Experiment No.06

A.1 Aim: Implementation of K-means clustering using any programming language like JAVA, C++, Python or WEKA Tool.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)

Roll No. 50	**Name:** AMEY THAKUR
Class: Comps TE B	**Batch:** B3
Date of Experiment: 15/04/2021	**Date of Submission:** 15/04/2021
Grade:	

B.1 Software Code written by a student:

(Paste your problem statement related to your case study completed during the 2 hours of practice in the lab here)

```
# importing libraries
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('diabetes_csv.csv')
x = dataset.iloc[:, [7, 5]].values

#finding optimal number of clusters using the elbow method
from sklearn.cluster import KMeans
wcss_list= [] #Initializing the list for the values of WCSS

#Using a loop for iterations from 1 to 10.
for i in range(1, 11):
   kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
   kmeans.fit(x)
   wcss_list.append(kmeans.inertia_)
mtp.plot(range(1, 11), wcss_list)
mtp.title('The Elbow Method Graph')
```

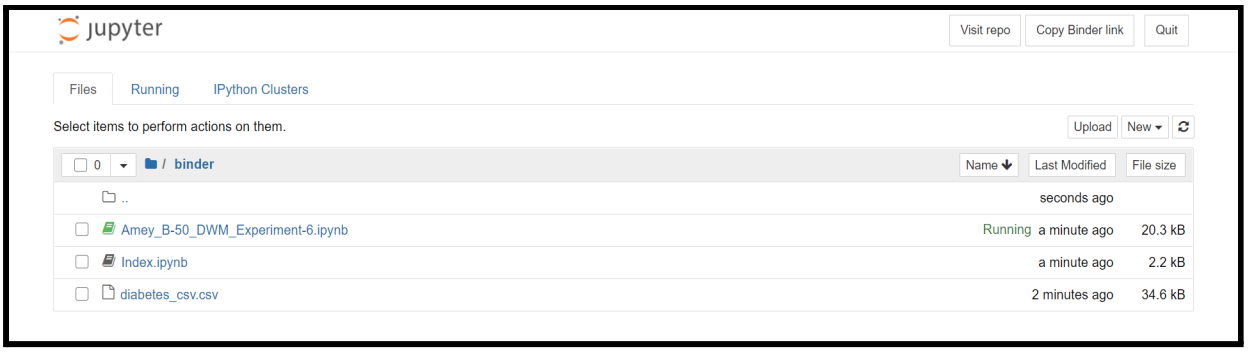
```
mtp.xlabel('Number of clusters(k)')
mtp.ylabel('wcss_list')
mtp.show()

#training the K-means model on a dataset
kmeans = KMeans(n_clusters=2, init='k-means++', random_state= 42)
y_predict= kmeans.fit_predict(x)

mtp.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c = 'blue', label = 'Cluster 1') #for first cluster
mtp.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c = 'green', label = 'Cluster 2') #for second cluster
mtp.scatter(kmeans.cluster_centers_[:, 0],
kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroid')
mtp.title('Clusters of patients')
mtp.xlabel('Age(in years)')
mtp.ylabel('BMI(Body Mass Index)')
mtp.legend()
mtp.show()
```

B.2 Input and Output:

(Paste your program input and output in the following format, If there is an error then paste the specific error in the output part. In case of an error with the due permission of the faculty, an extension can be given to submit the error-free code with output in due course of time. Students will be graded accordingly.)

Jupyter Notebook

Jupyter/Binder:



Jupyter Notebook:

jupyter Amey_B-50_DWM_Experiment-6 Last Checkpoint: 4 minutes ago (autosaved)

Visit repo | Copy Binder link

File Edit View Insert Cell Kernel Widgets Help

Notebook saved | Trusted | Python 3

Run | Code | Download | GitHub | Binder

Memory: 225 / 2048 MB

In [1]:
```
# Importing libraries
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
```

In [2]:
```
# Importing the dataset
dataset = pd.read_csv('diabetes_csv.csv')
x = dataset.iloc[:, [7, 5]].values
```

In [3]:
```
#finding optimal number of clusters using the elbow method
from sklearn.cluster import KMeans
wcss_list= [] #Initializing the list for the values of WCSS
```

jupyter Amey_B-50_DWM_Experiment-6 Last Checkpoint: 4 minutes ago (autosaved)

Visit repo | Copy Binder link

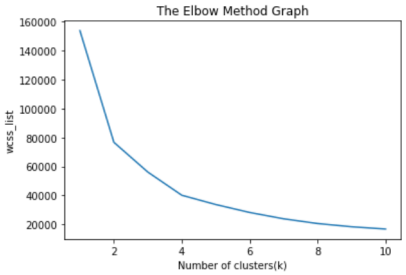
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Code | Download | GitHub | Binder

Memory: 225 / 2048 MB

In [4]:
```
#Using for loop for iterations from 1 to 10.
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
mtp.plot(range(1, 11), wcss_list)
mtp.title('The Elbow Method Graph')
mtp.xlabel('Number of clusters(k)')
mtp.ylabel('wcss_list')
mtp.show()
```



jupyter Amey_B-50_DWM_Experiment-6 Last Checkpoint: 5 minutes ago (autosaved)

Visit repo | Copy Binder link

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Code | Download | GitHub | Binder

Memory: 225 / 2048 MB

In [5]:
```
#training the K-means model on a dataset
kmeans = KMeans(n_clusters=2, init='k-means++', random_state= 42)
y_predict= kmeans.fit_predict(x)
```

In [6]:
```
mtp.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c = 'blue', label = 'Cluster 1') #for first cluster
mtp.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c = 'green', label = 'Cluster 2') #for second cluster
mtp.scatter(kmeans.cluster_centers_[:, 0],
kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroid')
mtp.title('Clusters of patients')
mtp.xlabel('Age(in years)')
mtp.ylabel('BMI(Body Mass Index)')
mtp.legend()
mtp.show()
```

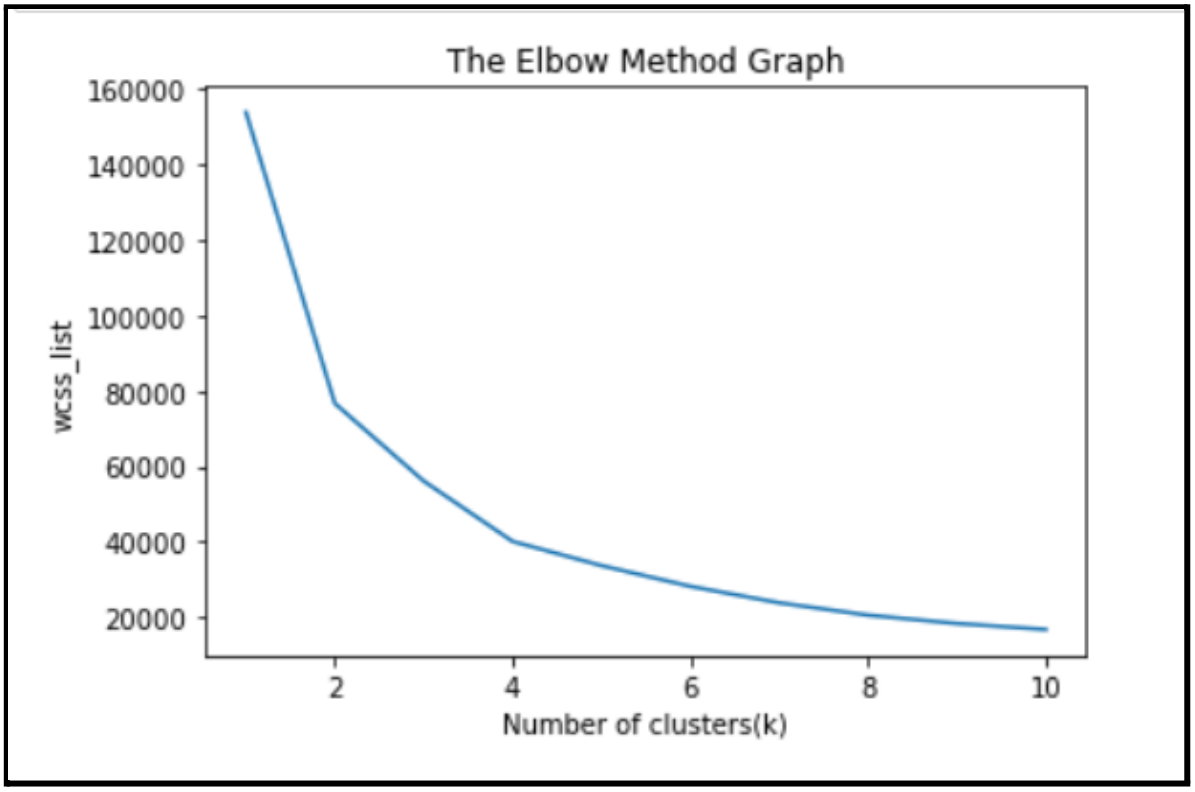
In [ ]:

Sample Dataset:

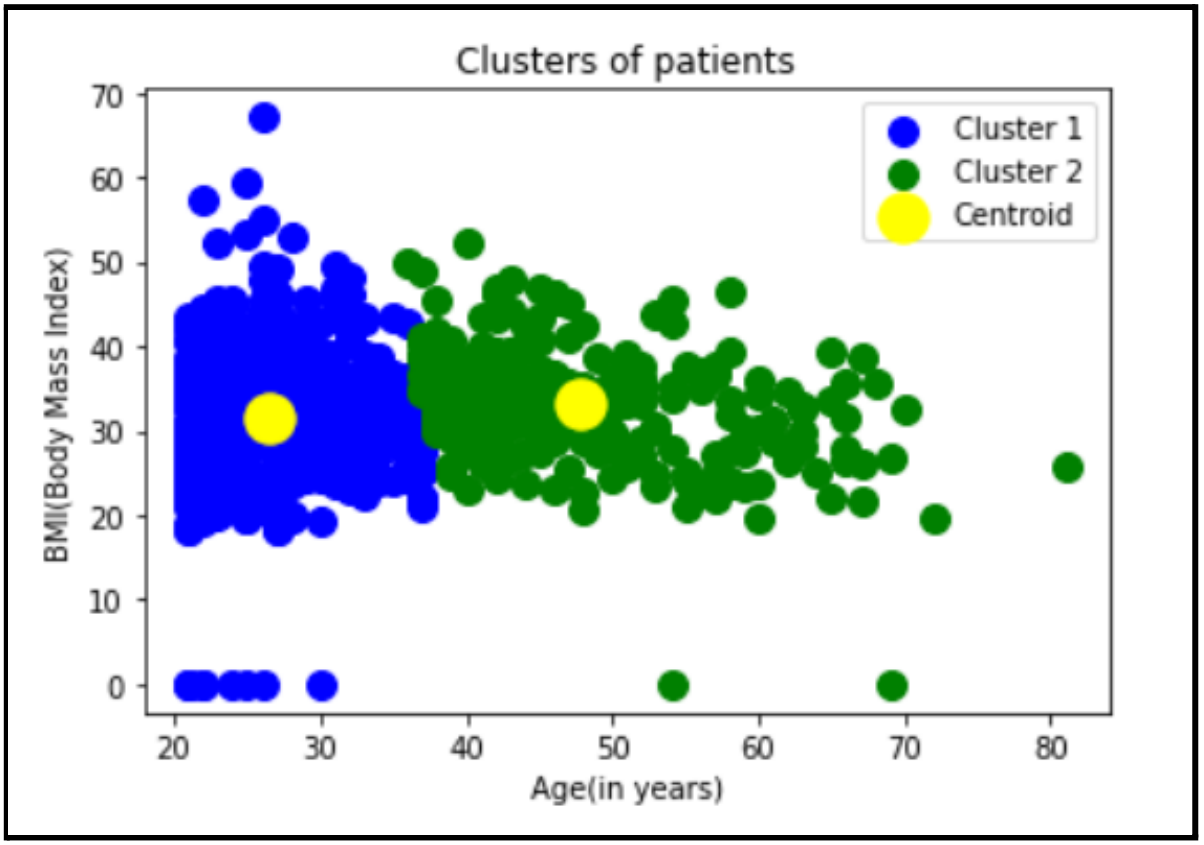
	preg	plas	pres	skin	insu	mass	pedi	age	class
0	6	148	72	35	0	33.6	0.627	50	tested_positive
1	1	85	66	29	0	26.6	0.351	31	tested_negative
2	8	183	64	0	0	23.3	0.672	32	tested_positive
3	1	89	66	23	94	28.1	0.167	21	tested_negative
4	0	137	40	35	168	43.1	2.288	33	tested_positive
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	tested_negative
764	2	122	70	27	0	36.8	0.340	27	tested_negative
765	5	121	72	23	112	26.2	0.245	30	tested_negative
766	1	126	60	0	0	30.1	0.349	47	tested_positive
767	1	93	70	31	0	30.4	0.315	23	tested_negative

768 rows × 9 columns

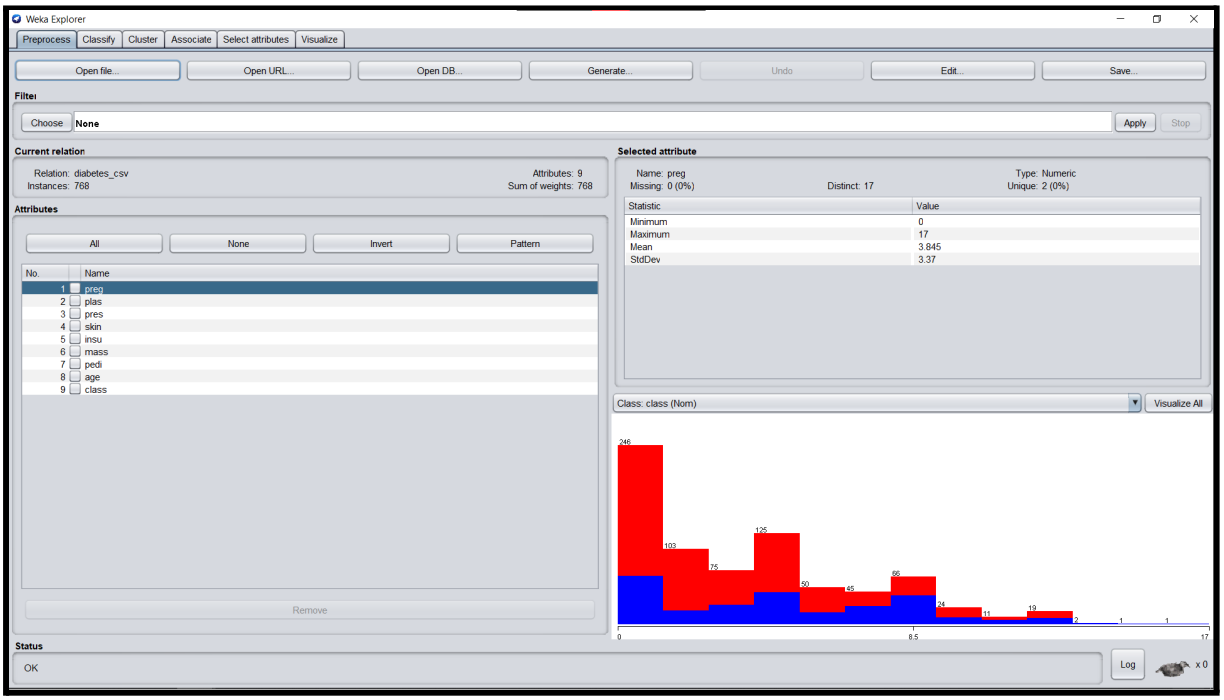
The Elbow Method Graph:

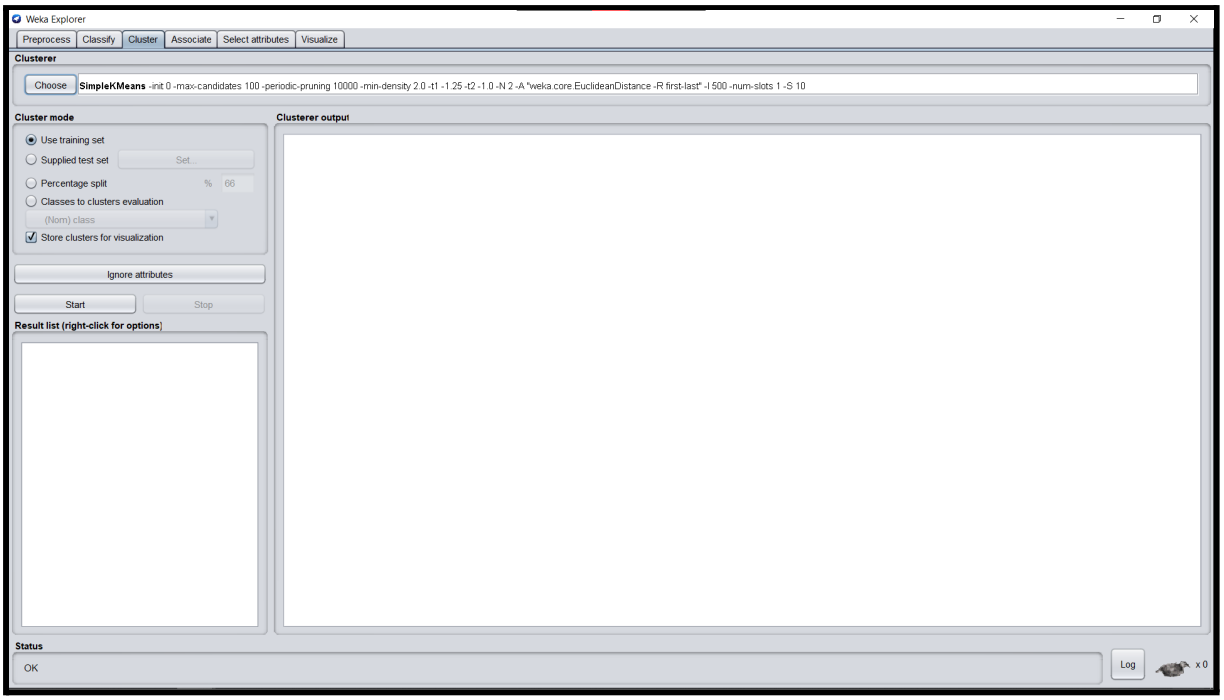


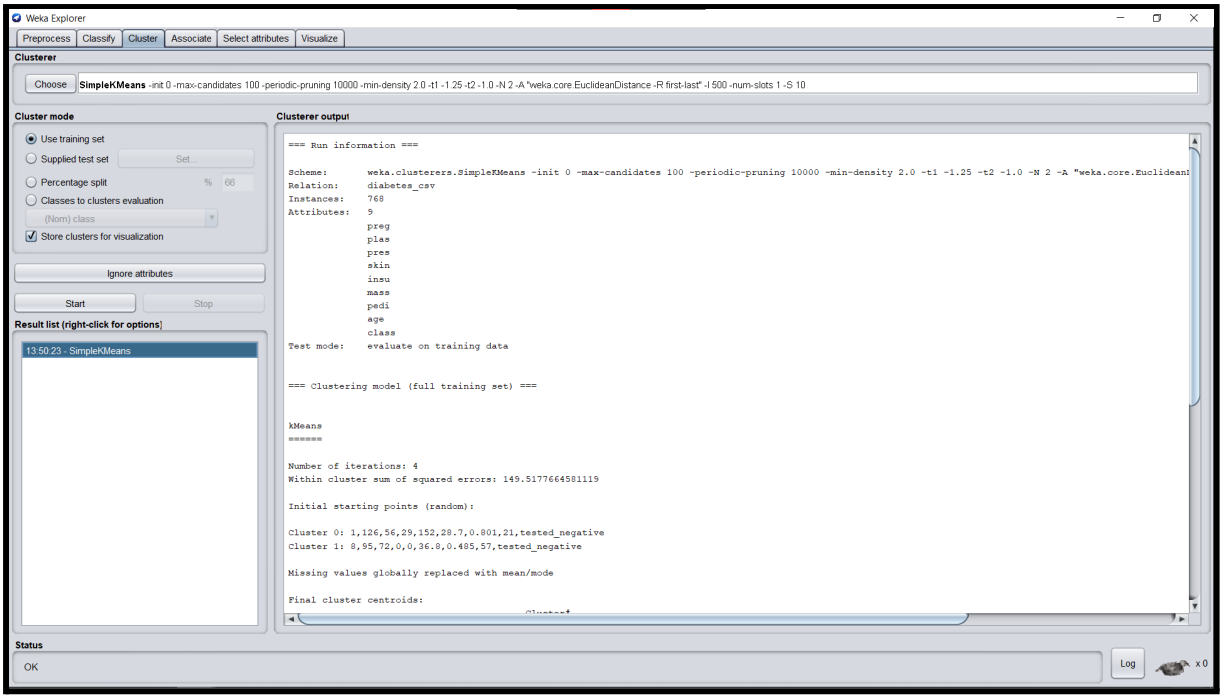
Python Output:

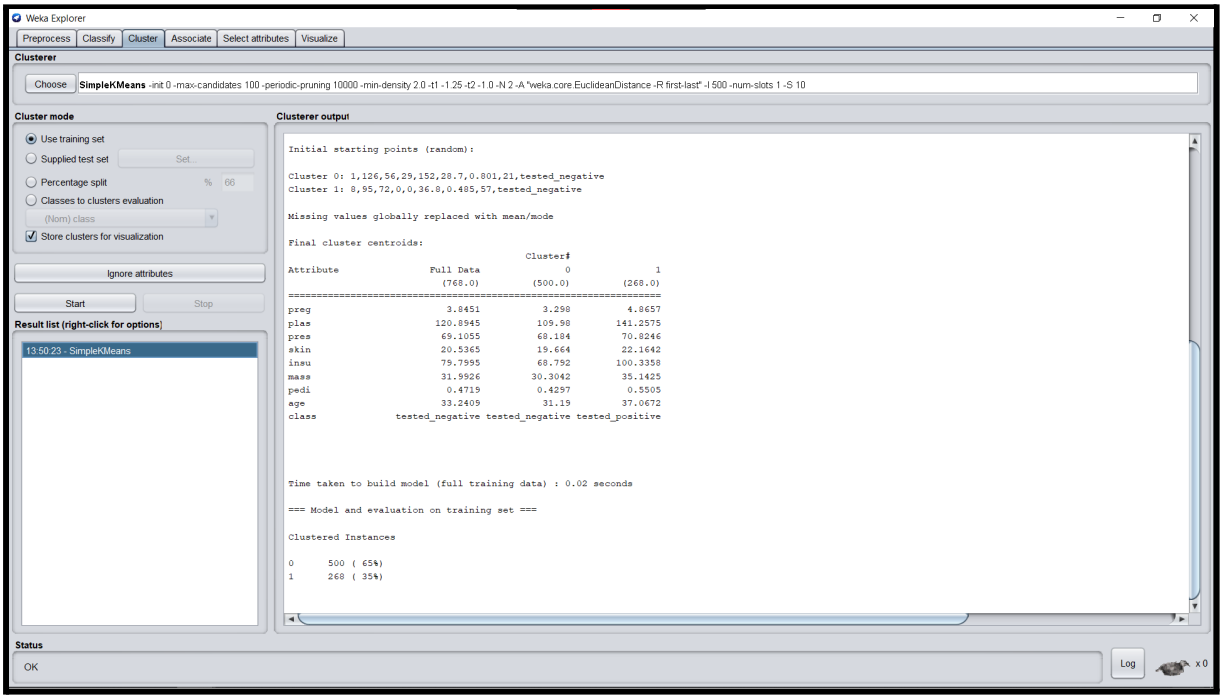


Weka Tool

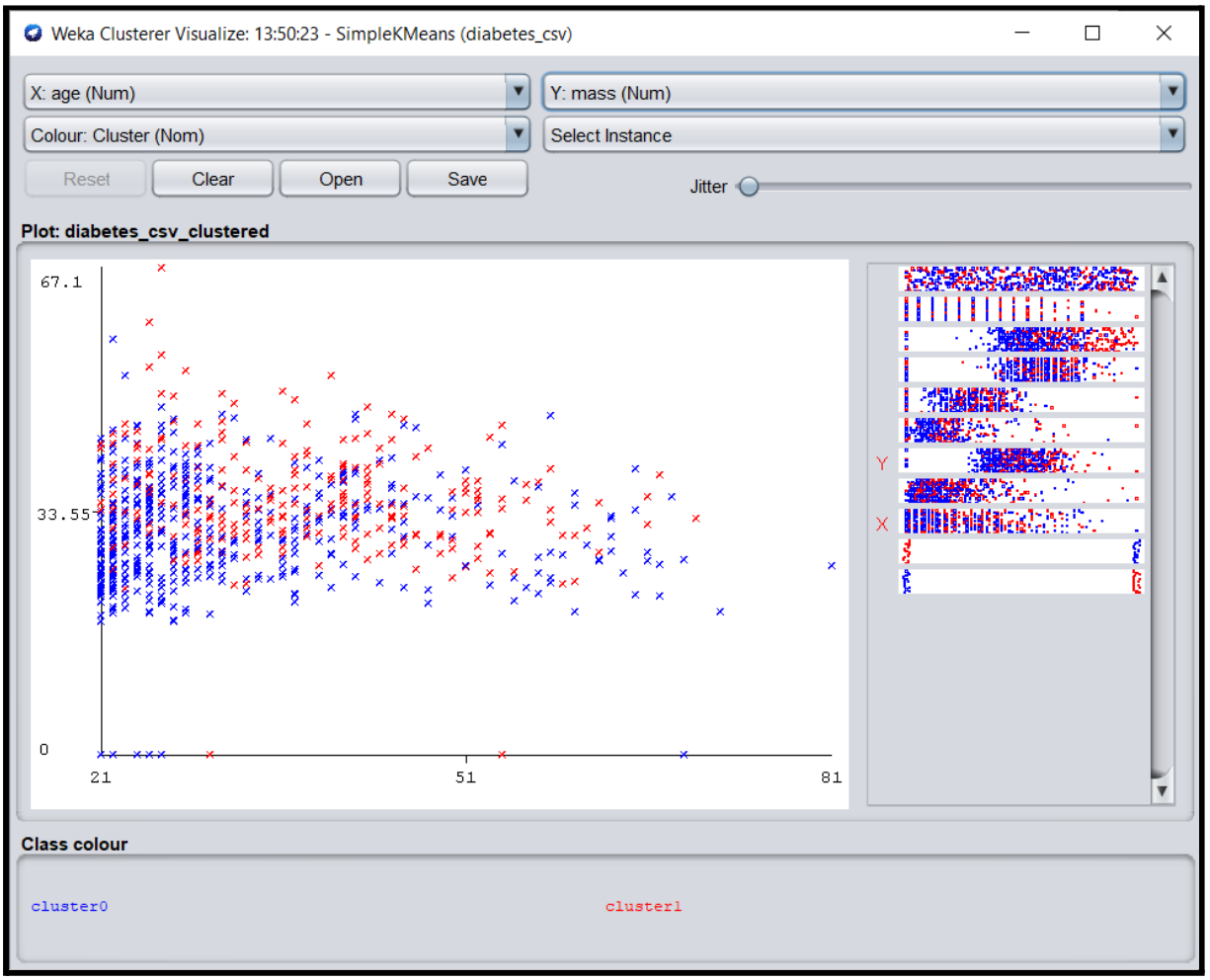

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: diabetes_csv
Instances: 768
Attributes: 9
Sum of weights: 768
Selected attribute
Name: preg
Missing: 0 (0%)
Distinct: 17
Type: Numeric
Unique: 2 (0%)
Statistic
Value
Minimum 0
Maximum 17
Mean 3.845
StdDev 3.37
Attributes
All
None
Invert
Pattern
No.
Name
1 preg
2 plas
3 pres
4 skin
5 insu
6 mass
7 pedi
8 age
9 class
Remove
Class: class (Nom)
Visualize All
246
103
75
125
50
45
66
24
11
19
2
1
1
0
8.5
17
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
Clusterer output
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
13:50:23 - SimpleKMeans
Clusterer output
=== Run information ===
Scheme: weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.Euclidean
Relation: diabetes_csv
Instances: 768
Attributes: 9
preg
plas
pres
skin
insu
mass
pedi
age
class
Test mode: evaluate on training data
=== Clustering model (full training set) ===
kMeans
======
Number of iterations: 4
Within cluster sum of squared errors: 149.5177664581119
Initial starting points (random):
Cluster 0: 1,126,56,29,152,28.7,0.801,21,tested_negative
Cluster 1: 8,95,72,0,0,36.8,0.485,57,tested_negative
Missing values globally replaced with mean/mode
Final cluster centroids:
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
13:50:23 - SimpleKMeans
Clusterer output
Initial starting points (random):
Cluster 0: 1,126,56,29,152,28.7,0.801,21,tested_negative
Cluster 1: 8,95,72,0,0,36.8,0.485,57,tested_negative
Missing values globally replaced with mean/mode
Final cluster centroids:
Cluster#
Attribute Full Data 0 1
(768.0) (500.0) (268.0)
preg 3.8451 3.298 4.8657
plas 120.8945 109.98 141.2575
pres 69.1055 68.184 70.8246
skin 20.5365 19.664 22.1642
insu 79.7995 68.792 100.3358
mass 31.9926 30.3042 35.1425
pedi 0.4719 0.4297 0.5505
age 33.2409 31.19 37.0672
class tested_negative tested_negative tested_positive
Time taken to build model (full training data) : 0.02 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 500 (65%)
1 268 (35%)
Status
OK
Log
x 0

Weka Output:



B.3 Observations and learning:

(Students are expected to comment on the output obtained with clear observations and learning for each task/ subpart assigned)

K -means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. A cluster refers to a collection of data points aggregated together because of certain similarities. The 'means' in the K-means refers to averaging of the data; that is, finding the centroid.

B.4 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

Hence we've successfully implemented K-means clustering through Python as well as Weka Tool.

B.5 Question of Curiosity

(To be answered by the student based on the practical performed and learning/observations)

1. What is Clustering? Types of clustering? Explain the advantages and disadvantages of clustering.

Ans:

Clustering

- ➔ Clustering is the task of dividing the population or data points into several groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is a collection of objects based on similarity and dissimilarity between them.
- ➔ Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters). It is the main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics, data compression, and computer graphics.

Types Of Clustering Algorithms

1. Connectivity-based Clustering (Hierarchical clustering)
2. Centroids-based Clustering (Partitioning methods)
3. Distribution-based Clustering
4. Density-based Clustering (Model-based methods)
5. Fuzzy Clustering
6. Constraint-based (Supervised Clustering)

Advantages and Disadvantages of Clustering

- ➔ The main advantage of a clustered solution is automatic recovery from failure, that is, recovery without user intervention.
- ➔ Disadvantages of clustering are complexity and inability to recover from database corruption.

2. Give the advantages and disadvantages of K- means clustering.

Ans:

Advantages of K-means clustering:

1. Relatively simple to implement.
2. Scales to large data sets.
3. Guarantees convergence.
4. Can warm-start the positions of centroids.
5. Easily adapts to new examples.
6. Generalizes to clusters of different shapes and sizes, such as elliptical clusters.

Disadvantages of K-means clustering:

1. Choosing manually.
2. Being dependent on initial values.
3. Clustering data of varying sizes and density.
4. Clustering outliers.
5. Scaling with several dimensions.

3. How is the number of clusters chosen?

Ans:

A fundamental step for any unsupervised algorithm is to determine the optimal number of clusters into which the data may be clustered. The Elbow Method is one of the most popular methods to determine this optimal value of k. Distortion: It is calculated as the average of the squared distances from the cluster centres of the respective clusters. Typically, the Euclidean distance metric is used. Inertia: It is the sum of squared distances of samples to their closest cluster centre. We iterate the values of k from 1 to 9 and calculate the values of distortions for each value of k and calculate the distortion and inertia for each value of k in the given range.