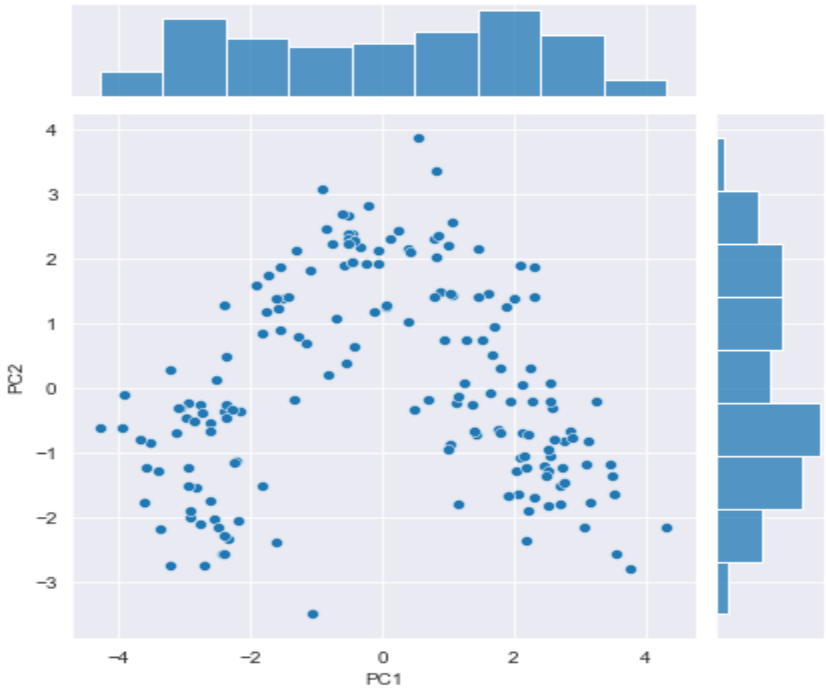
Pattern Recognition and Machine Learning 2022 Winter Semester

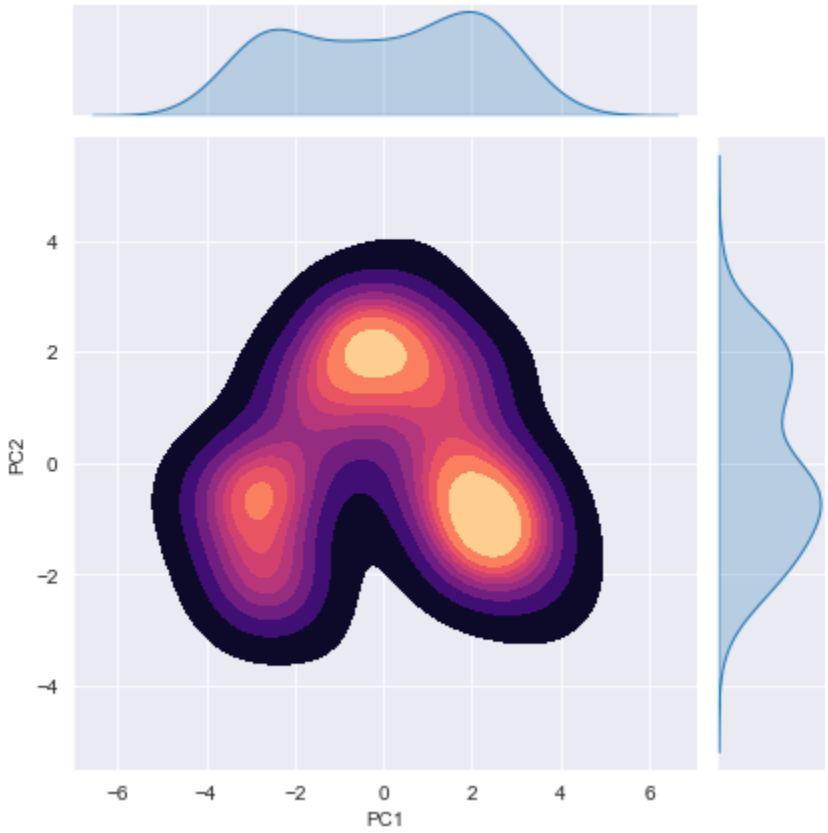
Assignment 8

Question 1 - Unsupervised learning algorithm

1. The given ‘Wine Quality Dataset’ is imported and read in as a pandas data frame along with appropriate column names obtained from the given link. The ‘Class’ target is 0-indexed. Statistics of the data are analyzed via ‘.head()’, ‘.info()’, and ‘.describe()’. Following observations are made: There are thirteen(13) features and one hundred and seventy-seven (177) data points. Only continuous features are available. The scale of continuous features is different. No missing (NaN) values. The dataset is split based on ‘features’ and ‘target’. *Heatmap* and *pair plot* can be found in the notebook itself. Preprocessing involved standard scaling to get the mean of the data as zero and the standard deviation as one. Principle component analysis (PCA) dimension reduction technique with ‘n_components = 2’ (for two-dimensional data) is used. Visualization of the data after implementing the mentioned technique is as follows:
   Scatter Plot:



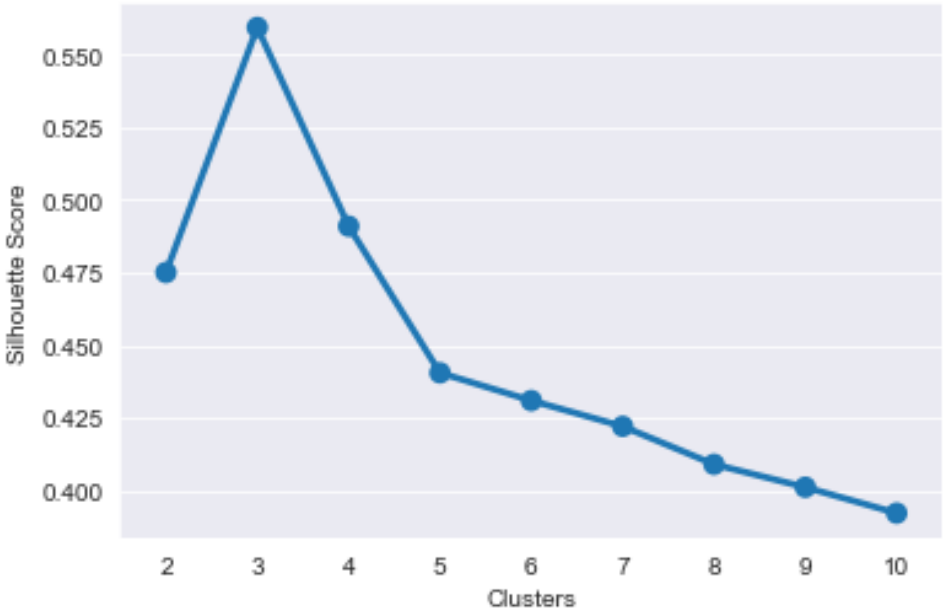
KDE Plot:



By looking at the plot, it can be observed that ‘k = 3’ will be the best-suited value. This can be justified by the fact that we can observe three(3) different clusters (upper half, lower left, and lower right) in the plots.

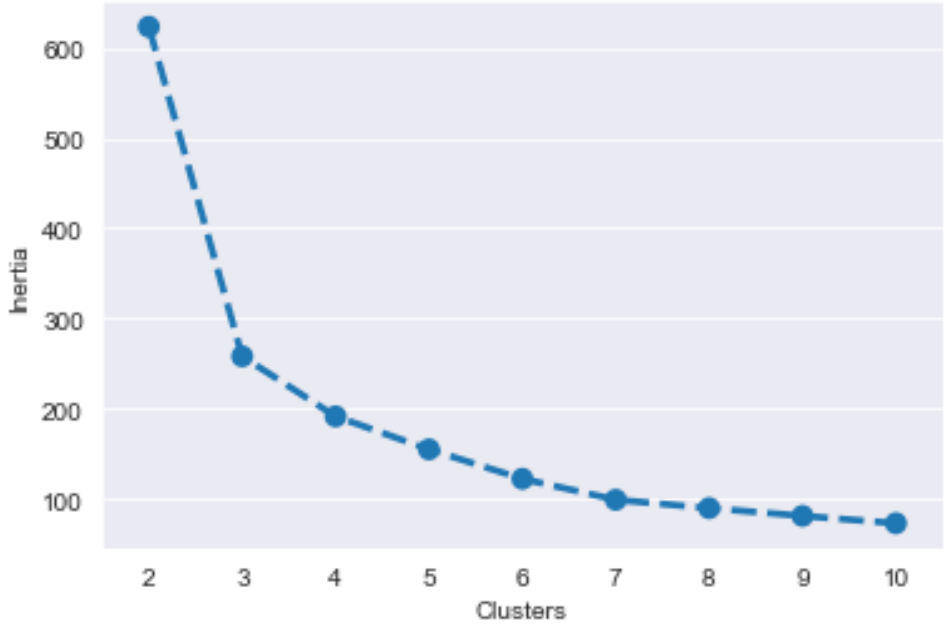
2. A k-means clustering algorithm (using sklearn library) is built and implemented using ‘k = 3’, as mentioned in Subpart - 1. Visualization of clusters along with the centroids is as follows:

3. Different values of ‘k’ are used to find the respective ‘Silhouette Score’. The results are plotted below:

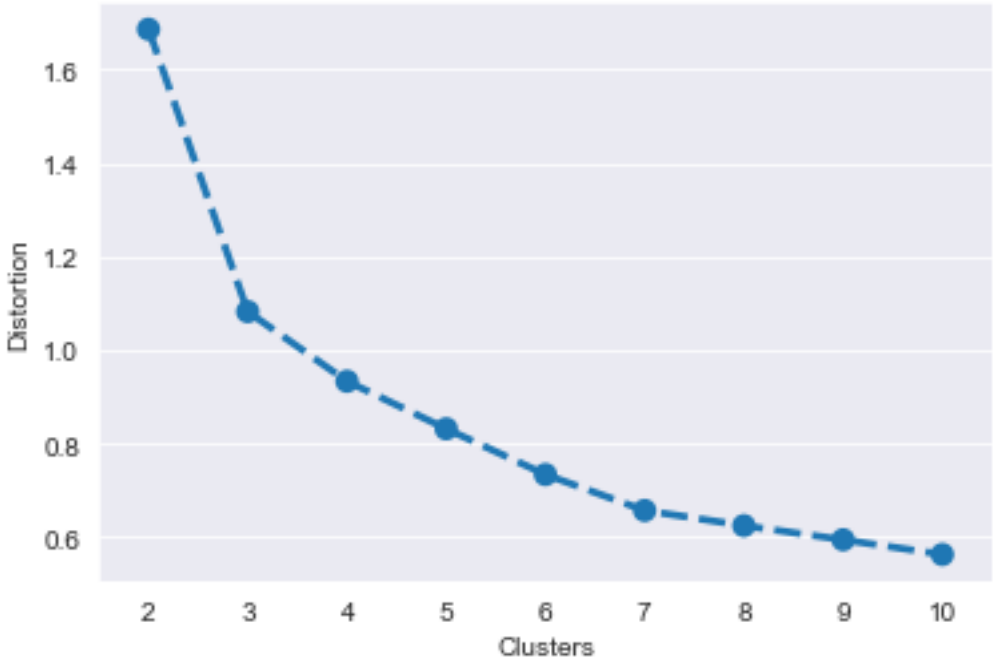


The Silhouette Score of one(1) means that the clusters are very dense and nicely separated. A score of zero(0) means that clusters are overlapping. A score of less than zero(0) means that the data belonging to clusters may be wrong/incorrect. Considering the maximum silhouette score is observed at k = 3, hence it is the optimal value of the same.

4. Elbow Method is used to find the optimal value of k. The plot is as follows: Inertia based:



Distortion based:



The intuition is that increasing the number of clusters will naturally improve the fit (explain more of the variation) since there are more parameters (more clusters) to use, but at some point this is over-fitting, and the elbow reflects this. The ‘elbow’ occurs at ‘k = 3’. That is, the decrease in slope becomes less rapid at k = 3. Hence, as per the ‘Elbow Method’, ‘k = 3’ is the optimal value.

Question 2 - Kmeans from Scratch

1. The given ‘fashion-MNIST dataset’ is imported and read in as a pandas data frame along with appropriate column names. Statistics of the data are analyzed via ‘.head()’, ‘.info()’, and ‘.describe()’. Following observations are made:
   There are 784 features and 60000 data points. No missing (NaN) values. The dataset is split based on ‘features’ and ‘target’. The ‘features’ were divided by 255 to make all the entries range from 0 to 1. KMeans clustering algorithm is implemented from scratch. The class takes as input the following parameters:
   centroids: initial cluster center points from the user as its initializations.
   clusters: a value ‘k’ from the users to give ‘k’ clusters.
   tolerance: value below which fit loop breaks
   And others can be found in the colab notebook.

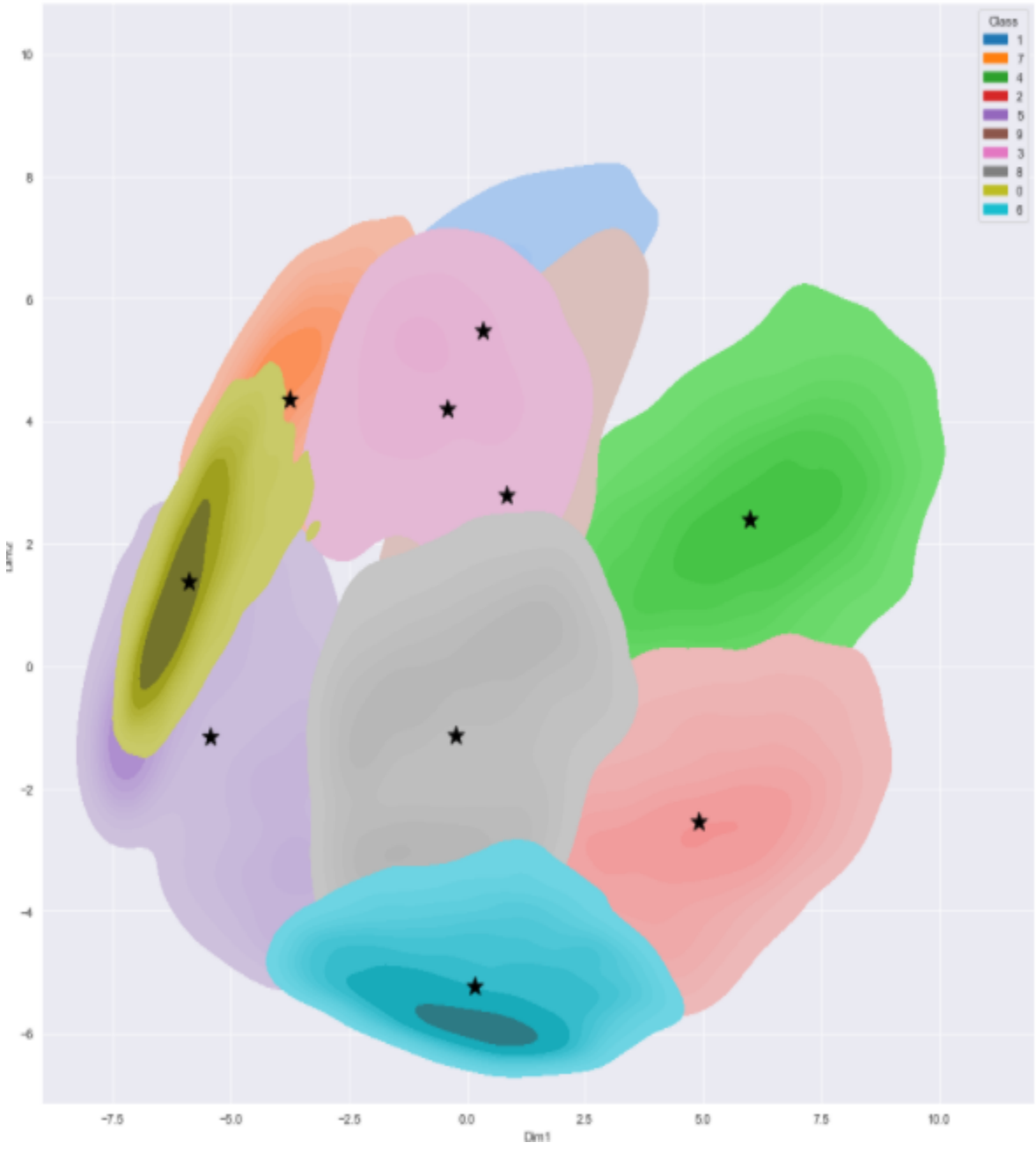
The class has the following functionalities:
Able to store the cluster centers.
Abe to store clusters and their respective sample points.
Able to predict cluster index for each sample point.
The class also provides us with an option to decide the distance metric to be used.

2. 2nd part is covered above and a well-commented code can be found in the colab notebook.
3. Training the k-means model on f-MNIST data with k = 10 and 10 random 784-dimensional points (in input range) as initializations. The number of points in each cluster is tabulated and visualized below:

Cluster	Number of points
0	6528
1	2341
2	7408
3	2860
4	9697
5	7575
6	9053
7	4354
8	7601
9	2583

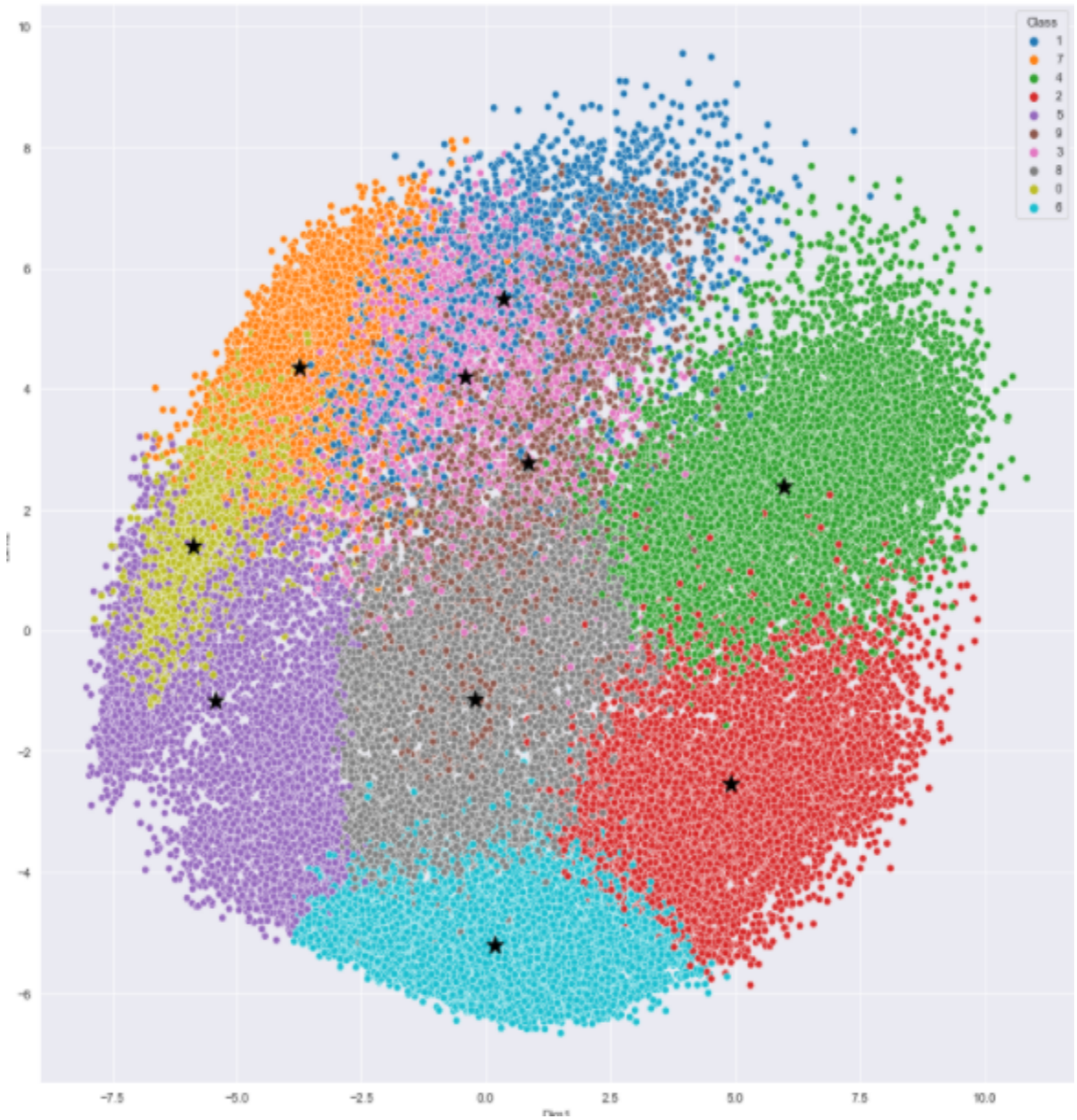
4. Plot of ‘PC 2’ against ‘PC1’ with ten(10) clusters and their respective cluster centers. NOTE: Considering the given fashion-MNIST dataset is 784-dimensional, it is not possible to plot a two-dimensional plot with all(784) features present. Hence, for simplicity dimension reduction technique (PCA) is applied to the dataset X and centroids. A KDE plot and a scatter plot have been plotted to visualize the clusters and respective centroids.
   The plots can be found below:

KDE Plot:



Black stars represent the centroids of the respective clusters.
Some overlap is seen among the clusters.

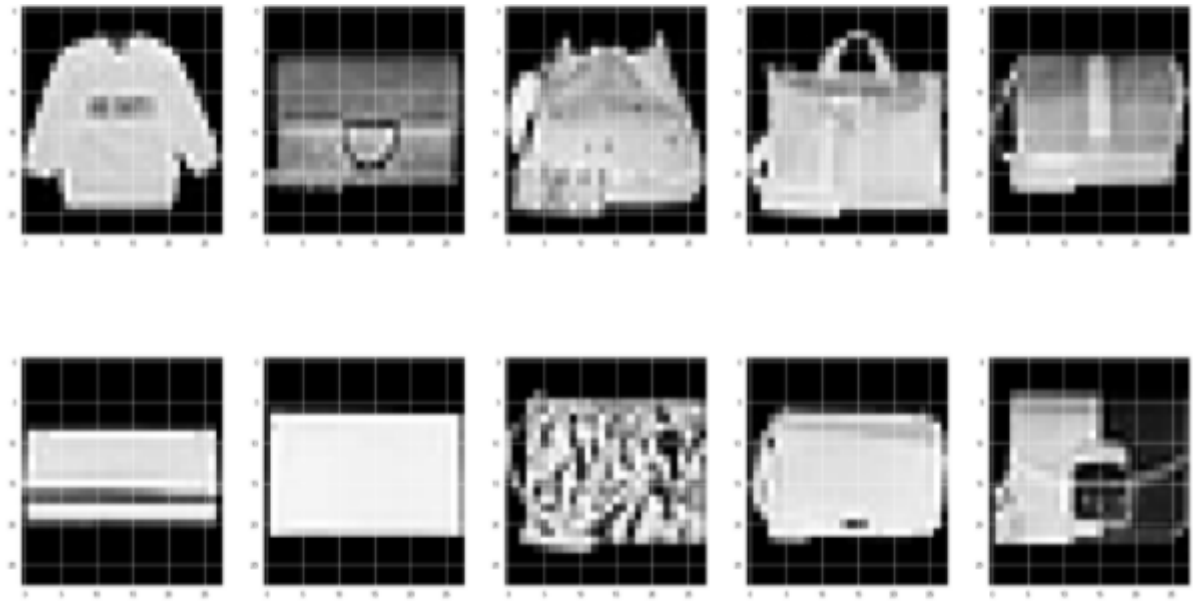
Scatter Plot:



Black stars represent the centroids of the respective clusters.
Some overlap is seen among the clusters.

5. Visualization of 10 images corresponding to each cluster is as follows:
   Cluster 0:

   Cluster 1:



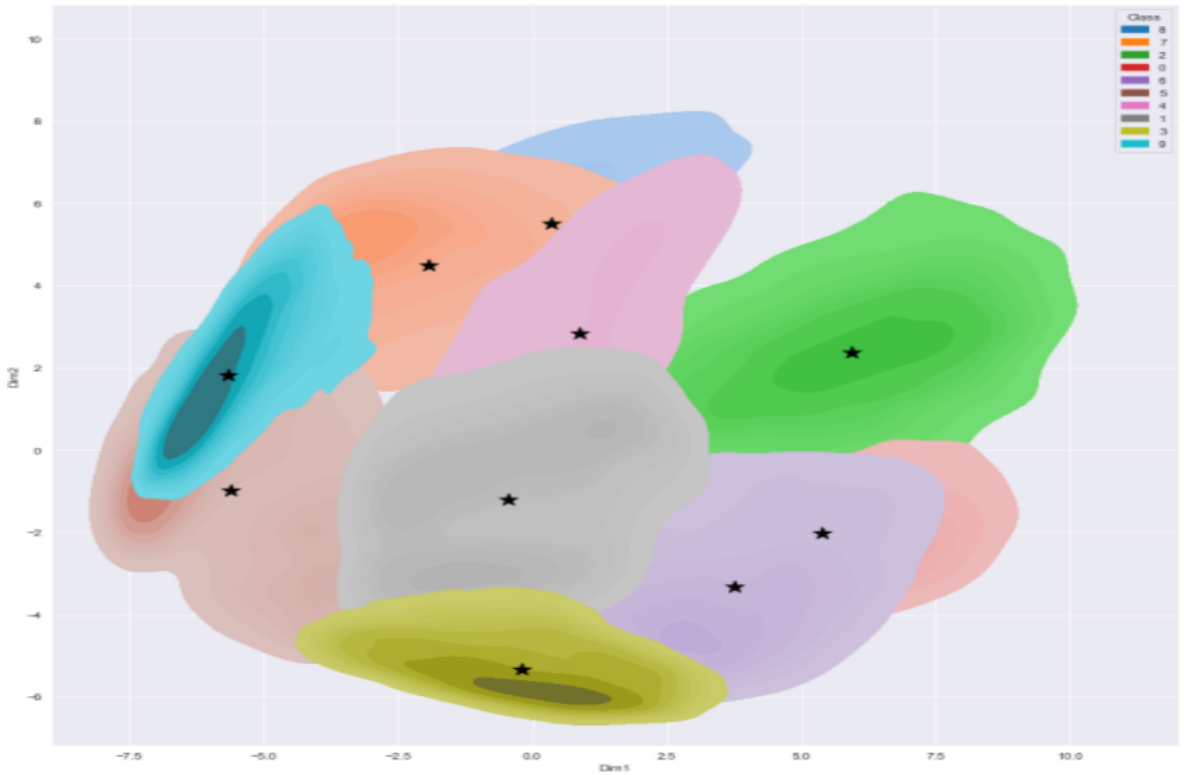
   Plots for all the clusters can be found in the colab notebook. Plots for all the other clusters were plotted each containing ten images belonging to that cluster.

6. Another KMeans model is trained with ‘10 images from each class’ as initializations. The number of points in each cluster is tabulated and visualized below:

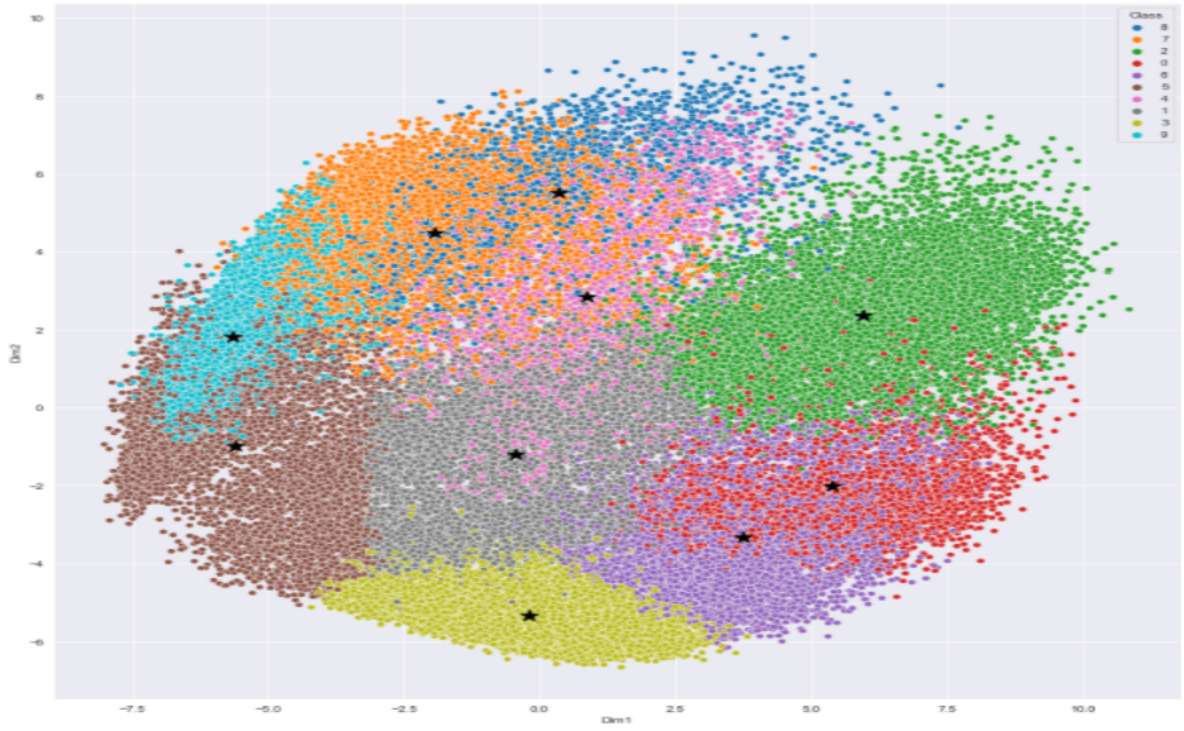
Cluster	Number of points
0	7507
1	2349
2	7510
3	2558
4	9759
5	7712
6	7772
7	3810
8	5834
9	5189

Plot of ‘PC 2’ against ‘PC1’ with ten(10) clusters and their respective cluster centers. NOTE: Considering the given fashion-MNIST dataset is 784-dimensional, it is not possible to plot a two-dimensional plot with all(784) features present. Hence, for simplicity dimension reduction technique (PCA) is applied to the dataset X and centroids. A KDE plot and a scatter plot have been plotted to visualize the clusters and respective centroids.

KDE Plot

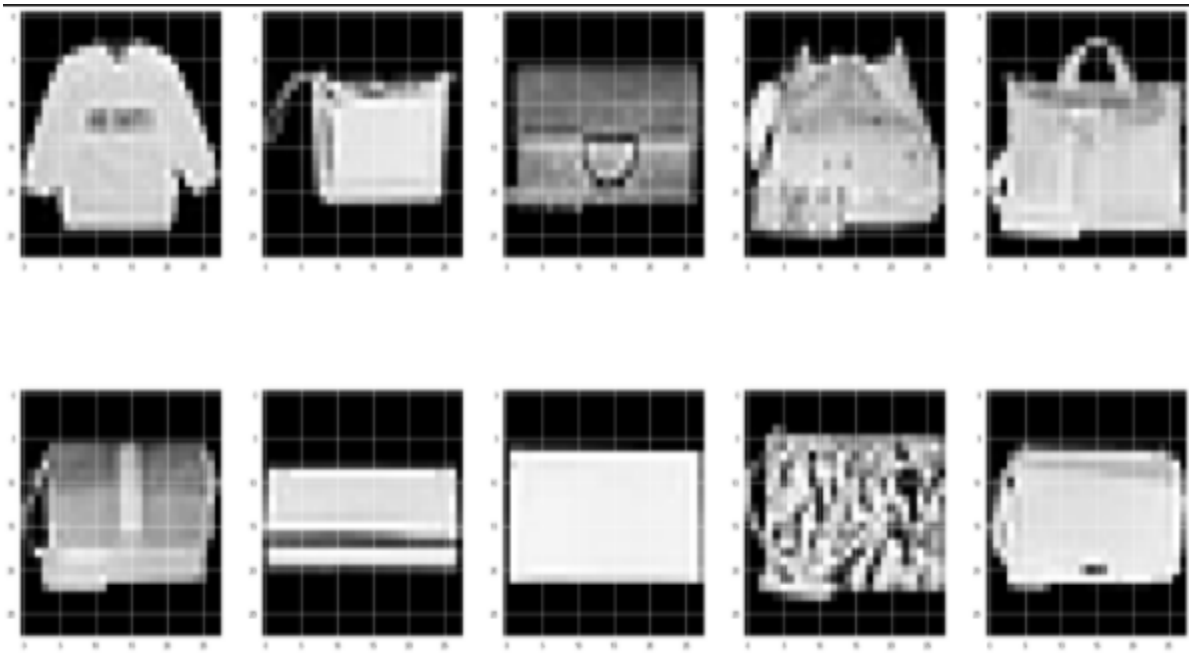


Scatter Plot:



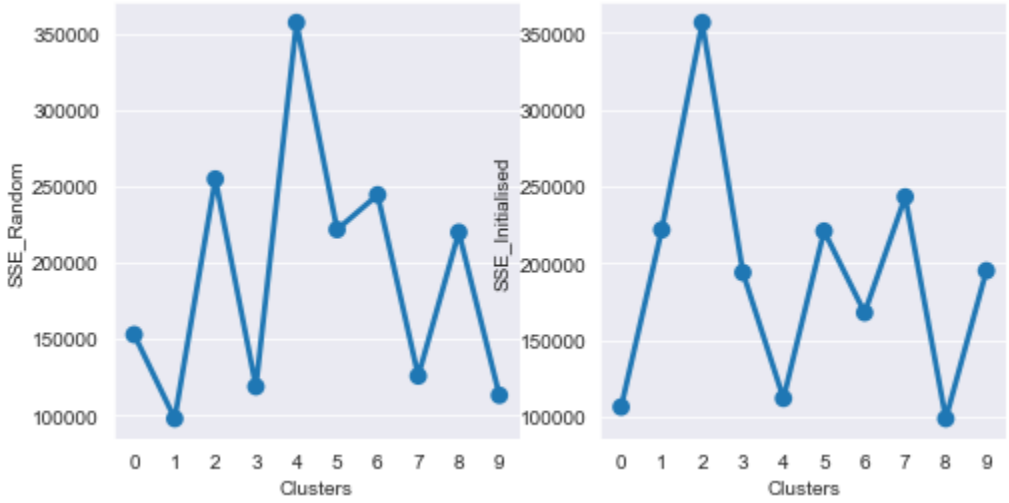
7. Visualization of 10 images corresponding to each cluster is as follows:
   Cluster 0:

   Cluster 1:



   Plots for all the clusters can be found in the colab notebook. Plots for all the other clusters were plotted each containing ten images belonging to that cluster.

8. Clusters of parts c and f are evaluated with the Sum of Squared Error (SSE) method. The respective scores are tabulated and visualized below:

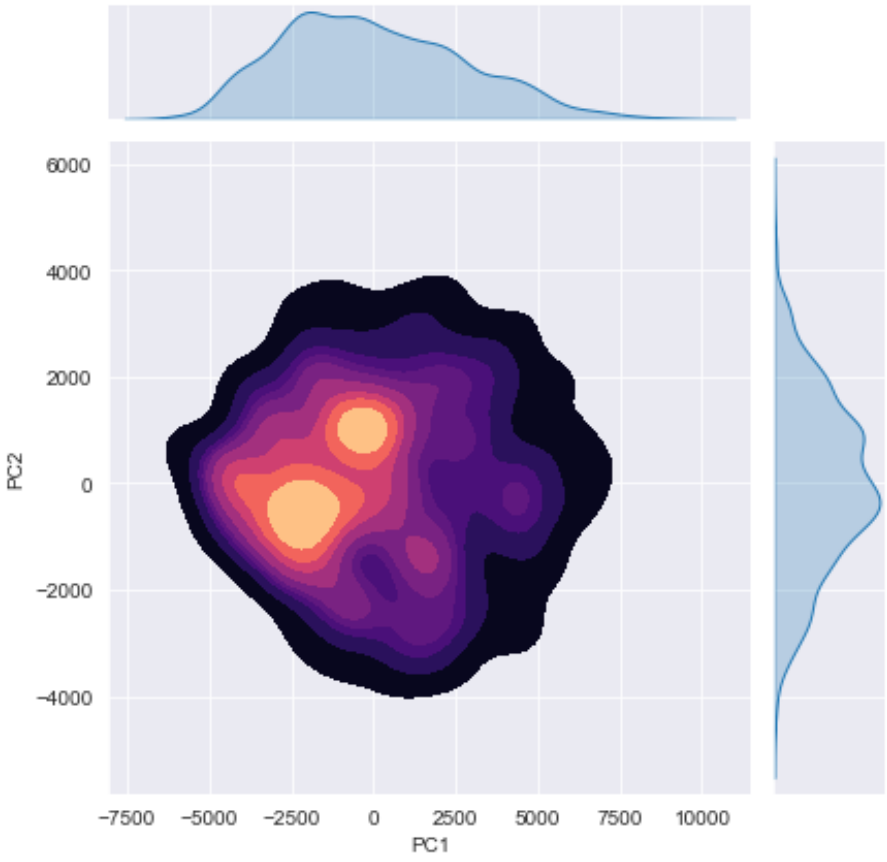


SSE --> Initialised --> 1905636.6901235175
SSE --> Random --> 1915335.1297465428

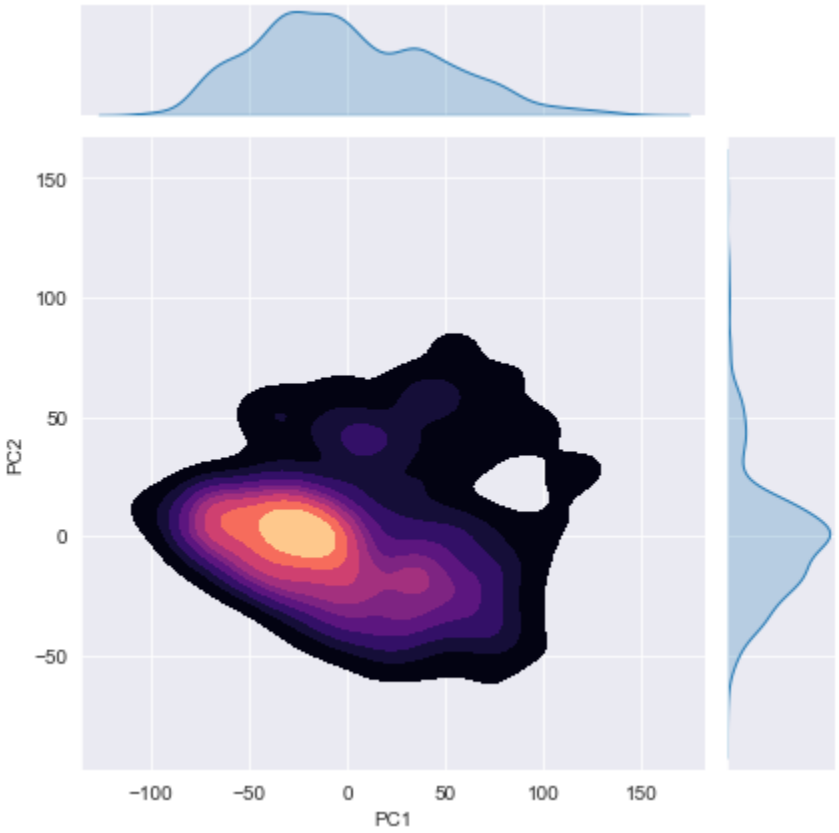
Observations and Conclusions
From the tabulated data as well as the point plot above, it can be observed that part ‘f’, that is ‘10 images from each class as initialization’ performs better clustering. This can be justified as follows: In part-c, where we take ‘10 random 784-dimensional points (in input range)’ as initialization the initial centroids (given by the user) may as well all belong to the same class in worst case scenario (because of the random nature of the technique). This results in poorly formed clusters that require a higher number of iterations to form separate clusters and that too are not guaranteed to yield good results. However, in part-f, where we take ‘10 images from each class’ as initialization the initial centroids all belong to different and unique classes. This results in better-formed clusters that require a lower number of iterations to form separate clusters. Furthermore considering the centroids were from different and unique classes initially, they are likely to contain sample points of that particular class only.

Question 3 - Brain cancer

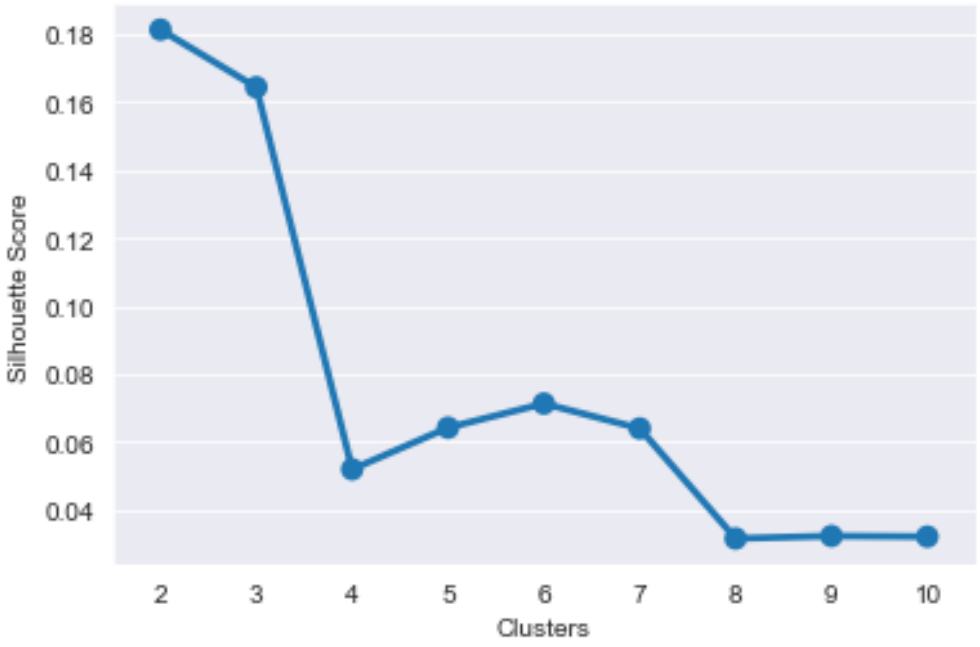
1. A joint dataset containing 1500 images from the given ‘Yes’ folder, and 1500 images from the given ‘No’ folder (total of 3000 images) is constructed externally and uploaded on Google Drive. The images from the said folder are read in and resized (100 x 100) using the cv2 library and stored as a list. The list is converted to a NumPy array, resized (3000, 10000), and finally converted to a Pandas Dataframe with appropriate column names and class labels. Statistics of the data are analyzed via ‘.head()’, ‘.info()’, and ‘.describe()’. Following observations are made: There are ten thousand(10000) features and three thousand(3000) data points. Only continuous features are available. The scale of continuous features is different. No missing (NaN) values. The features are normalized via StandardScaler() so that the scale of each variable will be the same. Some outliers were removed for better results.
2. Principle component analysis (PCA) dimension reduction technique with ‘n_components = 2’ (for two-dimensional data) is used. Visualization of the data after implementing the mentioned technique is as follows:
   Without Scaling:



With Scaling:

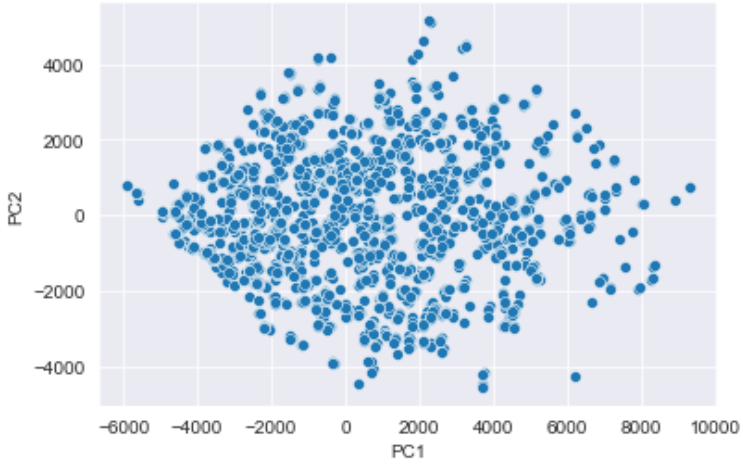


Silhouette Score Plot:

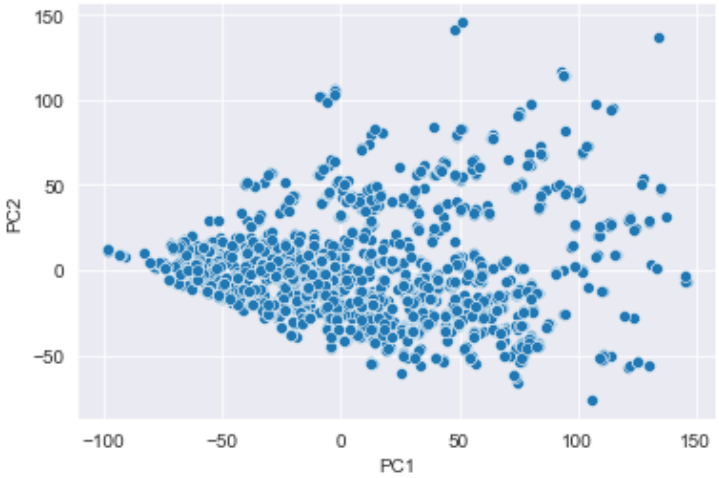


From the above KDEplots and Silhouette Score it can be seen that there are two communities.

3. Visualization of the communities from ‘Part A’ is as follows:
   Without Scaling:

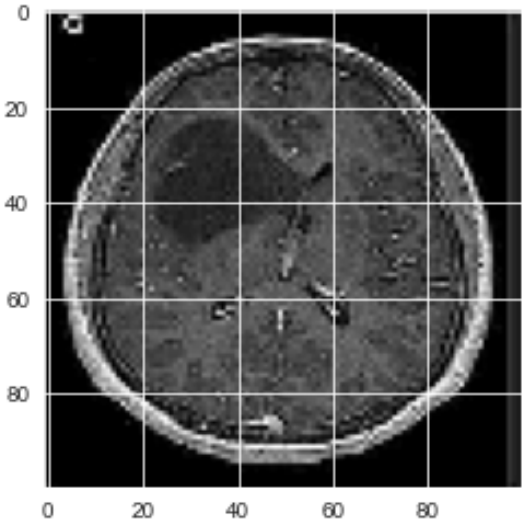


   With Scaling:

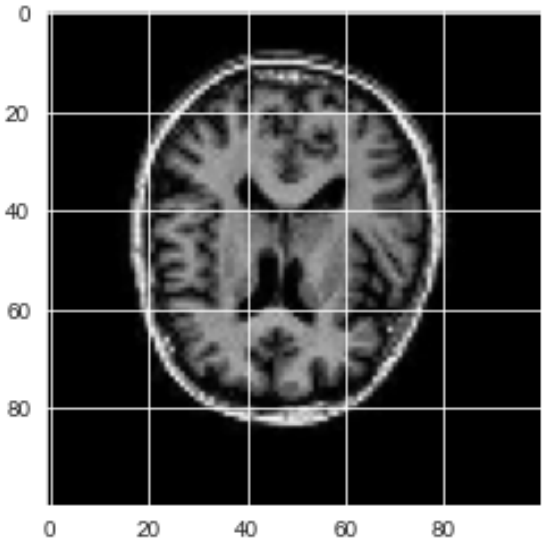


   With hue as original class:

From Cluster 1:



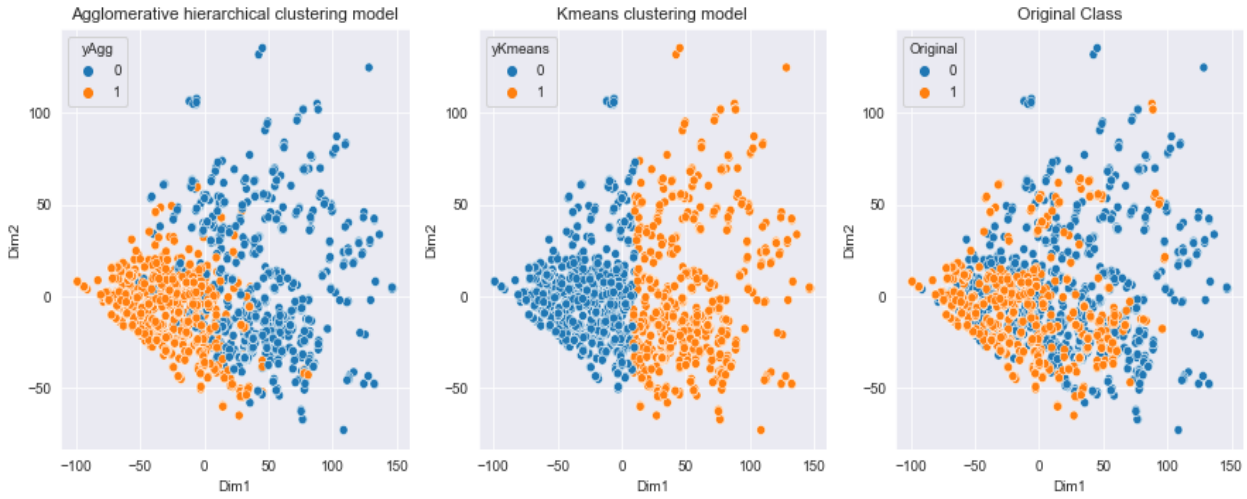
From Cluster 2:



4. Agglomerative hierarchical clustering (using sklearn) is applied to the mentioned dataset.
5. Comparison based on F1 score is tabulated and visualized below:

Clustering Technique	F1 Score
Agglomerative hierarchical	0.704
KMeans	0.696

Comparison Based on Visualisation:



Leftmost - Agglomerative, Center - Kmeans, Rightmost - Original Class

Observations:

a. Agglomerative clustering has a better F1 score.

b. KMeans divides the given samples into strictly two halves(left and right) from the middle.

c. Agglomerative hierarchical clustering divides the given samples along with a diagonal.

Conclusions:

Agglomerative returns better F1 score than Kmeans and fits the data more correctly. Agglomerative hierarchical clustering is more efficient in the sense that it resembles the original classification. This can be justified by the fact that though KMeans cluster based on distance from the centroid of the cluster sample points, Agglomerative hierarchical clustering follows a bottom-up approach. A structure that is more informative than the unstructured set of clusters returned by flat clustering.