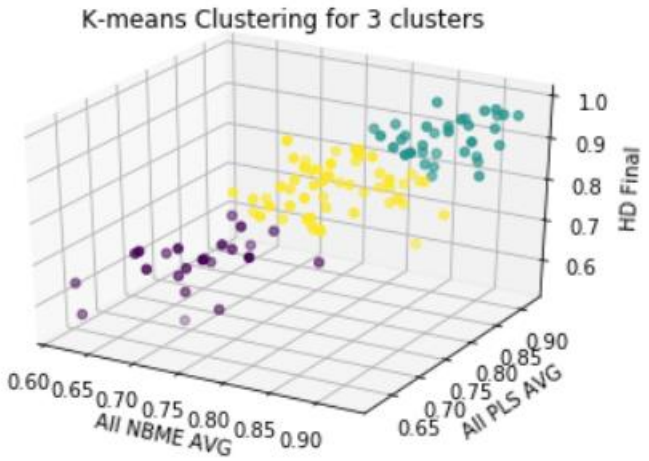
CS 7830 Machine Learning Assignment 1

Vidhya Lakshmi Sankaranarayanan U00924508

1. **K-means clustering with different number of clusters**
   a. **Scatter plot for Number of clusters 3**

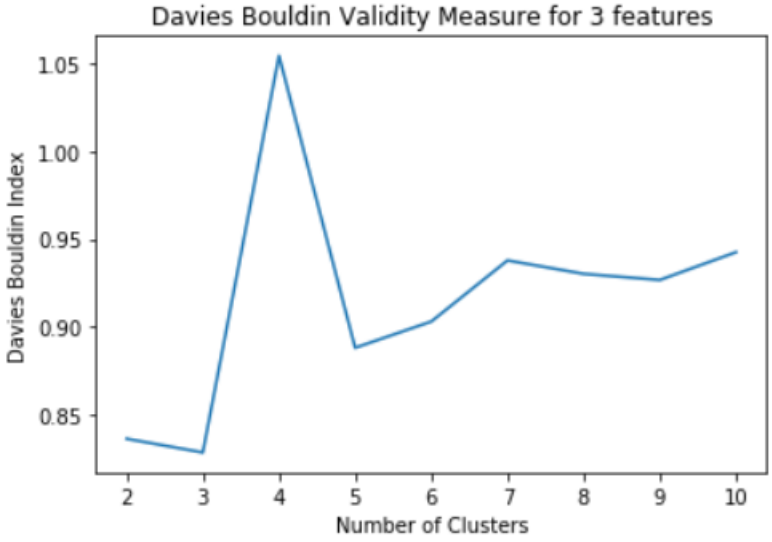


Three features such as All NBME AVG, ALL PLS AVG and HD FINAL are scattered in 3D and plotted in 3 different colors belonging to three clusters.

   b. **Best number of clusters**

Based on comparing the scatter plot of data with different number of clusters, it is easy to predict that number of clusters can't be above 4 because having more clusters for closer density of data is inefficient. Between 2, 3 and 4 number of clusters, it is difficult to predict the perfect number of clusters as the data are scattered randomly. Visually, number of clusters being 3 serves a better solution to this respective dataset because purple cluster is lightly dense and in lower range whereas yellow cluster is in middle range and green cluster holding higher range of data. So, number of clusters as 3 looks visually appealing.

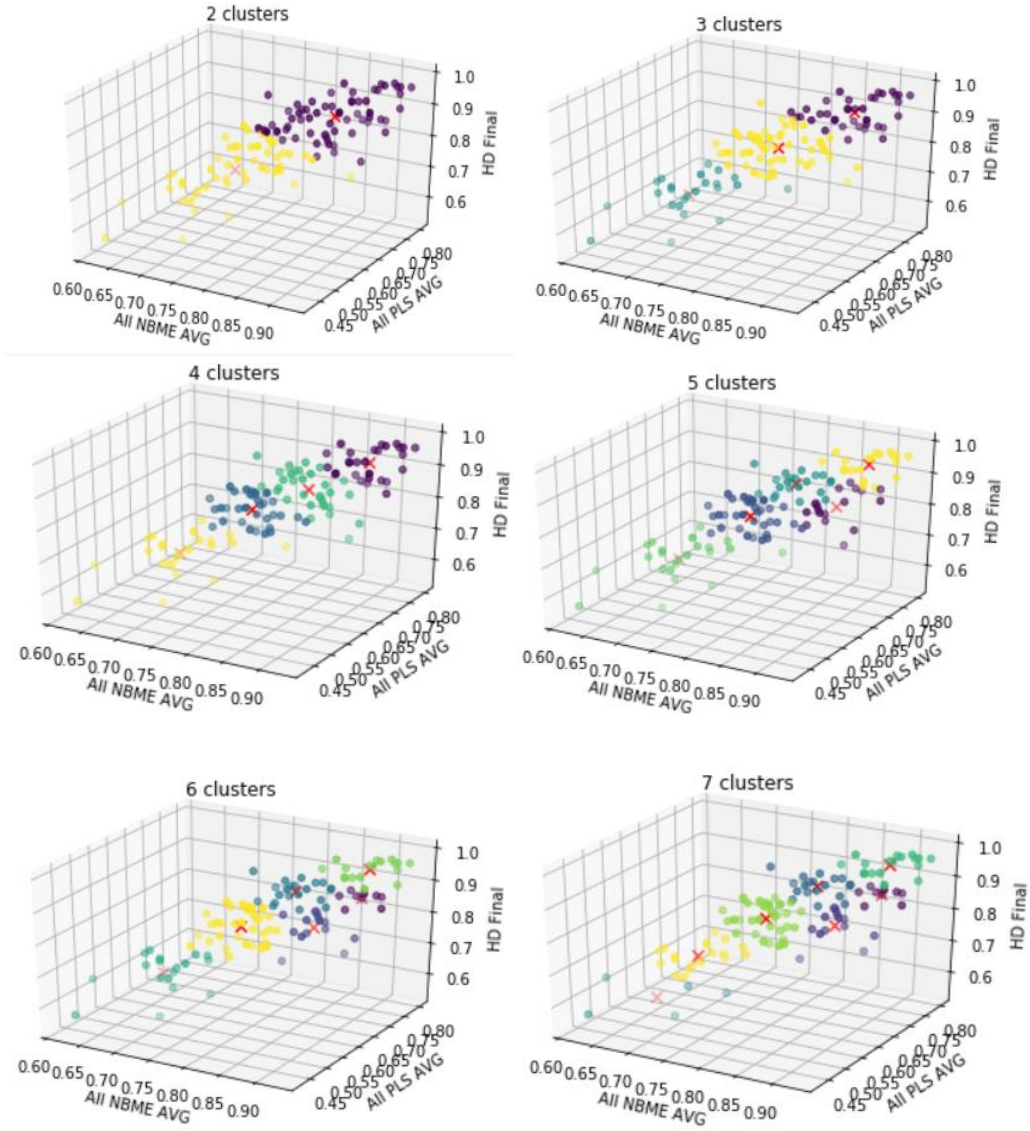
   c. **Davies Bouldin Validity measure Implementation**

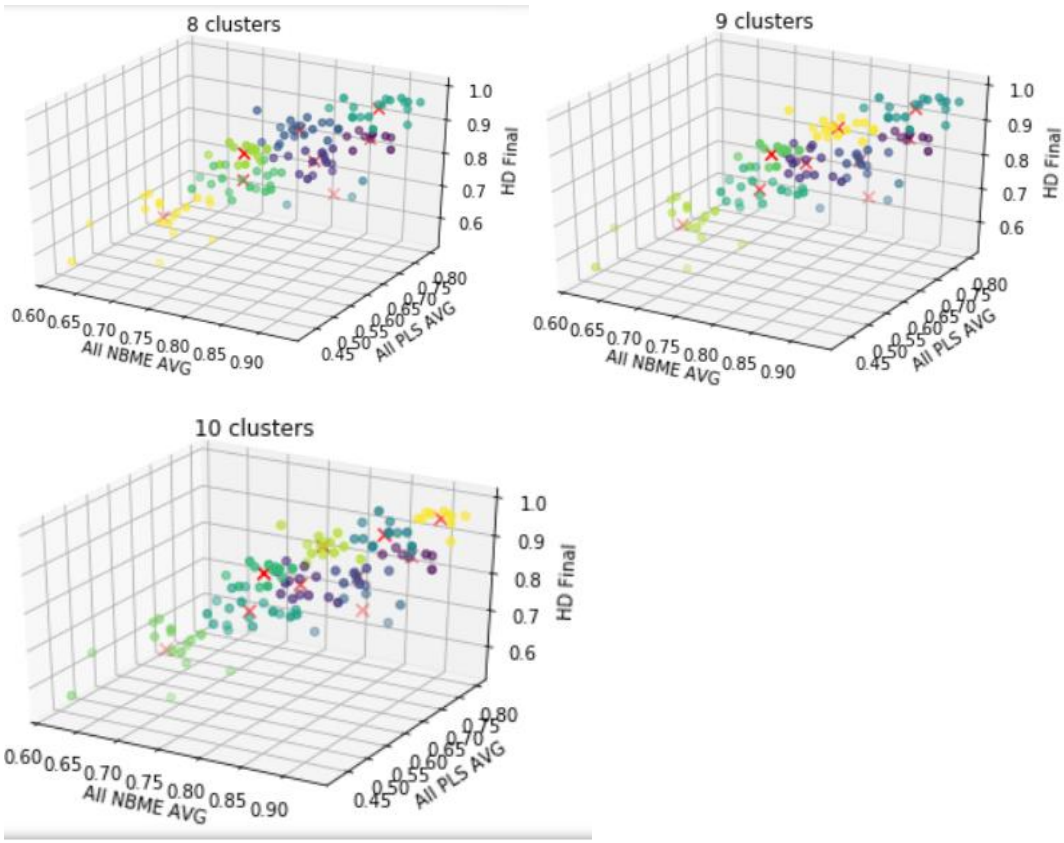


Davies index for 3 features from k=2 to 10 : [0.8361969521027511, 0.8283853976267831, 1.0544353622884346, 0.8880260866458065, 0.9029982946395885, 0.9378052324852735, 0.9301516222940234, 0.9266630698482502, 0.9424895553788771]

Davies Bouldin index is ratio of intra cluster distance to inter cluster distance. Inter cluster distance should be more to get good clustering results. So, lower the DB index, better the quality of clusters.

In the figure, first minimum is seen when the number of clusters is 3(0.82838). For 4 clusters, DB index suddenly rises and reduces in value for 5 number of clusters. According to elbow method, first fall is seen when number of clusters is 3. Hence **number of clusters being 3** could provide good quality of clustering according to DB index. Also, it satisfies my observation.

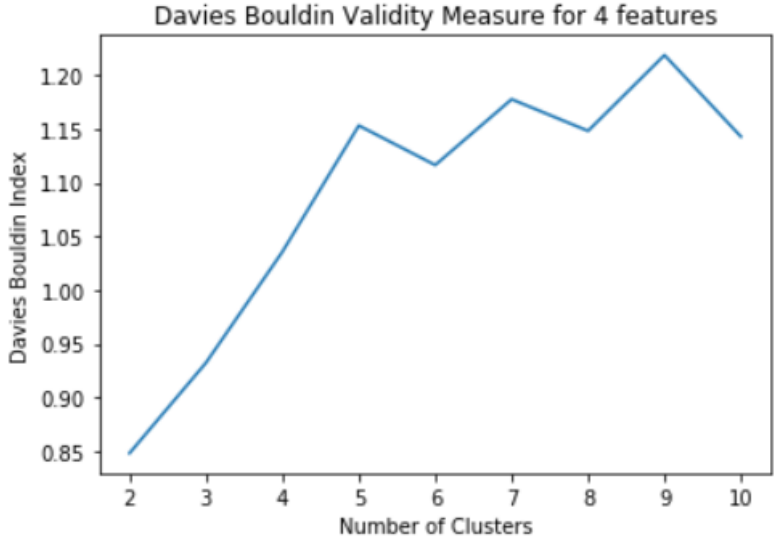




2. K-means clustering with different features

a. Adding 'all_irats_avg_n34' feature

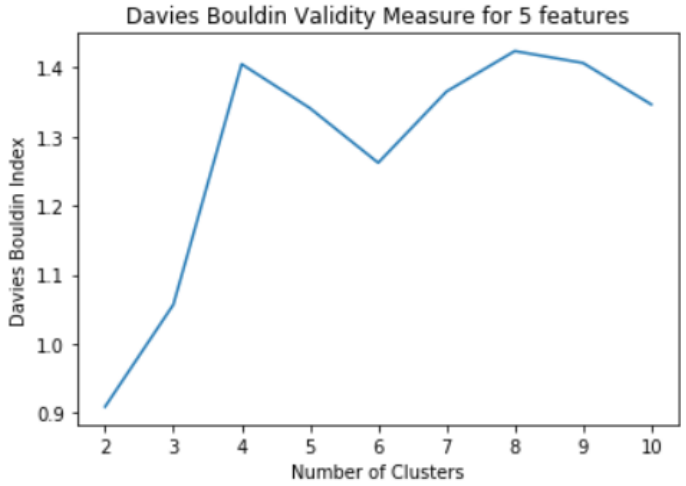
Davies index for 4 features from k=2 to 10: [0.8479071099001925, 0.932171966532 6081, 1.0356652283385022, 1.1533806977182353, 1.1165417524702899, 1.1778802 064100504, 1.1483813654119288, 1.2190573310238182, 1.143059927043281]



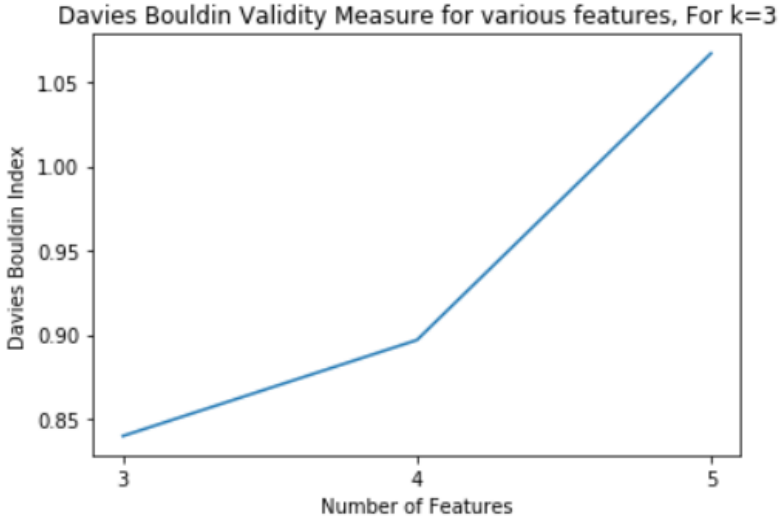
Adding fourth feature did not provide any improvements in the quality of clustering by looking at the DBI metric. For k=2, there is little increase in DB whereas after that there is continuous increase in DB index with the addition of 'all_irats_avg_n34'.

b. Adding 'HA_final' feature

Davies index for 5 features from k=2 to 10 : [0.908626714469095, 1.05658682551248, 1.4051011367243522, 1.3412538704957426, 1.2620722206354915, 1.3653176280564745, 1.4239534228565003, 1.4067070720773551, 1.346638010452112]



Same as 4 features, adding 5 features also increased the db index. Above graph shows the DB index of 5 features for cluster number between 2 to 10. From the DB index ,we can say that 5 features may not provide good clustering results.



With best number of clusters already found as 3, adding 4 features as well as 5 features increased the DB index which is clear from the graph showing comparison between the number of features and DB index for number of clusters being 3. As the DB index increase for 4 and 5 features, this cannot be a good combination of clusters to provide good clustering results.

From DB index results, adding 'all irats avg n34' or HA_final did not improve the quality of clusters.

3. **Fuzzy C-means clustering**
   a. **With Num of clusters 3 and num of features 3**
   Comparing Kmeans and Fuzzy cmeans clustering using the centroid values.
   K-means Centroids values are

```
centroids after converging: [[0.87121951 0.72460732 0.92073171]
 [0.81019608 0.63046078 0.82803922]
 [0.71326087 0.54484783 0.69521739]]
```

Fuzzy c means centroid values are

```
Centroids while converging: [[0.80679723 0.62840748 0.83491692]
 [0.71539846 0.54668657 0.69773623]
 [0.8750833  0.72888197 0.91805188]]
```

Centroid values are almost similar which implies that fuzzy c means provided clusters similar to k-means clustering for this dataset.

When we look at the some of the elements of membership matrix(1 to 10),

[4.43687400e-01 9.15075963e-02 4.64805003e-01] – cluster 2

[1.46076381e-02 1.19186107e-02 9.73473751e-01] – cluster 2

[3.59639611e-01 4.63111963e-02 5.94049193e-01] – cluster 2

[8.85367770e-01 2.50938316e-02 8.95383985e-02] – cluster 0

[2.86505191e-01 5.74298477e-02 6.56064962e-01] – cluster 2

[4.89199041e-02 3.46970875e-02 9.16383008e-01] – cluster 2

[5.97259067e-04 9.97564077e-01 1.83866408e-03] – cluster 1

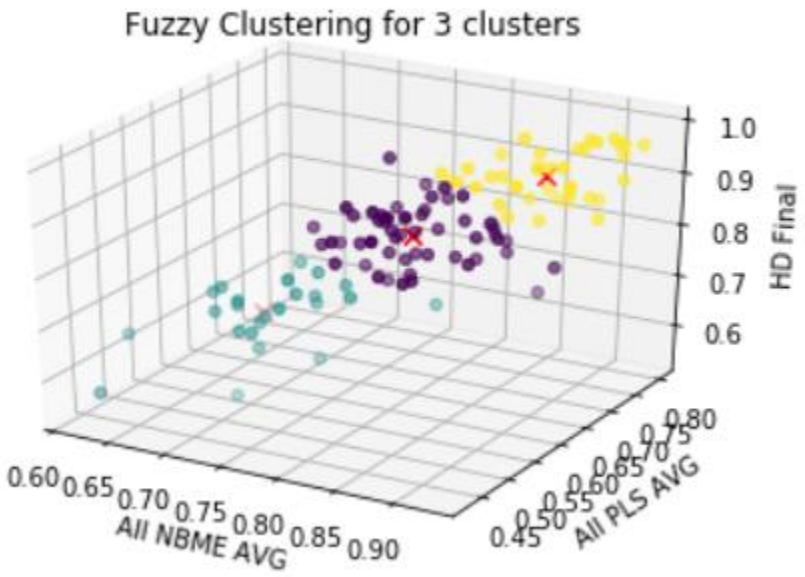
[1.67376479e-01 9.71473721e-02 7.35476149e-01] – cluster 2

[2.64274188e-02 8.61212626e-01 1.12359955e-01] – cluster 1

[9.91209783e-01 1.57517372e-03 7.21504334e-03] – cluster 0

When compared with Kmeans clustering

[2 2 2 0 2 2 1 2 1 0 2 2 0 2 0 2 2 2 1 2 0 1 0 0 0 2 0 0 0 1 2 1 0 2 0 0 2
2 2 2 2 0 1 2 2 0 1 0 2 1 2 2 1 1 0 1 1 0 0 0 2 0 1 0 0 1 1 1 0 2 1 1 2 2
0 0 2 2 2 2 0 2 2 2 2 0 2 2 2 1 0 2 2 2 0 2 1 0 0 1 0 2 0 2 0 2 2 2 1 2 1
2 0 2 2].



The above sample membership values are same in fuzzy and k means when fuzzy member values are hardened. This may imply that both kmeans and fuzzy provided same clustering results for this set of features. Usually Fuzzy c means provides better clustering results. In the dataset provided, with my observation of best number of clusters and set of features, both fuzzy and kmeans provided good results

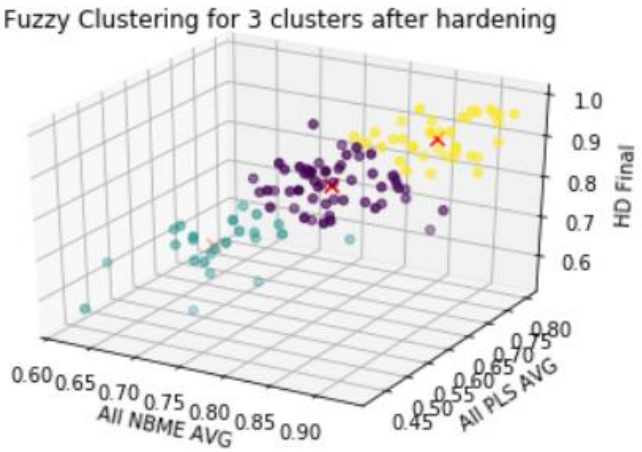
b. Harden the fuzzy c means and comparison

Fuzzy c means results after hardening the membership values

array([2, 2, 2, 0, 2, 2, 1, 2, 1, 0, 2, 2, 0, 2, 0, 2, 2, 2, 1, 2, 0, 1,
0, 0, 0, 2, 0, 0, 0, 1, 2, 1, 0, 2, 0, 0, 2, 2, 2, 2, 2, 0, 1, 2,
2, 0, 1, 0, 2, 1, 2, 2, 1, 1, 0, 1, 1, 0, 0, 0, 2, 0, 1, 0, 0, 1,
1, 1, 0, 2, 1, 1, 2, 2, 0, 0, 2, 2, 2, 2, 0, 2, 2, 2, 2, 0, 2, 2,
2, 1, 0, 2, 2, 2, 0, 2, 1, 0, 0, 1, 0, 2, 0, 0, 0, 2, 2, 2, 1, 2,
1, 2, 0, 2, 2], dtype=int64)

Both the labels of fuzzy c means and Kmeans are the same.

Fuzzy c means with 3 number of clusters



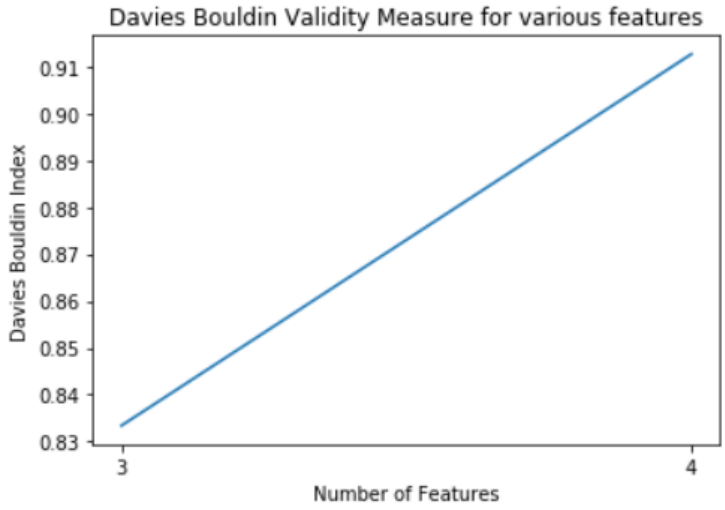
For Kmeans, DB index is ***0.8399436766064587***

For Fuzzy c means, DB index is ***0.8333976773821119***

According to this DB index, in very minute difference as Fuzzy DB index is lesser, Fuzzy c means provides better clustering results. Also, for this dataset, as the variation is very minute, K-means is not at all a bad clustering as it also served to provide better results similar to Fuzzy with respective of this dataset.

c. Adding one more feature

From the above DB index of Fuzzy c means, adding 4th features resulted in increase in DB index which indicates adding 4 features does not provide good quality of clustering results.



With the DB index adding 4th feature does not provide better results. But if the features reduction is done and visualized using PCA, then it would be clear to come to a decision.