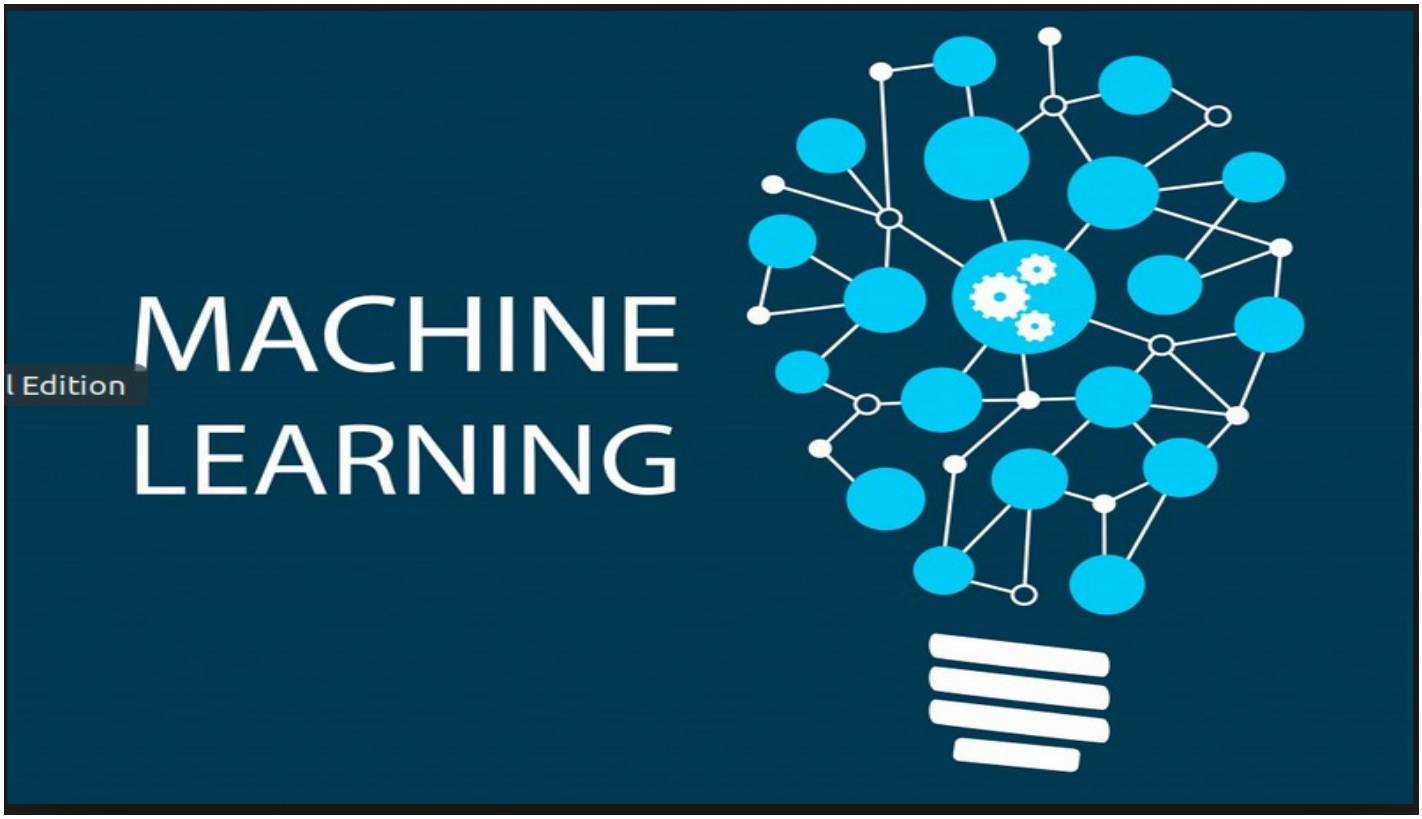
Machine Learning Cluster



Name: Merna Zakaria

ID:4106

Cluster:

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.

DATASET USED:

CIFAR:

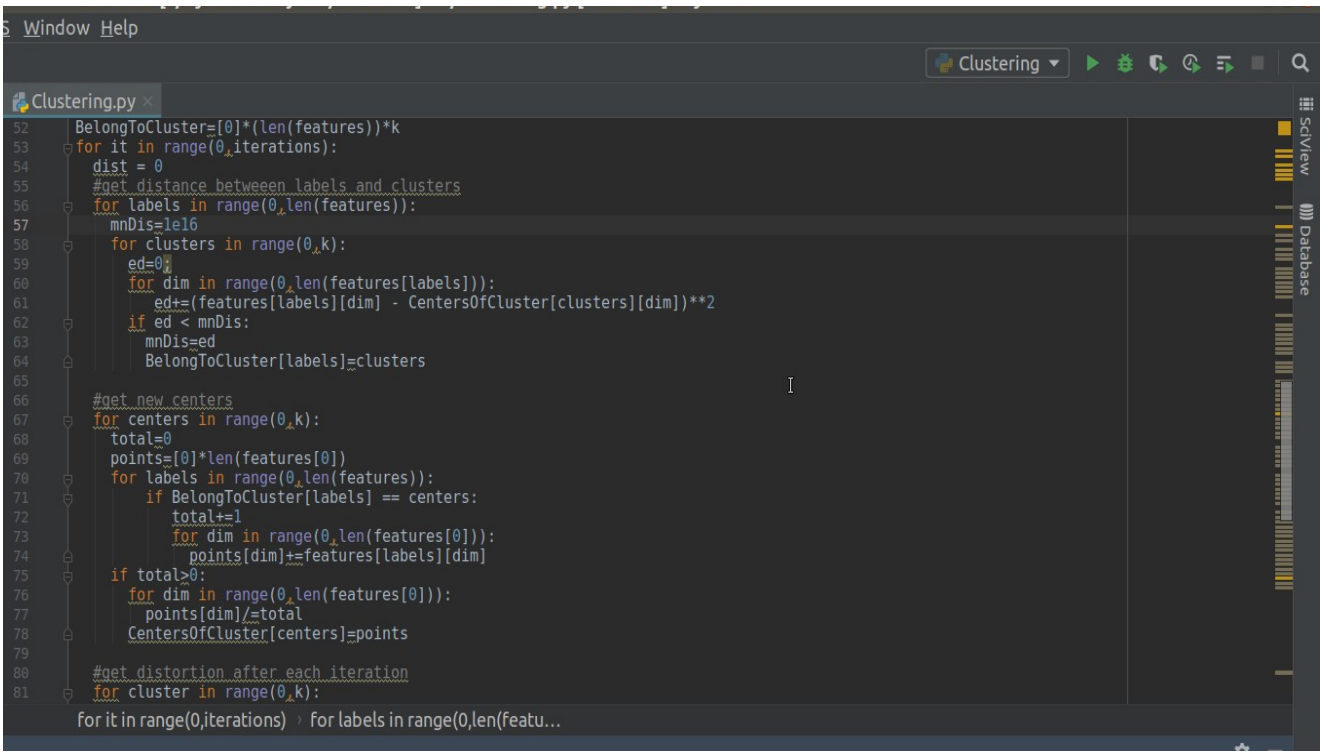
The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.

Each of the batch files contains a dictionary with the following elements:

- **data** -- a 10000x3072 numpy array of uint8s. Each row of the array stores a 32x32 colour image. The first 1024 entries contain the red channel values, the next 1024 the green, and the final 1024 the blue. The image is stored in row-major order, so that the first 32 entries of the array are the red channel values of the first row of the image.
- **labels** -- a list of 10000 numbers in the range 0-9. The number at index *i* indicates the label of the *i*th image in the array data.

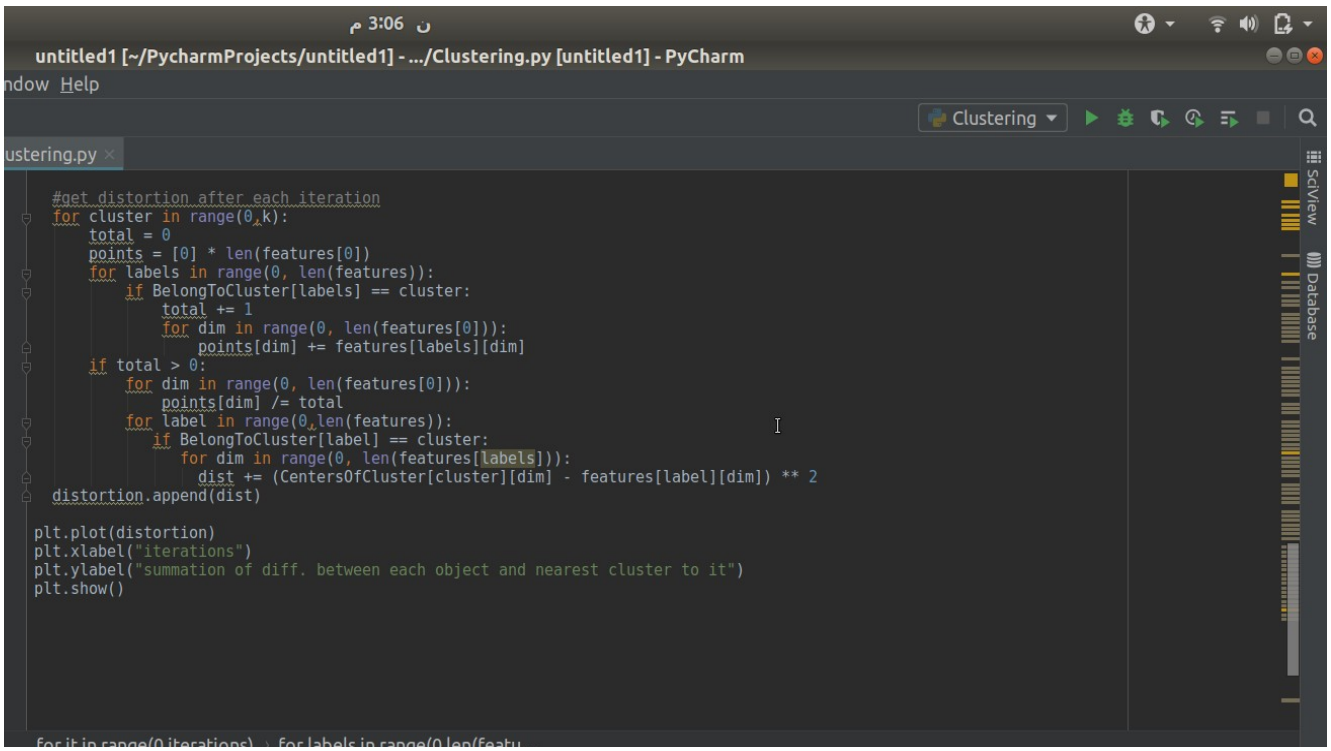
Screen Shot OF Cluster Code:



```
Window Help
Clustering
Clustering.py
BelongToCluster=[0]*(len(features))*k
for it in range(0,iterations):
  dist = 0
  #get distance betweeen labels and clusters
  for labels in range(0,len(features)):
    mnDis=1e16
    for clusters in range(0,k):
      ed=0;
      for dim in range(0,len(features[labels])):
        ed+=(features[labels][dim] - CentersOfCluster[clusters][dim])**2
      if ed < mnDis:
        mnDis=ed
        BelongToCluster[labels]=clusters

  #get new centers
  for centers in range(0,k):
    total=0
    points=[0]*len(features[0])
    for labels in range(0,len(features)):
      if BelongToCluster[labels] == centers:
        total+=1
        for dim in range(0,len(features[0])):
          points[dim]+=features[labels][dim]
    if total>0:
      for dim in range(0,len(features[0])):
        points[dim]/=total
      CentersOfCluster[centers]=points

  #get distortion after each iteration
  for cluster in range(0,k):
for it in range(0,iterations) › for labels in range(0,len(featu...
SciView
Database
```



```
3:06
untitled1 [~/PycharmProjects/untitled1] - .../Clustering.py [untitled1] - PyCharm
ndow Help
Clustering
ustering.py
  #get distortion after each iteration
  for cluster in range(0,k):
      total = 0
      points = [0] * len(features[0])
      for labels in range(0, len(features)):
          if BelongToCluster[labels] == cluster:
              total += 1
              for dim in range(0, len(features[0])):
                  points[dim] += features[labels][dim]
      if total > 0:
          for dim in range(0, len(features[0])):
              points[dim] /= total
          for label in range(0,len(features)):
              if BelongToCluster[label] == cluster:
                  for dim in range(0, len(features[labels])):
                      dist += (CentersOfCluster[cluster][dim] - features[label][dim]) ** 2
  distortion.append(dist)

plt.plot(distortion)
plt.xlabel("iterations")
plt.ylabel("summation of diff. between each object and nearest cluster to it")
plt.show()
SciView
Database
```

1) K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.

Accuracy:

As we use clustering algorithm we cant calculate accuracy according to the labels of the image because cluster is unsupervised learning.

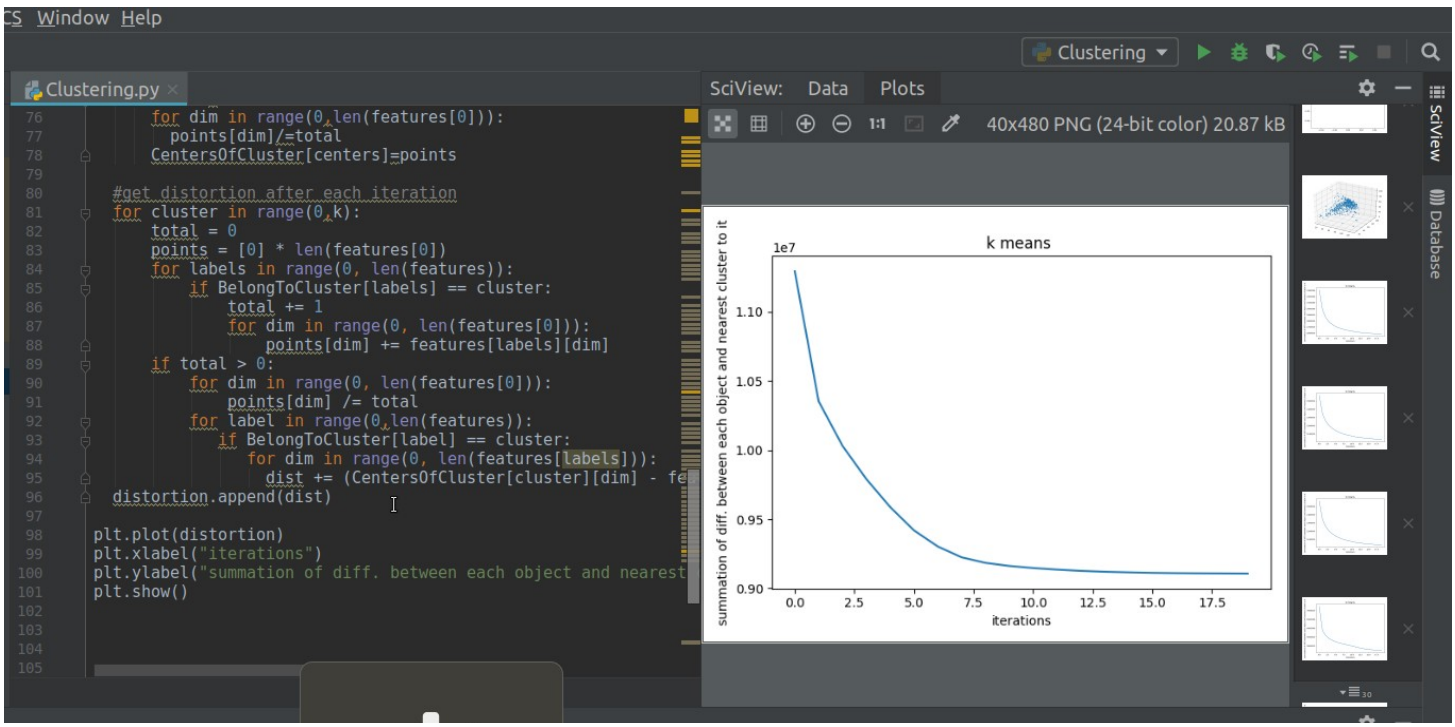
So we calculate our accuracy according to the distortion Function as we aim to make it tends to Zero.

What is Distortion Function?

It is how long our image far away from it's nearest Cluster's Center. So we get summation for this Distance at each iteration.

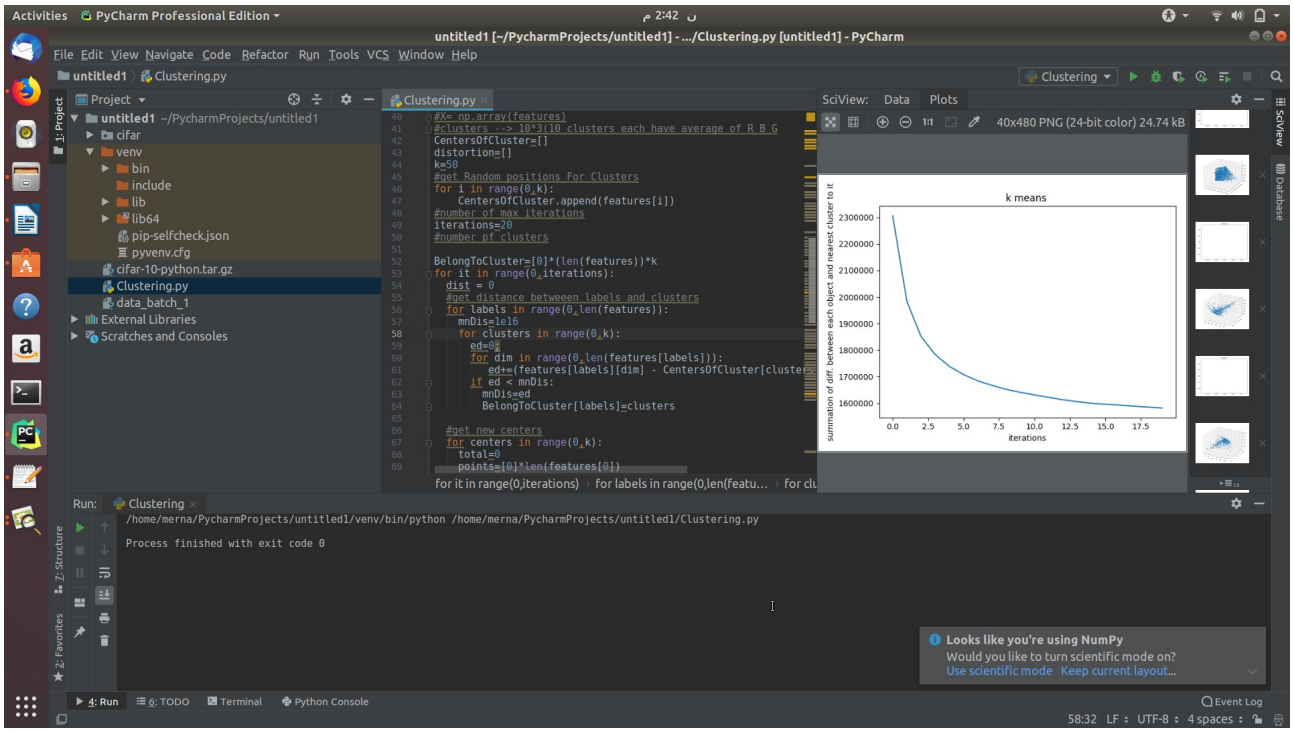
It can be represented using variance Function.

At k=10:



- it’s start to get near to x-axis at iteration:7 so it get to be stable and more accurate after the 7th iteration

  At k=50:



- it get closer to x-axis at iteration(13) so it get to be stable and more accurate after 13 iterations.

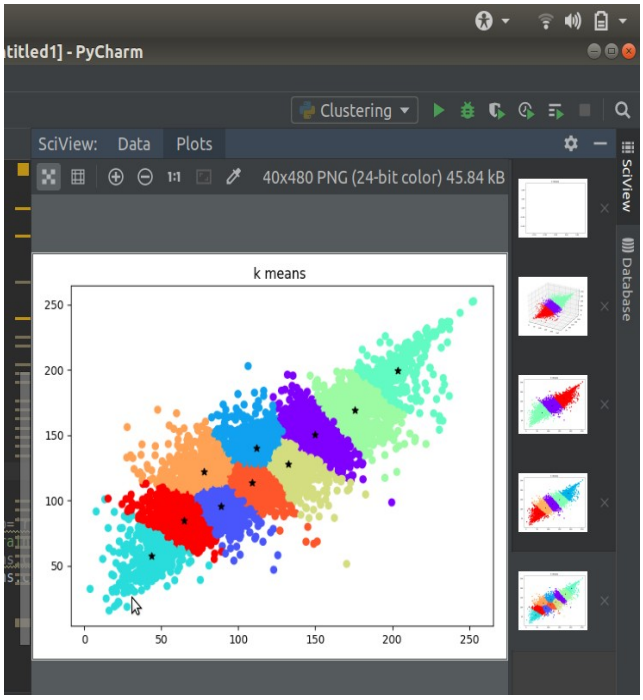
NOTE:

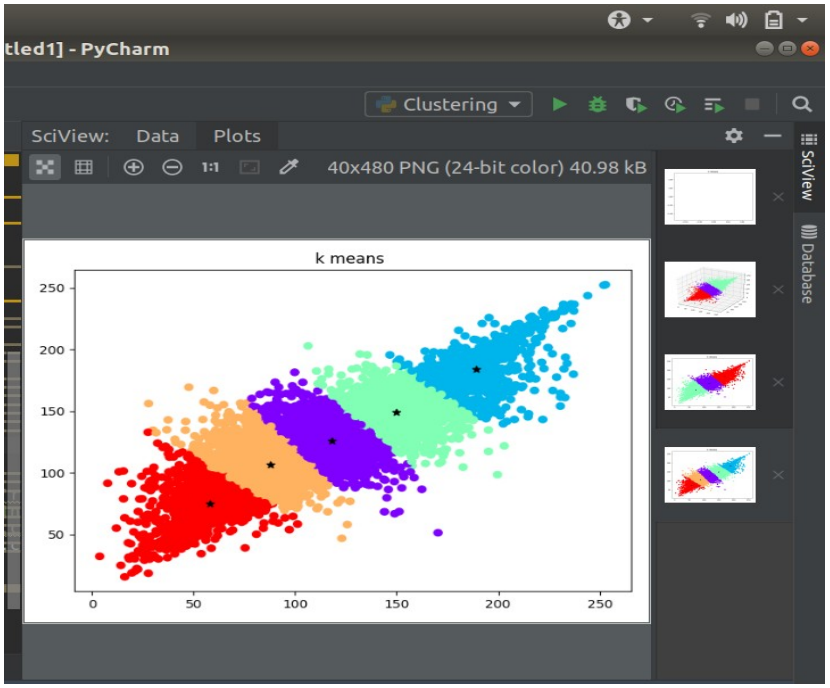
if we increase the number of clusters more than the normal as k=50 our function become less accuracy as each cluster has smaller amount of points belong to it and also if we make number of clusters so small as k=2 our function also become less accuracy as the distance increase.

2)Are the results wildly different for different restarts and/or different K?

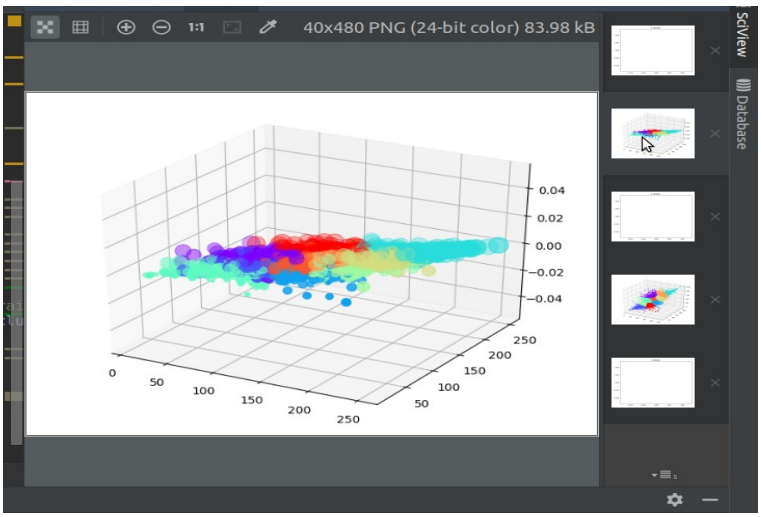
Yes, as we change our K accuracy Change.
Plotting Of clusters at difference Start and difference K .
For Difference Restart: As K means get local maximum if we reach that every point belong to same cluster every iteration and nothing change, that doesn’t mean that we reach the best case we have to start randomly and different restart to check for the best case.

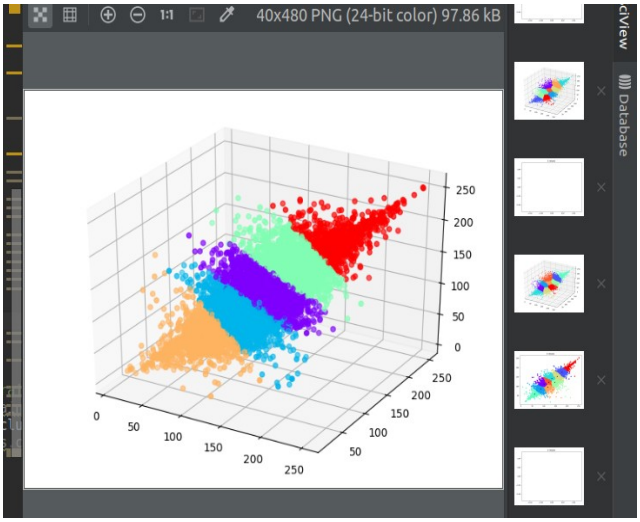
plotting if we choose 2 features only (average of blue color,average of green color).
Black points are our cluster’s Center.



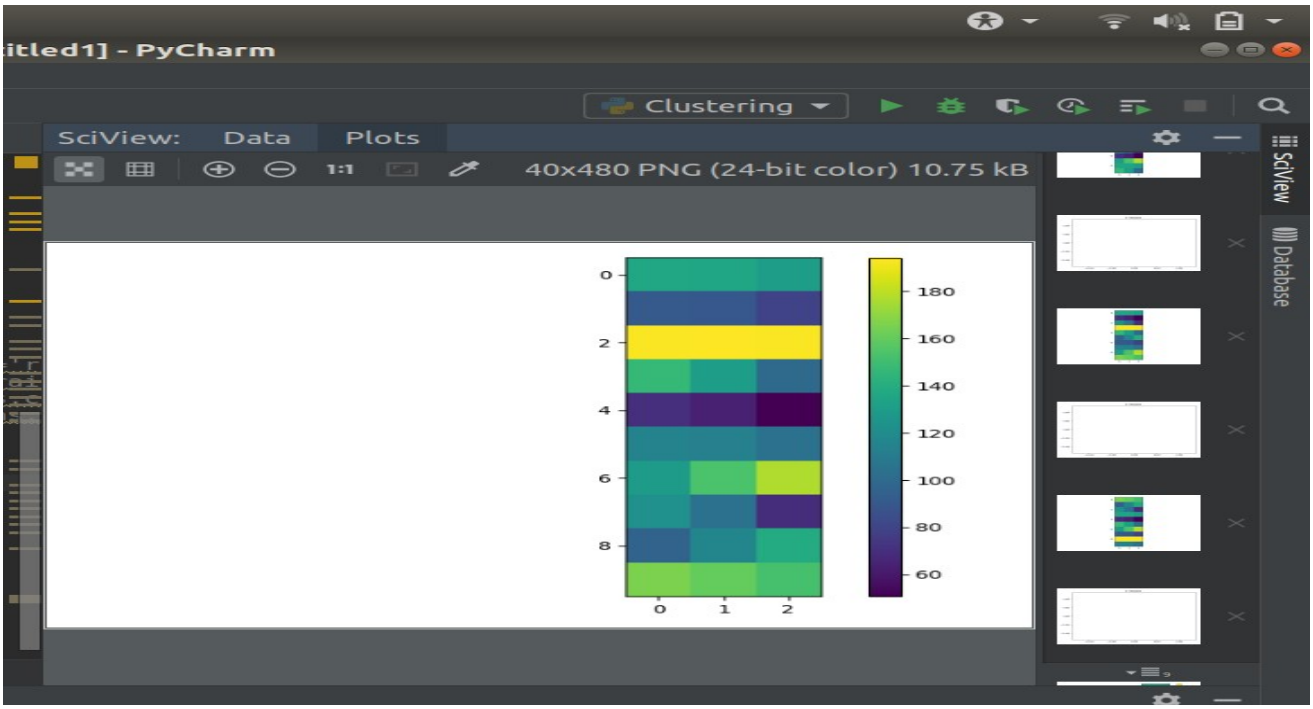


Plotting if we choose 3 features (average of red color, average of blue color,average of green color).


1:1
40x480 PNG (24-bit color) 83.98 kB
SciView
Database
0.04
0.02
0.00
−0.02
−0.04
0
50
100
150
200
250


1:1
40x480 PNG (24-bit color) 97.86 kB
SciView
Database
250
200
150
100
50
0

3)What do the mean images look like? Provide screenshots with comments.

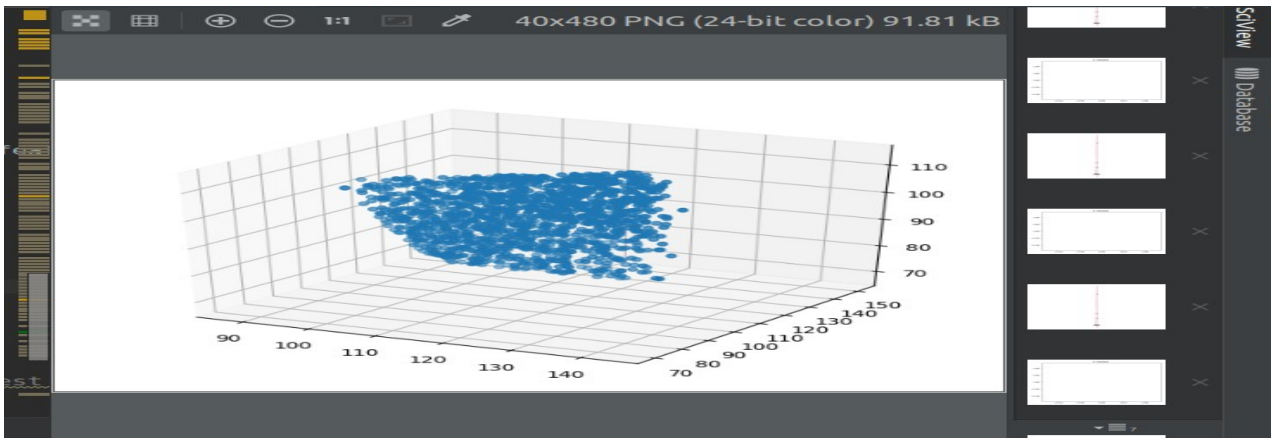


Note:

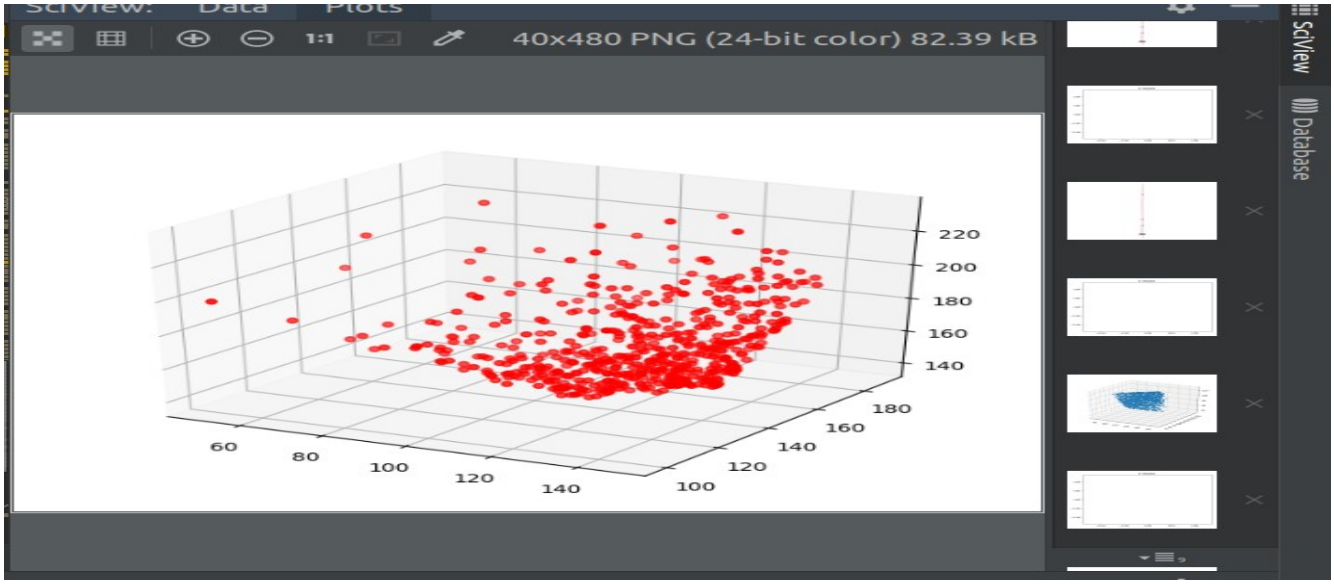
- K means get accuracy 40% with Cifar dataset as Cifar need deepLearning to get accuracy 90%.
- So it doesnt retrieve accurate images.

4)What are some representative images from each of the clusters?

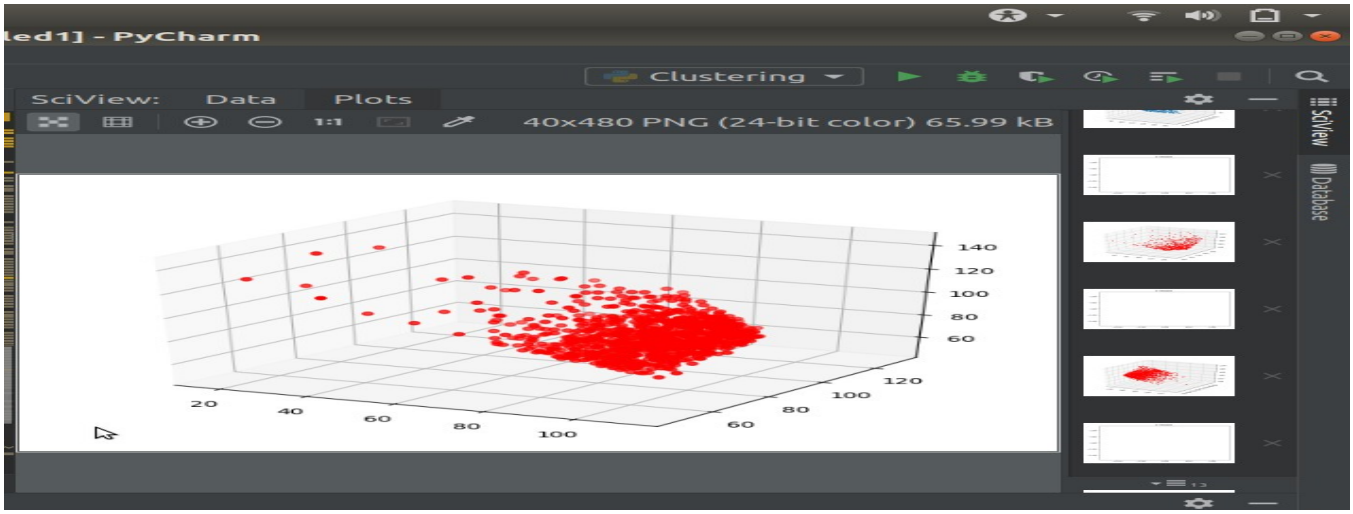
Cluster0:



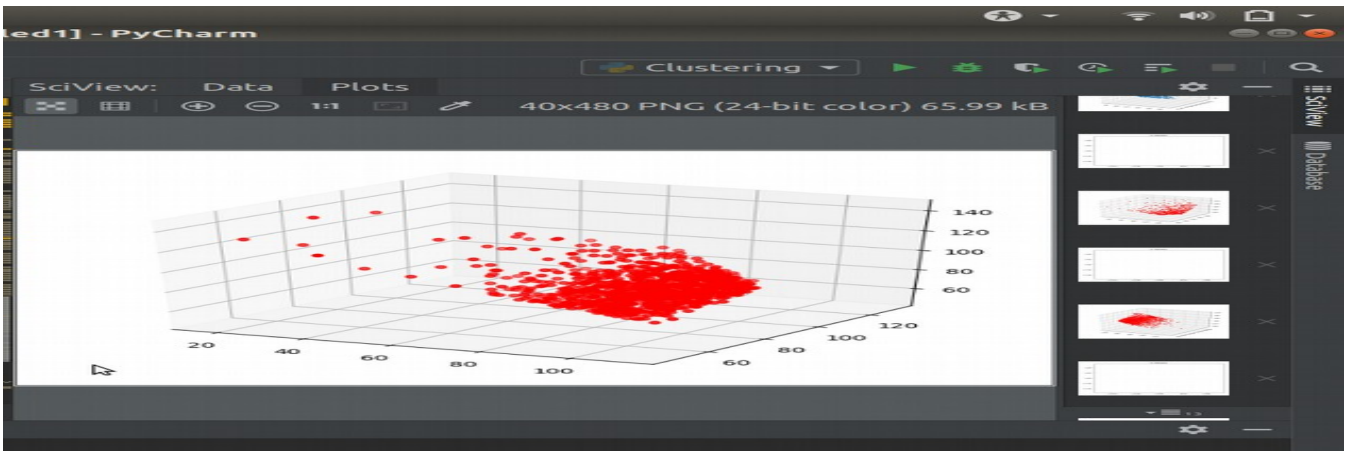
Cluster1:



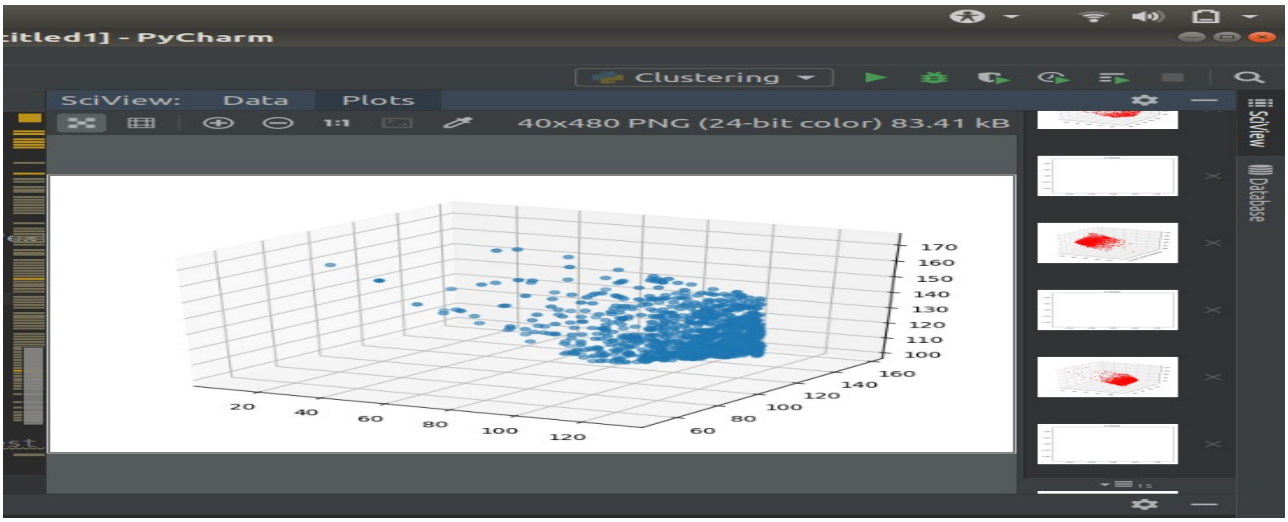
Cluster2:



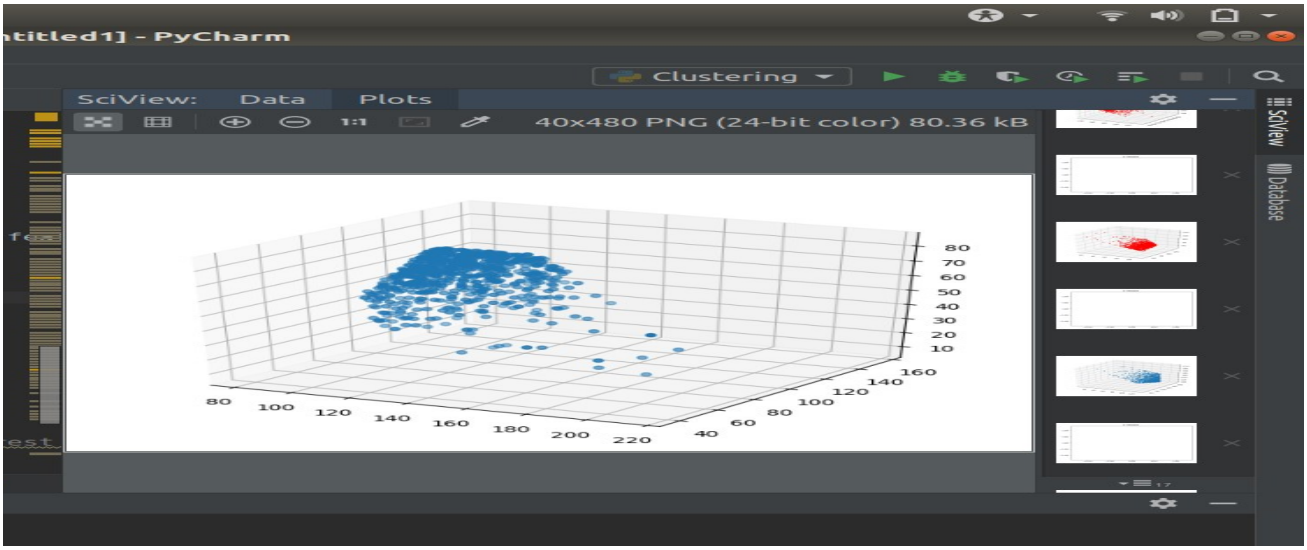
Cluster3:



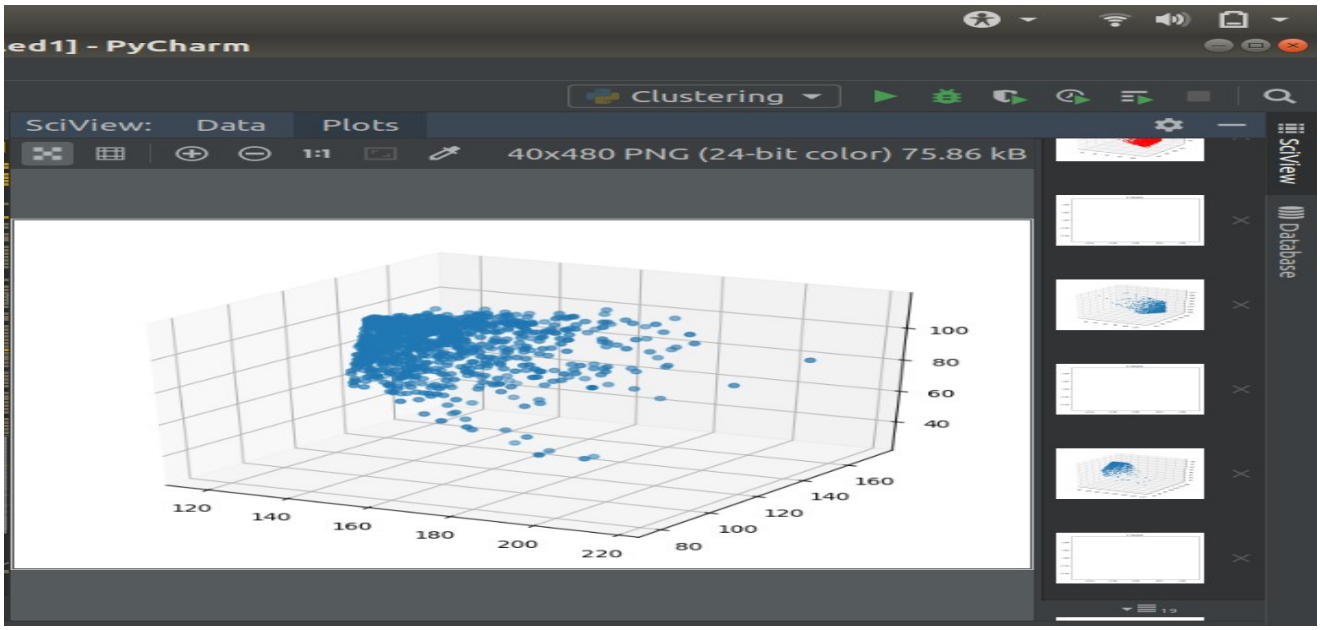
Cluster4:



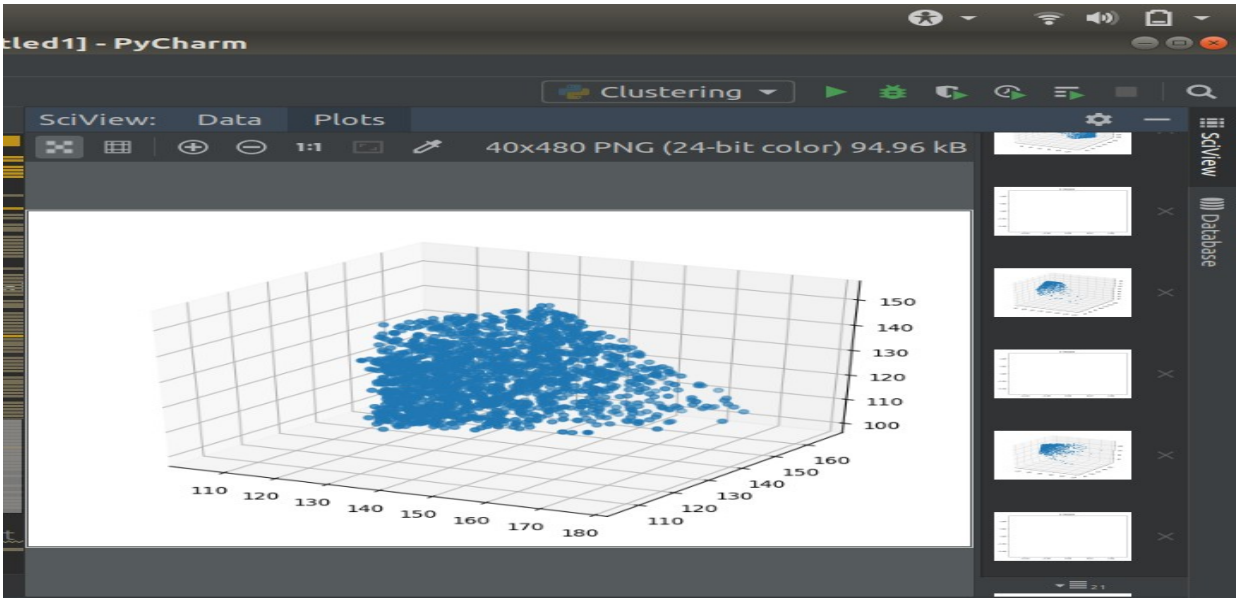
Cluster 5:



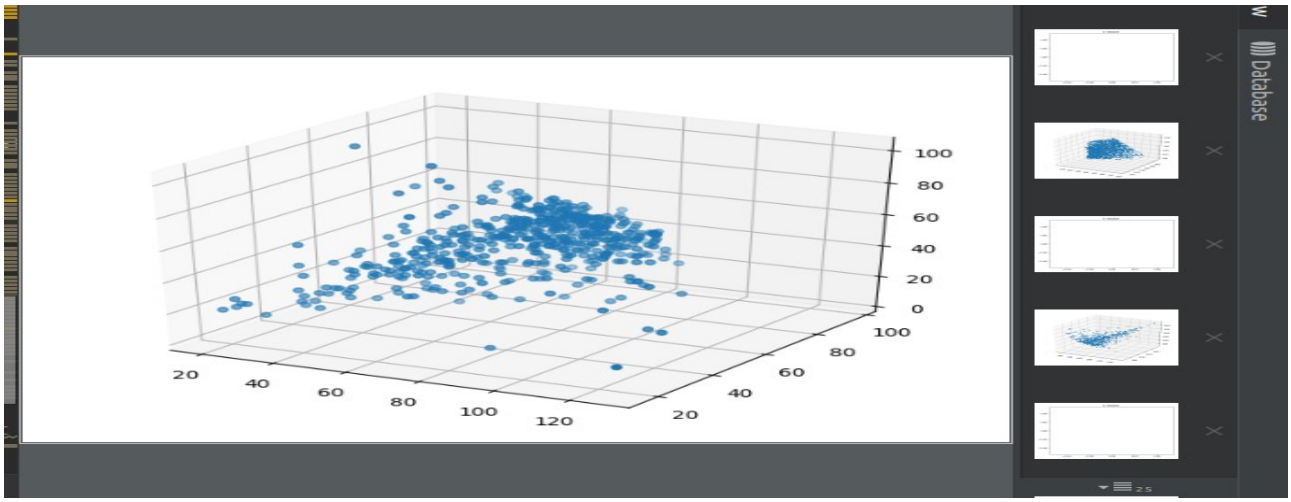
Cluster 6:



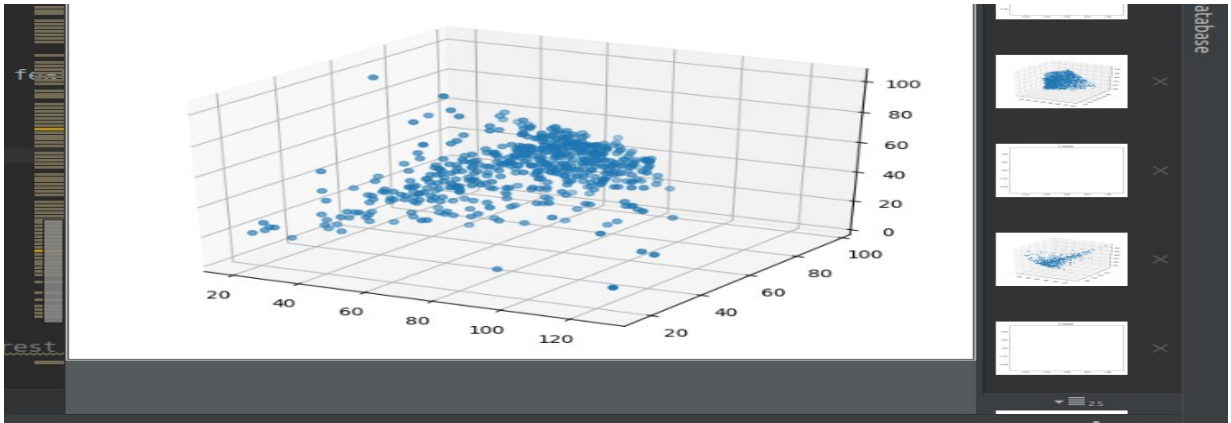
Cluster 7:



Cluster 8:

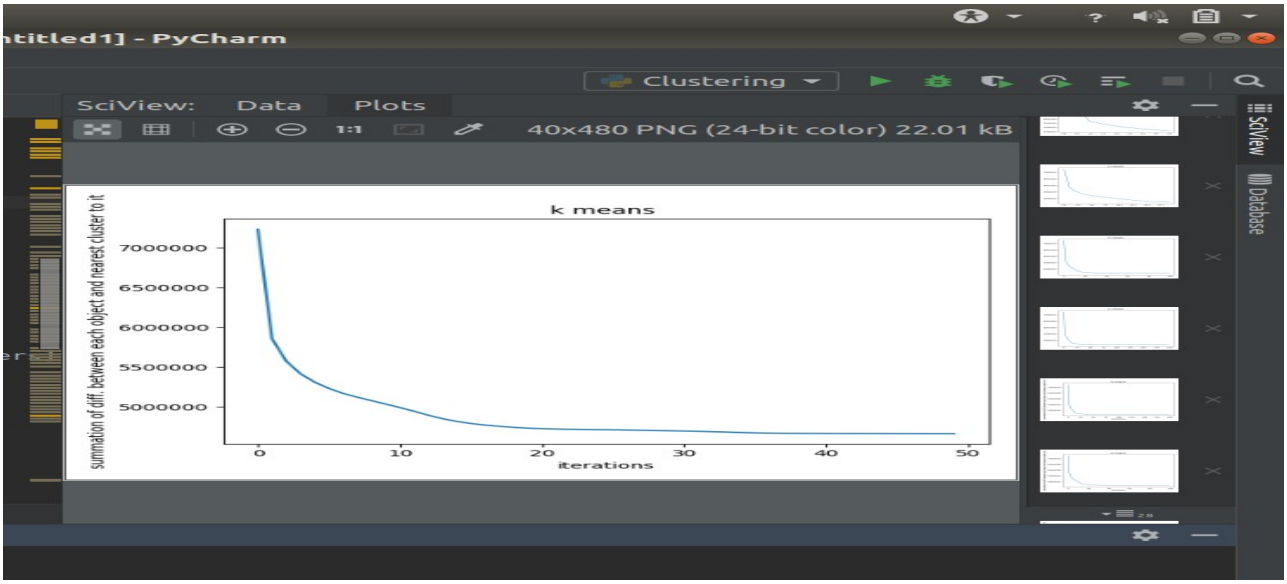


Cluster 9:

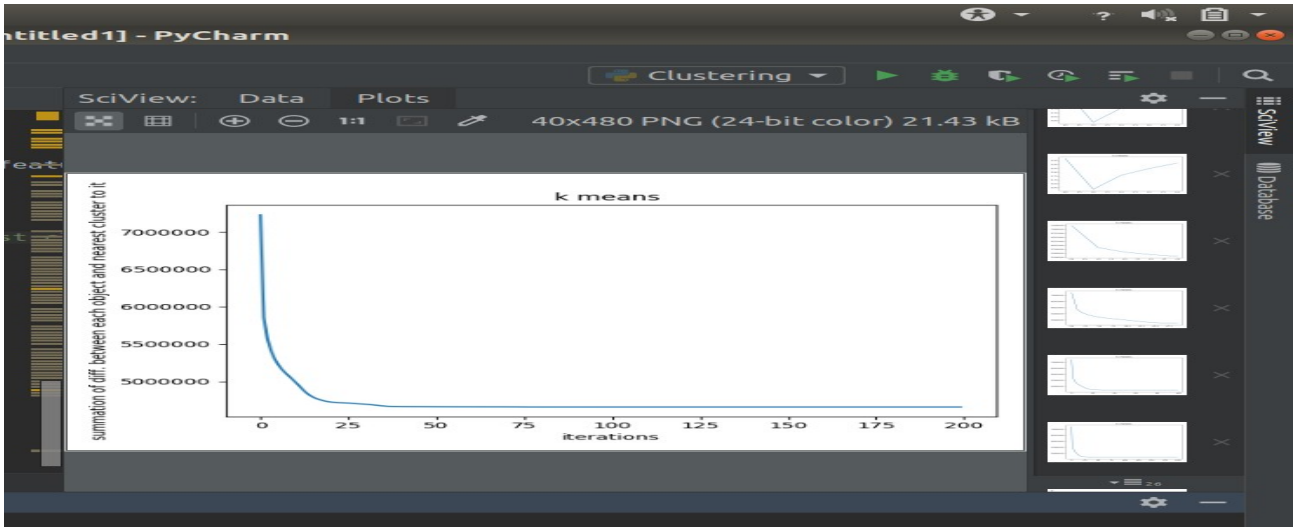


5)Plot the K-Means objective function (distortion measure) as a function of iteration and verify that it never increases.

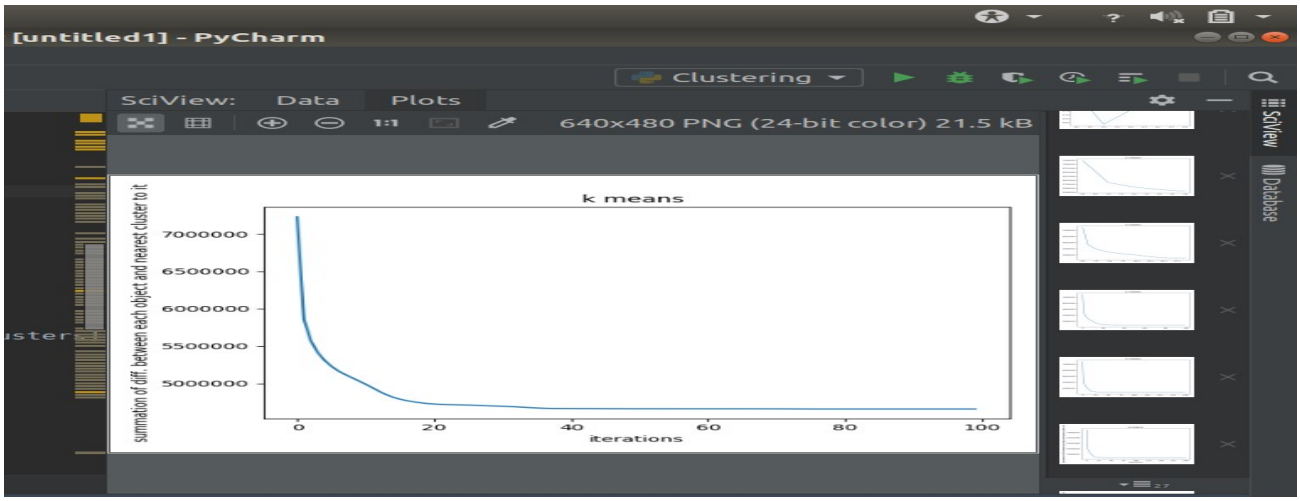
Iteration = 50 , k =10:



Iteration = 200, k=10:



Iteration = 100 , k= 10:



Iteration = 100, k= 3 :

Iteration = 50, k = 5: