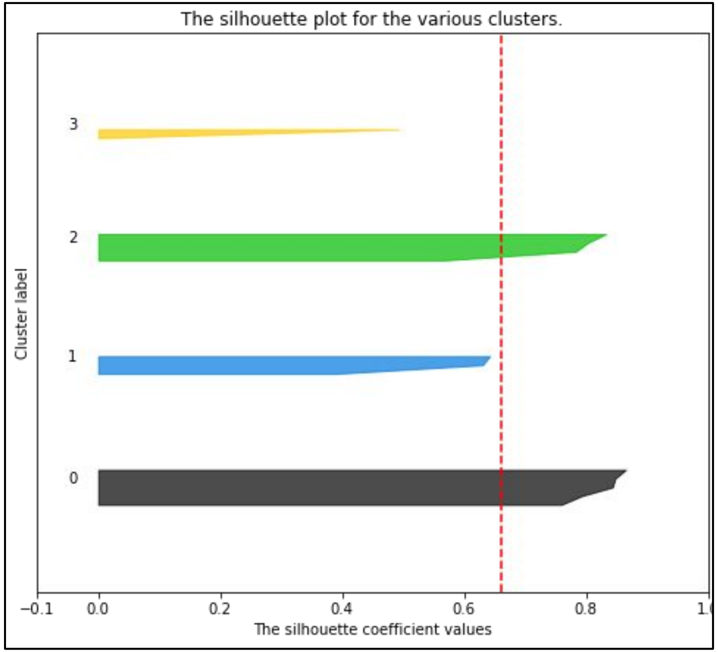


Silhouette coefficient less than average score

- Generally the silhouette score of each cluster should be greater than the average score

- In this figure, the average score is shown by a red dotted vertical line (----)

- The silhouette coefficient for the 1st and 3rd cluster is less than the average score; thus K = 4 is not a good choice of K



Maximum silhouette score

The value of K associated with the highest average silhouette score can be considered as an optimal value.

```
For n_clusters = 2 The average silhouette_score is : 0.6290071473108994
For n_clusters = 3 The average silhouette_score is : 0.7425098174909255
For n_clusters = 4 The average silhouette_score is : 0.6607175518100986
For n_clusters = 5 The average silhouette_score is : 0.5620626321631084
For n_clusters = 6 The average silhouette_score is : 0.48617806040304107
```

PLEASE!! NOTE

In python, the ‘silhouette_score()’ is used to calculate the silhouette score for a specific value of K.

```
# import the function
from sklearn.metrics import silhouette_score

# consider an empty list to store the silhouette score for each K
score = []

# perform K-means with different K values
# 'silhouette_score' function computes the silhouette score for each K
for i in list_K:
    cluster = KMeans (n_clusters= i, random_state= 10)
    # fit the model and predict the cluster label
    predict = cluster.fit_predict(df_data)

    # 'silhouette_score()' computes the silhouette score for a specific K
    score.append(silhouette_score(df_data, predict))
```

Summary

- Elbow method and Silhouette score are two of the methods used to find the optimal value of K
- Elbow method uses intra-cluster distance to determine the value of K
- Silhouette score uses intra-cluster and inter-cluster distance
- Silhouette plot can be used to detect the outliers

PES UNIVERSITY

Thank You