PATTERN RECOGNITION AND MACHINE LEARNING

# LAB 09

# LAB REPORT

## QUESTION 1.

### 1)

- PCA was chosen as the dimensional reduction method.
- n_components was chosen to be 4.

### 2)

- KMeans was implemented through SKLearn.
- n_clusters was chosen to be 3, by observing the dataset visually.

### 3)

- A plot was plotted for different values of k for visualization.
- The best value of k was sorted out through maximum Silhouette Score.
- The optimal value of k was found to be 3, which matches with visual observation.

### 4)

- Sum of Squared Error was used as the parameter for finding the optimal value of k.
- Similar method as of the 3rd part was used to find the optimum value and the value was found to be 3.

# QUESTION 2.

## 1) 2)

- KMeans clustering was implemented from scratch.
- It is able to:
- Store the cluster centers.
- Take a value of k from users to give k clusters.
- Take initial cluster center points from the user as its initialization.
- Stop iterating when it converges (cluster centers are not changing anymore) or, a maximum
- iteration (given as max_iter by user) is reached.

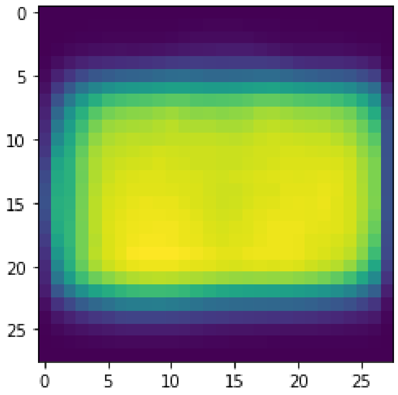
## 3)

- The KMeans Clustering model was applied on the dataset with 10 clusters.
- The number of points in each cluster was reported to be 3762, 7786, 2373, 5162, 7379, 7908, 9561, 7624, 2578, 5867.

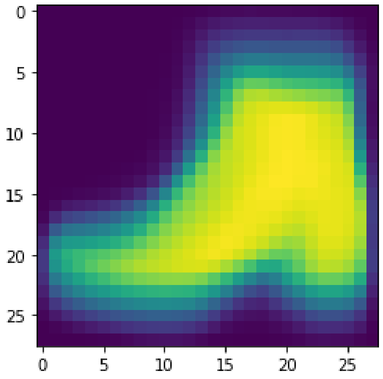
## 4)

- The averages of all clusters depicting the centroids of corresponding clusters were converted to 28 x 28 images and were shown.
- Visualization of centroids of each corresponding cluster gives a clear idea about the contents of the cluster (eg bag, shirt, etc)

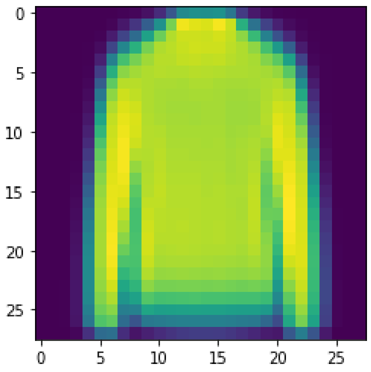
Cluster Center: 0



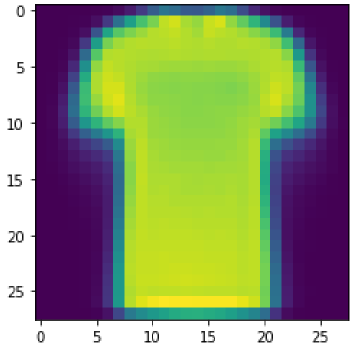
Cluster Center: 1



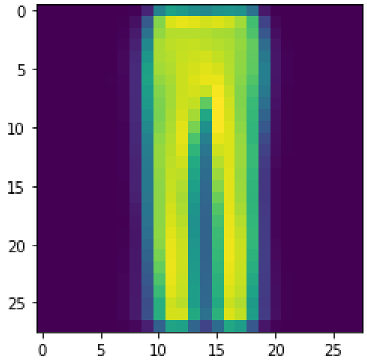
Cluster Center: 2



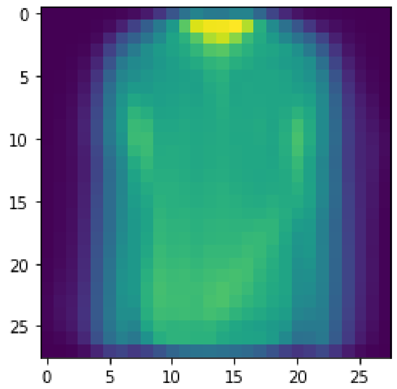
Cluster Center: 3



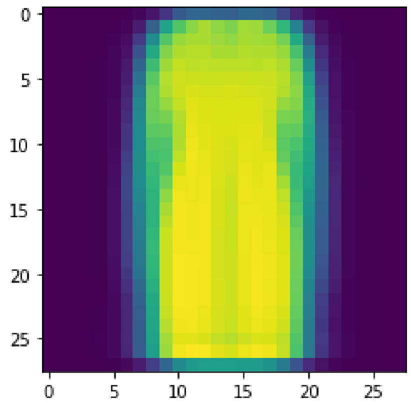
Cluster Center: 4



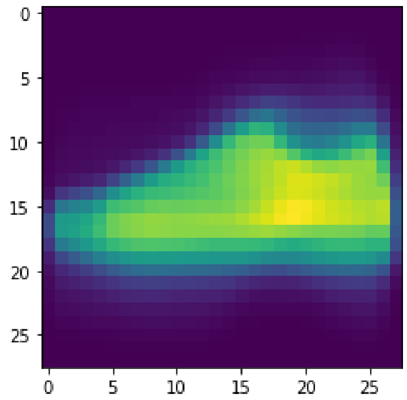
Cluster Center: 5



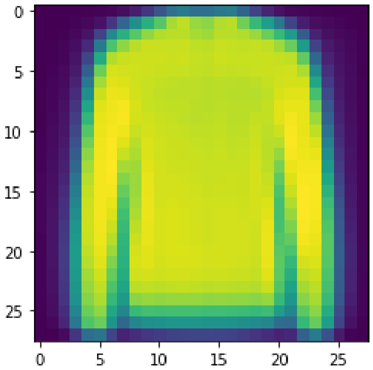
Cluster Center: 6



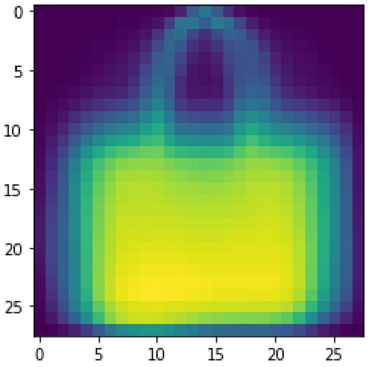
Cluster Center: 7



Cluster Center: 8



Cluster Center: 9

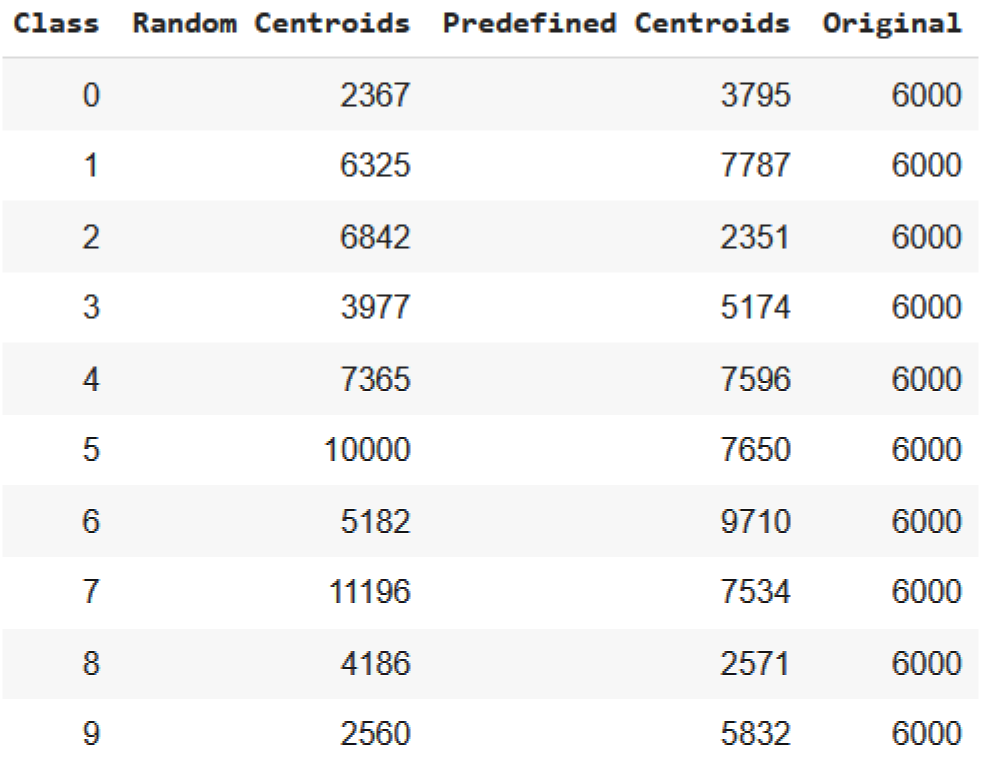


**5)**

- 10 images from each of the clusters were shown in Colab, with a total of 100 images.
- Each image was observed to nearly follow the same pattern as depicted by its corresponding centroid.

**6)**

- Centroids were now defined initially with each centroid lying in the same defined cluster, with the help of labels in the dataset.
- The number of elements in each cluster was found to be
- 3889
- 7495
- 2374
- 5373
- 6522
- 8638
- 8961
- 8265
- 2557
- 5926



| Class | Random Centroids | Predefined Centroids | Original |
|---|---|---|---|
| 0 | 2367 | 3795 | 6000 |
| 1 | 6325 | 7787 | 6000 |
| 2 | 6842 | 2351 | 6000 |
| 3 | 3977 | 5174 | 6000 |
| 4 | 7365 | 7596 | 6000 |
| 5 | 10000 | 7650 | 6000 |
| 6 | 5182 | 9710 | 6000 |
| 7 | 11196 | 7534 | 6000 |
| 8 | 4186 | 2571 | 6000 |
| 9 | 2560 | 5832 | 6000 |

## 7)

- 10 images were visualized corresponding to each cluster, with a total of 100 images.

## 8)

- `SSE for Random Centroids:     9808494422.876373`
- `SSE for Predefined Centroids: 9813398721.11784`
- SSE for both the initialization technique was found to be nearly the same. This is due to the fact that input initial centroids were actually random and present in nearly all classes, and hence results matched with predefined centroid case.

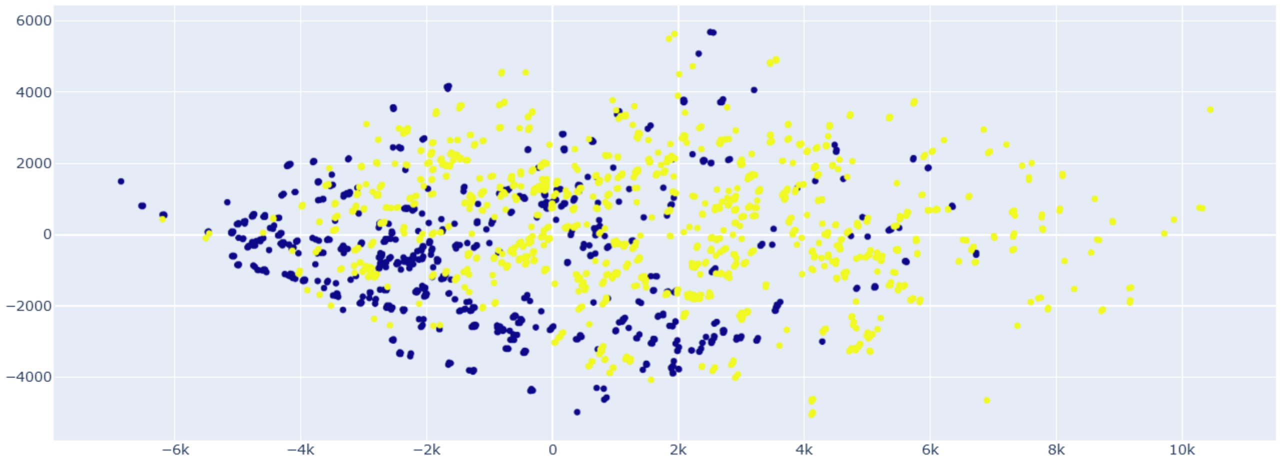
# QUESTION 3.

## 1)

- The images were first converted to RGB type.
- Then they were resized to 64 x 64.
- The Images were converted to arrays.
- The arrays were again resized to 12288 (64 * 64 *3).

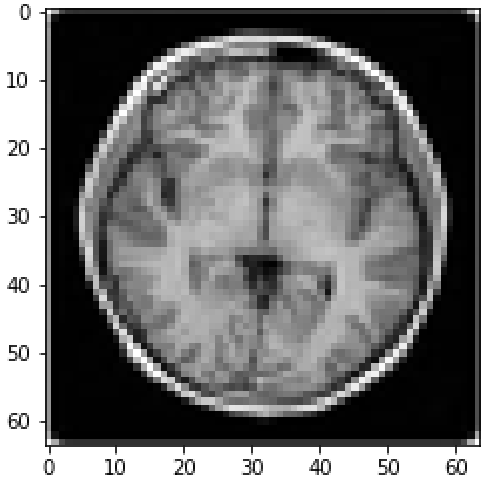
## 2)

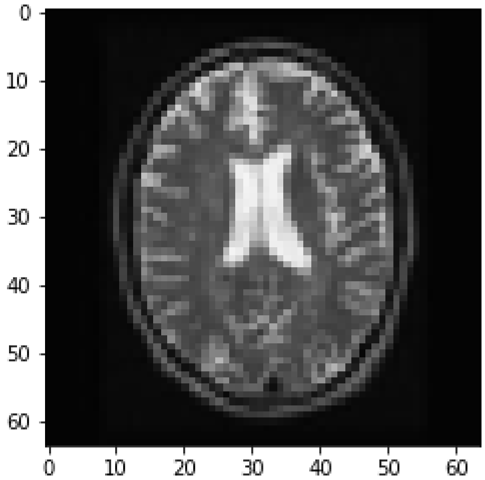
- PCA was used as the dimension reduction technique with n_components = 2.

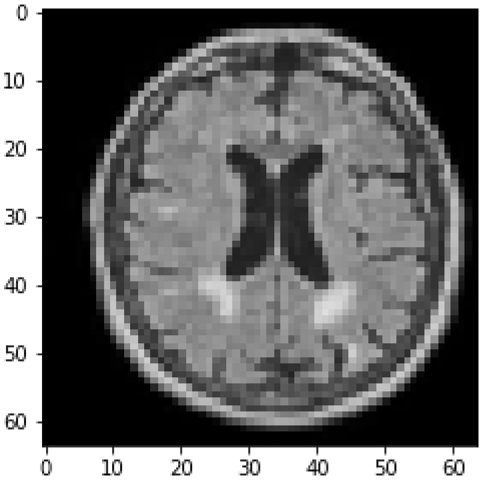


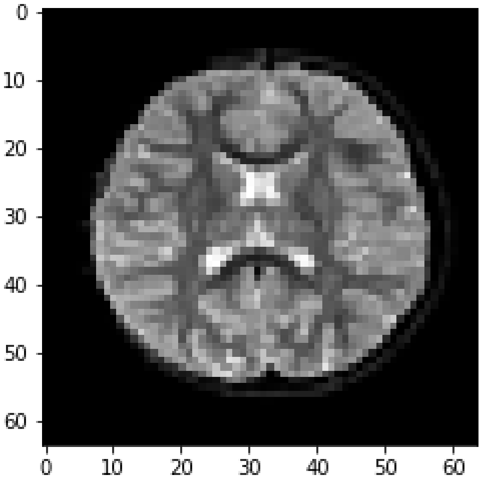
### 3)

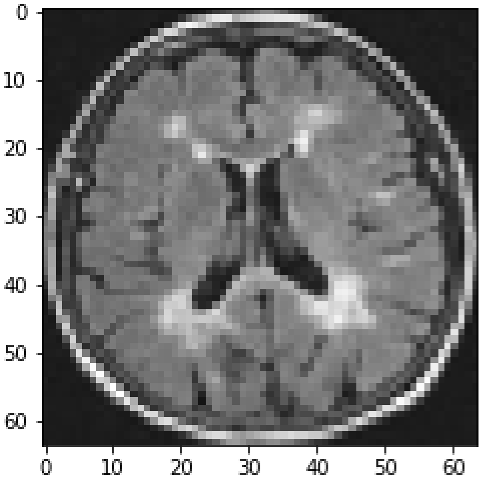
***NO***

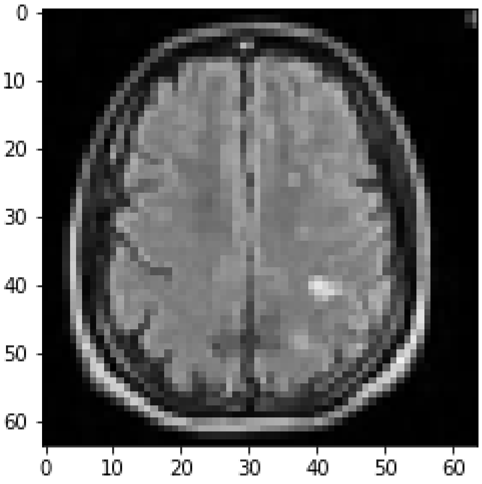

0
10
20
30
40
50
60
0
10
20
30
40
50
60

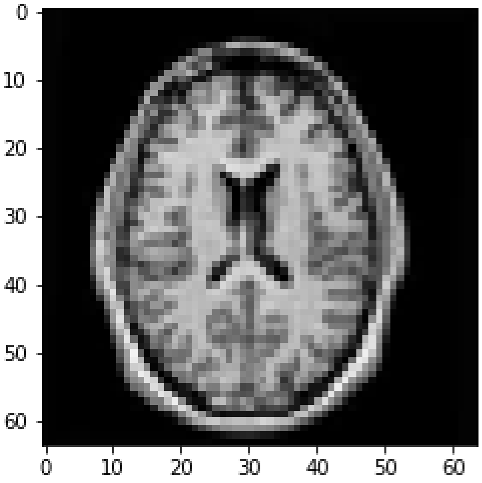

0
10
20
30
40
50
60
0
10
20
30
40
50
60

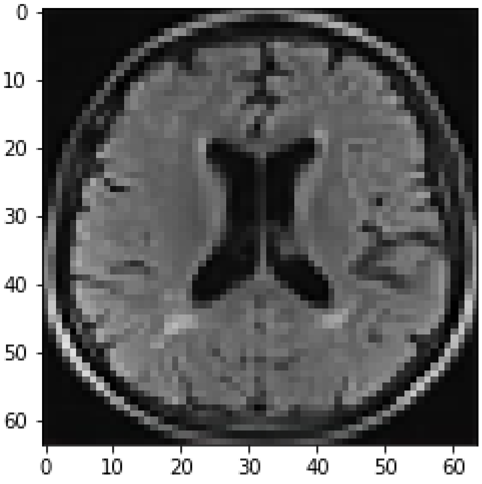

0
10
20
30
40
50
60
0
10
20
30
40
50
60

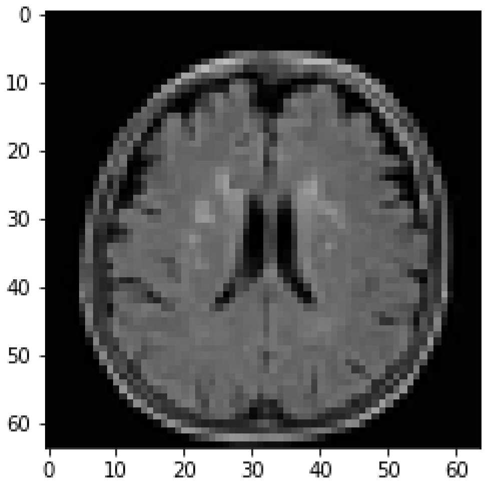

0
10
20
30
40
50
60
0
10
20
30
40
50
60

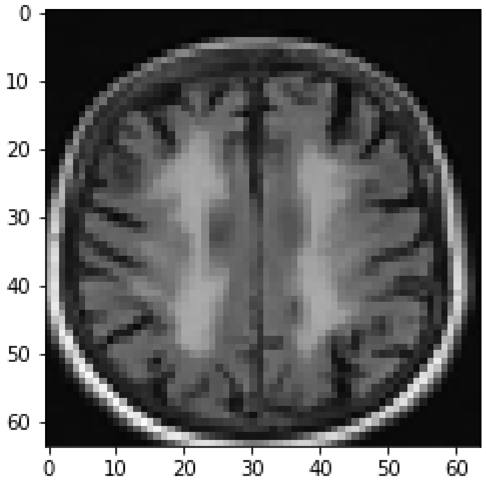

0
10
20
30
40
50
60
0
10
20
30
40
50
60


0
10
20
30
40
50
60
0
10
20
30
40
50
60

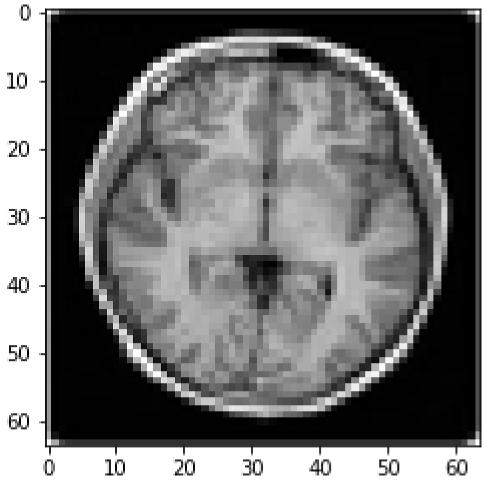

0
10
20
30
40
50
60
0
10
20
30
40
50
60

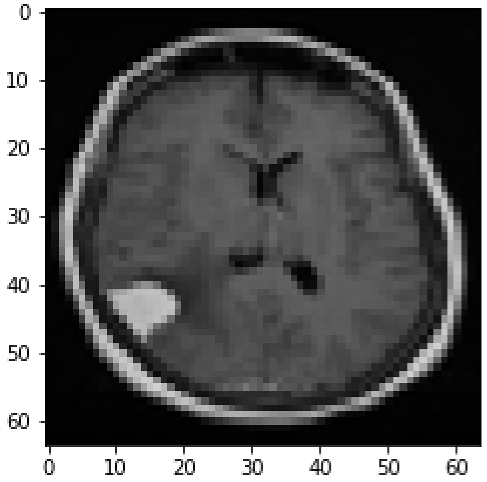

0
10
20
30
40
50
60
0
10
20
30
40
50
60

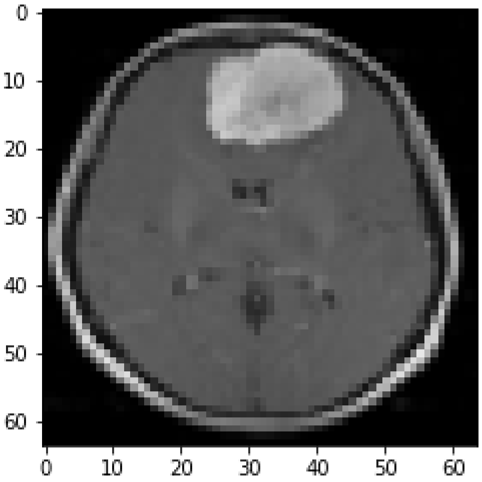

0
10
20
30
40
50
60
0
10
20
30
40
50
60

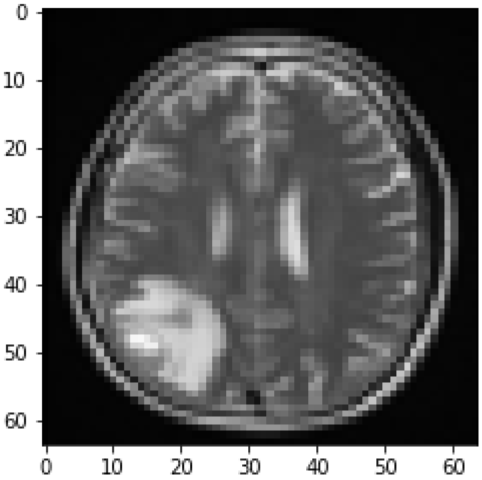

0
10
20
30
40
50
60
0
10
20
30
40
50
60

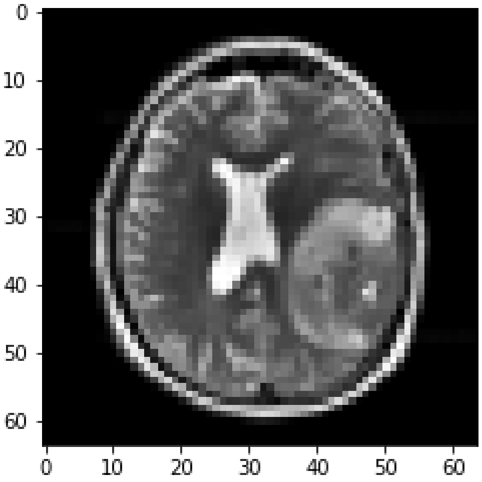
***YES***

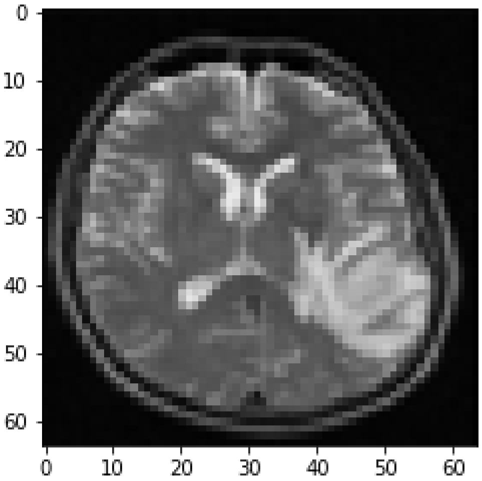

0
10
20
30
40
50
60
0
10
20
30
40
50
60

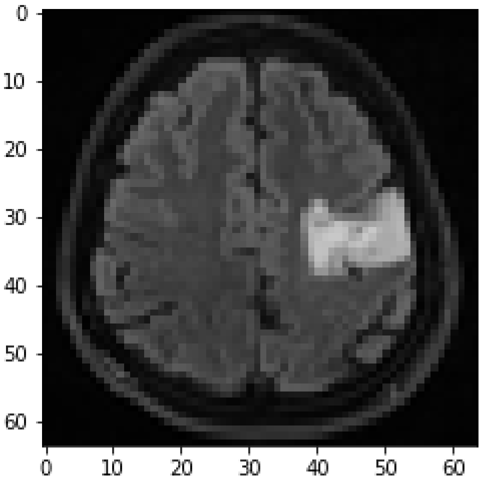

0
10
20
30
40
50
60
0
10
20
30
40
50
60

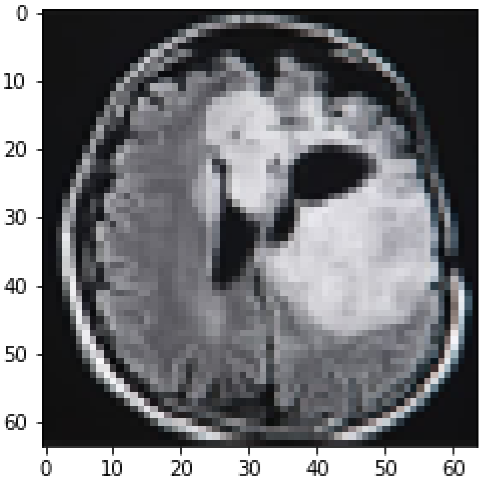

0
10
20
30
40
50
60
0
10
20
30
40
50
60

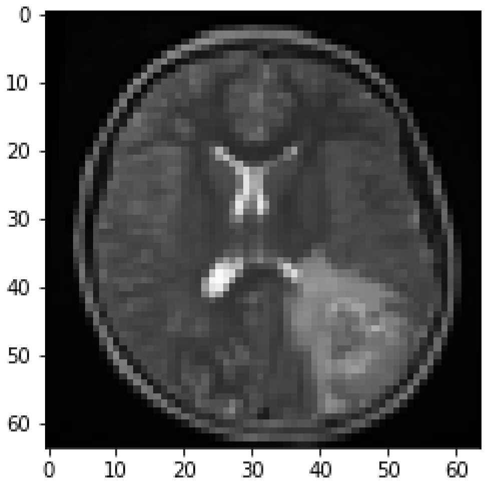

0
10
20
30
40
50
60
0
10
20
30
40
50
60

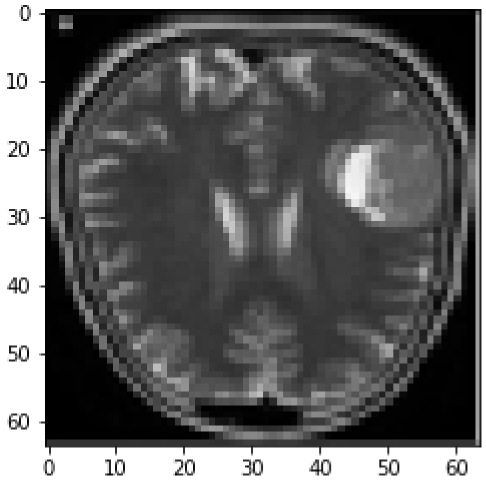

0
10
20
30
40
50
60
0
10
20
30
40
50
60


0
10
20
30
40
50
60
0
10
20
30
40
50
60


0
10
20
30
40
50
60
0
10
20
30
40
50
60

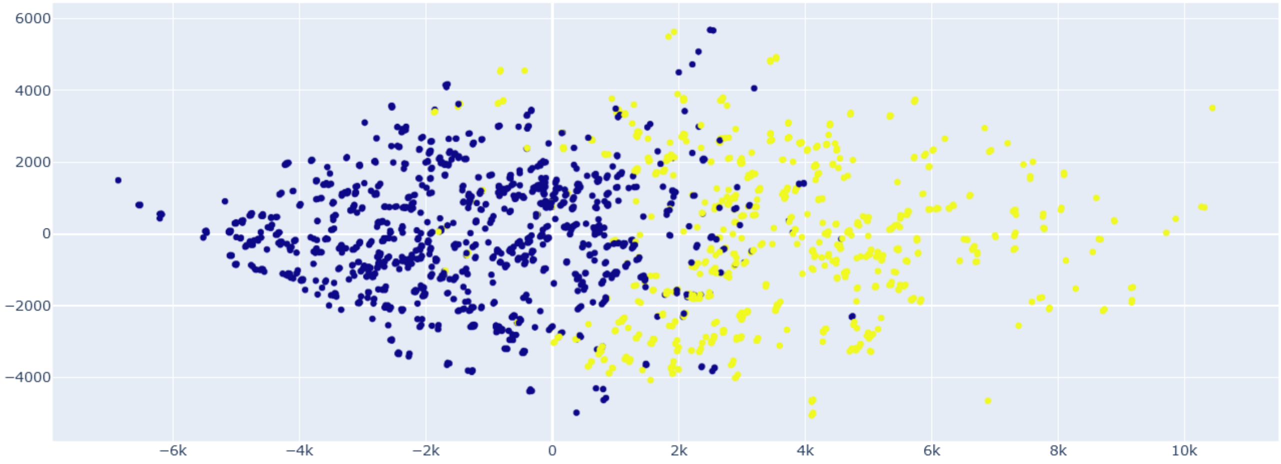

0
10
20
30
40
50
60
0
10
20
30
40
50
60


0
10
20
30
40
50
60
0
10
20
30
40
50
60


0
10
20
30
40
50
60
0
10
20
30
40
50
60

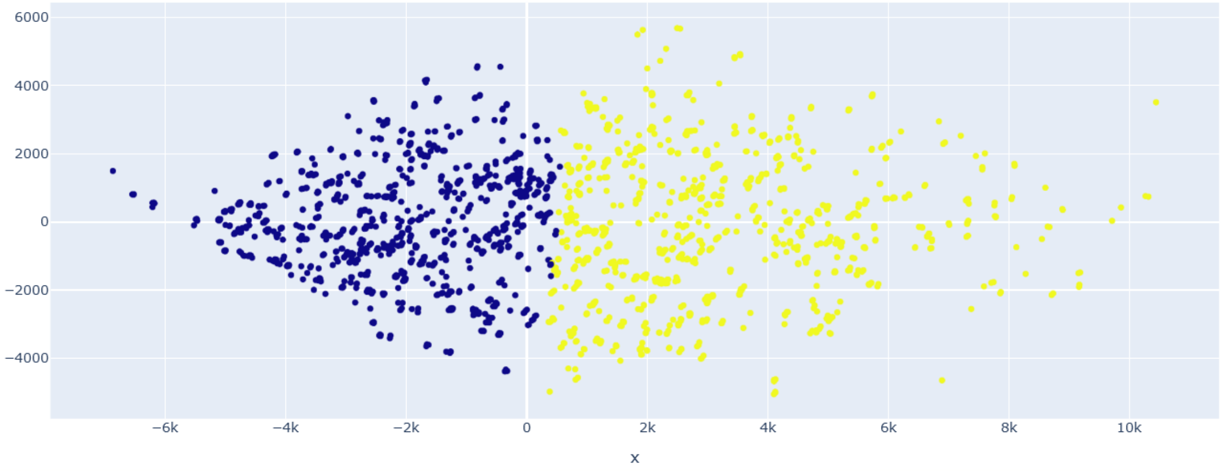
## 4)

- AgglomerativeClustering was applied through SKLearn with n_clusters



## 5)

- KMeans clustering was applied on the data and following was the visualization:



- Accuracy by AgglomerativeClustering: 0.6323676323676324
- Accuracy by KMeansClustering: 0.6636696636696636