Clustering(K-Means(hard) and Fuzzy C Means(soft))

In this assignment, K-Means and Fuzzy C Means algorithms are implemented from scratch. K-Means and Fuzzy C Means algorithms are applied to a real time student performance dataset to know the performance of the students in different exams and to identify the low performing students which helps to plan the academic schedule that helps to take care of the low performing students to perform better.

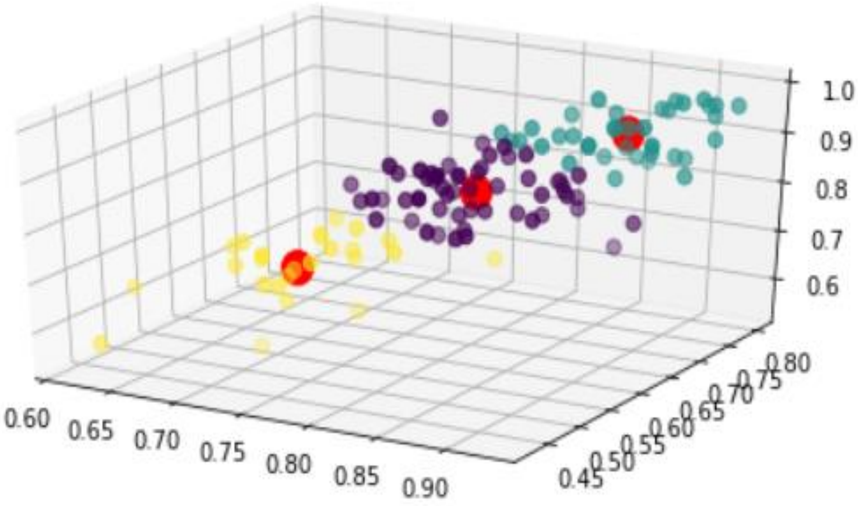
Note:

For K Means algorithm, initial centroids are randomly selected from the datapoints. The convergence condition is reached when there is no change in the centroids.

For Fuzzy C Means algorithm, membership matrix is initialized such that for a datapoint, sum of its membership in all the clusters is 1. The convergence condition is reached when the difference between the previous and current updated membership matrix is very small (0.01)

1. K-means clustering with different number of clusters

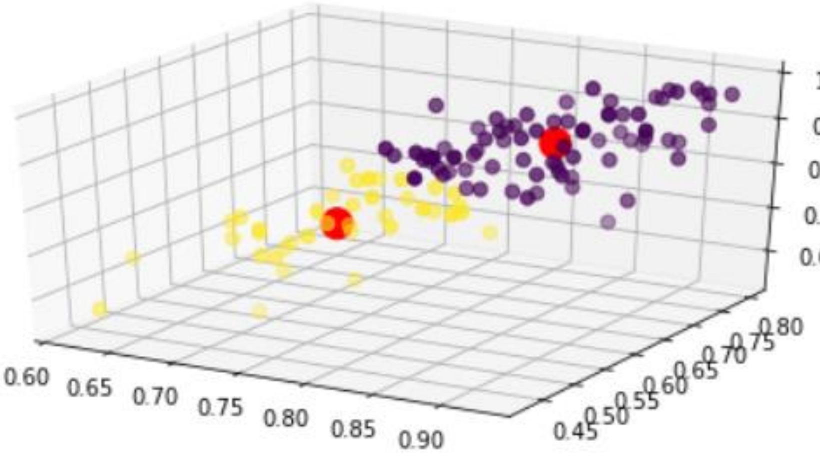
a) K-Means clustering algorithm is applied on data set with 3 features:
'avg_exam1', 'avg_exam2', 'exam3_final' and the number of clusters k=3
Below is the 3-D plot visualization.



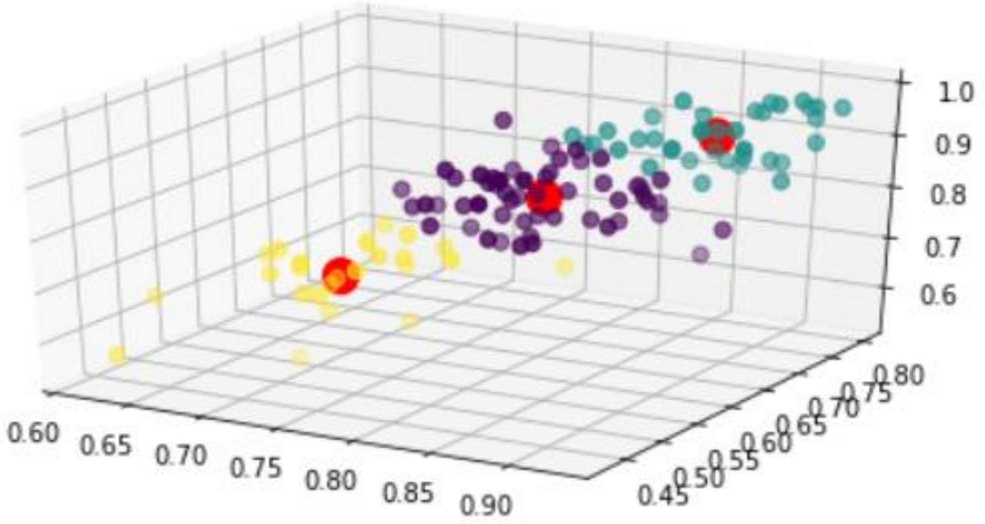
In the above plot, the red color circles are centroids of the clusters and yellow, violet and blue color dots are datapoints and each color represent one different cluster.

b) Now the K-Means clustering algorithm is applied on data with 3 features:
'avg_exam1', 'avg_exam2', 'exam3_final' and the number of clusters k ranging from 2 to 10
Below is the 3-D plot visualization.

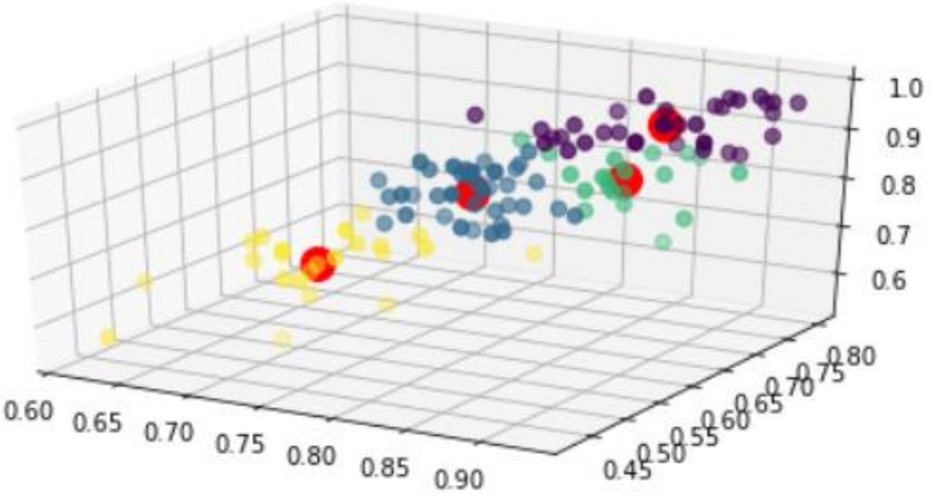
Below is for k=2:



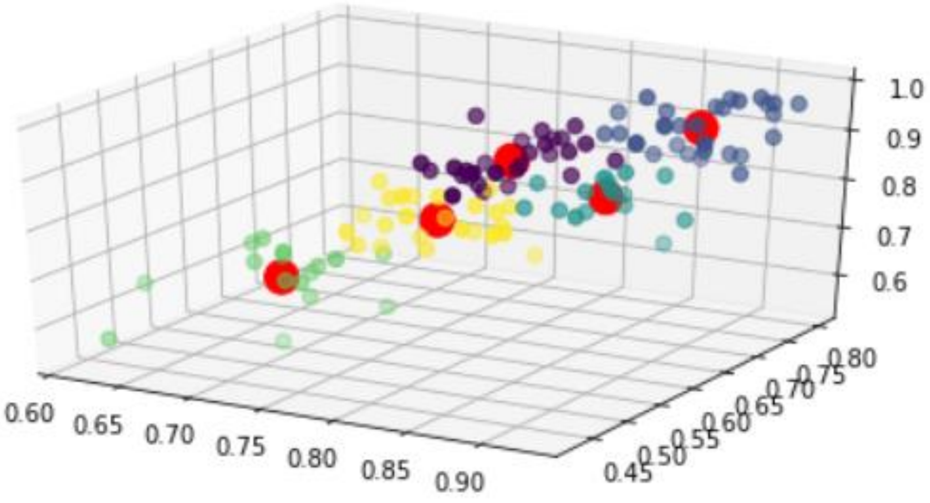
Below is for k=3:



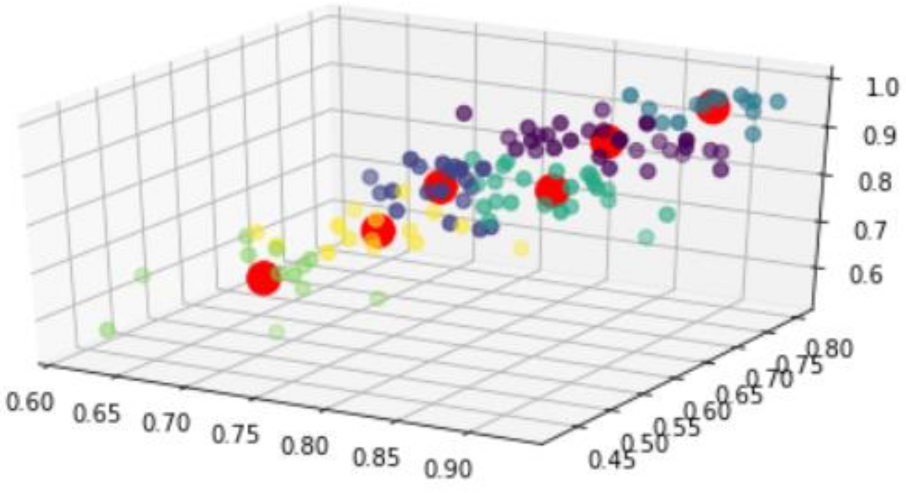
Below is for k=4:



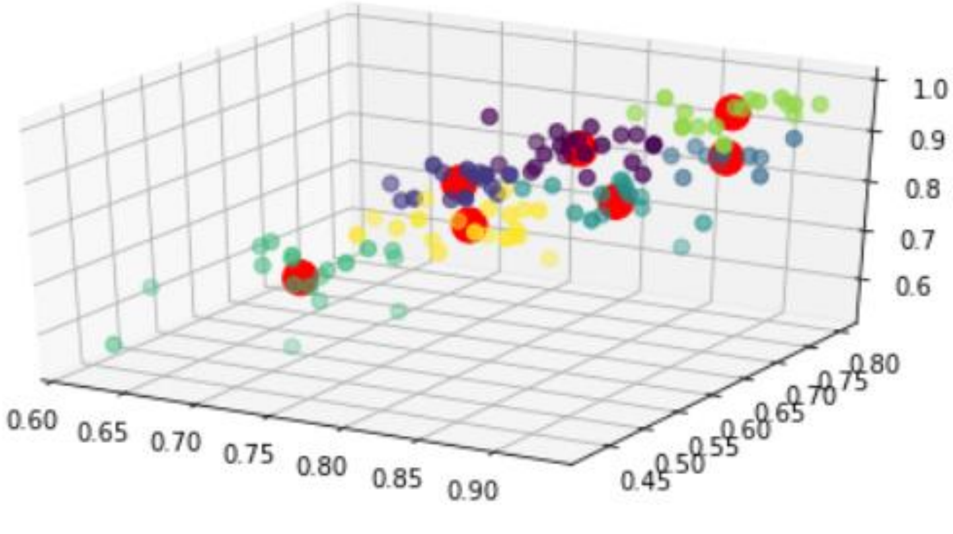
Below is for k=5:



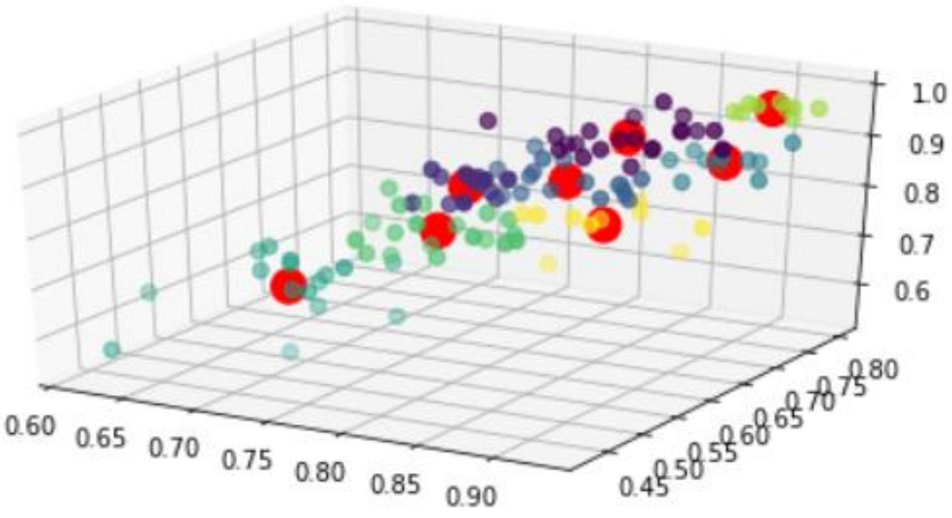
Below is for k=6:



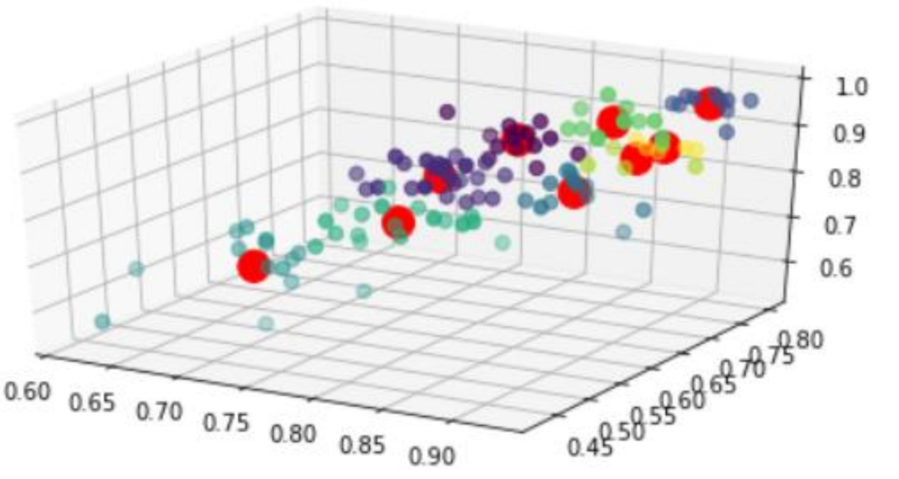
Below is for k=7:



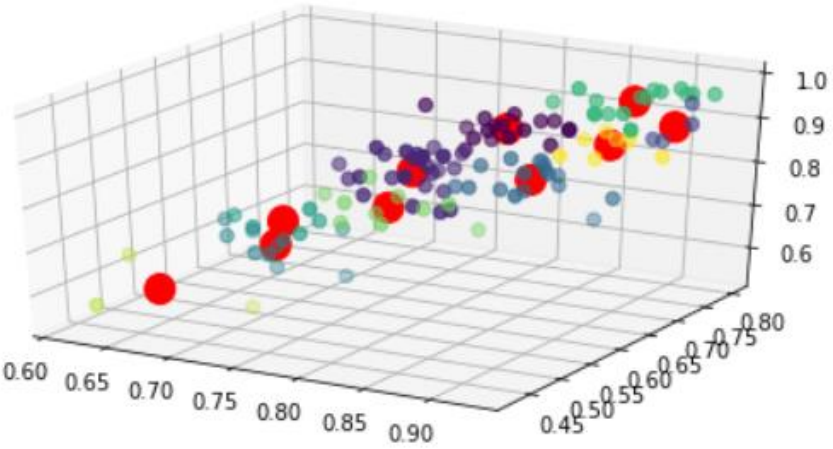
Below is for k=8:



Below is for k=9:



Below is for k=10:



In the above plots, the red color circles are centroids of the clusters and remaining colors represent datapoints and each color represent one different cluster. 'k' is the number of clusters.

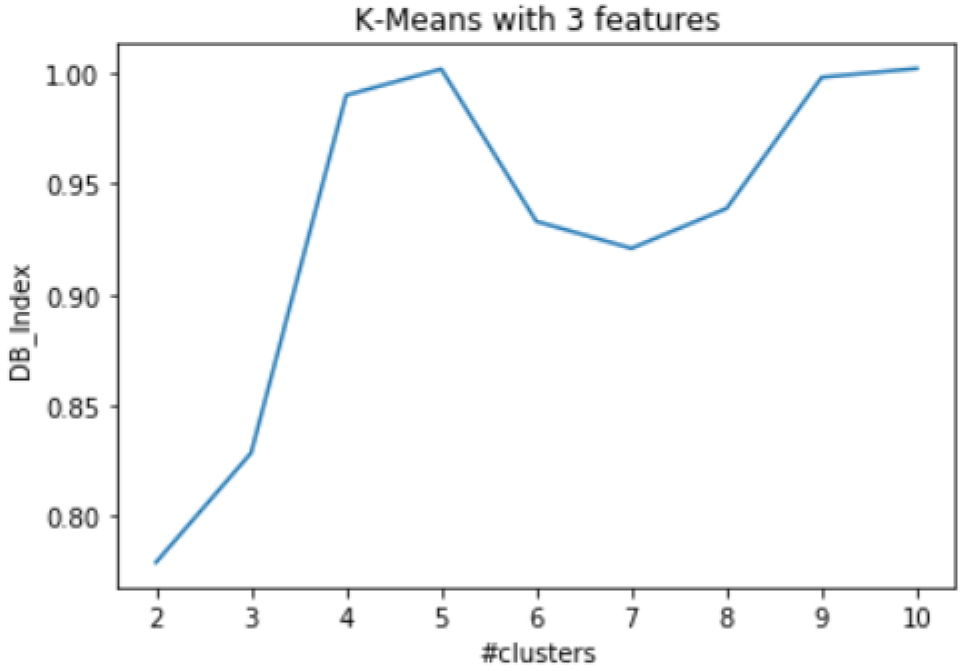
From the above plots, I believe the best number of clusters is 3. Because the clusters are separated well which means that the inter cluster distance is more and the clusters don't overlap with each other.
For the clusters more than 3, it can be observed that the clusters overlap with each other.
The clusters clearly separate the low, medium and high performing students with number of clusters=3

c) Davies Bouldin validity measure is applied on the data set with 3 features mentioned in section 1b and DB indices are calculated for all the k (#clusters) ranging from 2 to 10.

Below are the DB Indices

Table1:

#clusters(k)	**DB Index**
2	**0.7788663302582429**
3	0.8283853976267831
4	0.9900493886349256
5	1.0019112199640072
6	0.9329766044184401
7	0.9207832855790258
8	0.938863009944501
9	0.998195316511401
10	1.0021627162764717

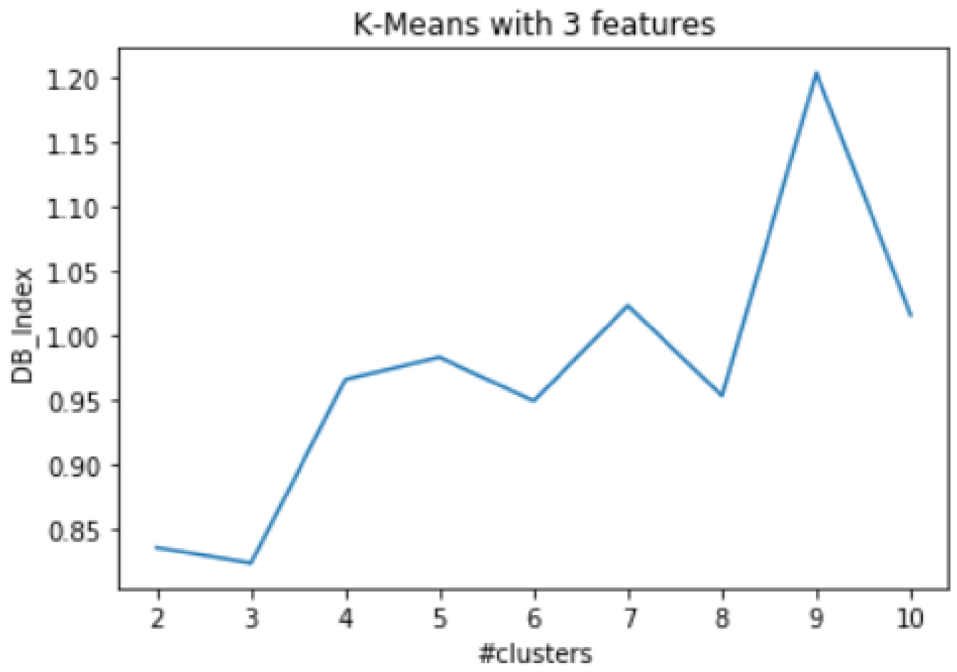


From the values of DB Index and the plot between #clusters and DB Index, it can be observed that optimum number of clusters is 2 (because lower the DB Index better the clustering).

It is also observed from the results of DB Indices that for few runs of the algorithm, DB Index is less when 'k' is 3 compared to DB Index when 'k' is 2. Below are the DB Indices in this case.

Table2:

#clusters(k)	**DB Index**
2	0.8361969521027511
3	**0.8239775799329068**
4	0.9661629833539197
5	0.9832886702303718
6	0.9494067849358477
7	1.023225185766923
8	0.9533332390332434
9	1.2032480107220291
10	1.0160494974063246



From the above two plots of #clusters and DB Index, it is observed that sometimes best number of clusters is 2 and sometimes 3. So, I believe the best number of clusters depends on the initial random choice of centroids.

2. K-means clustering with different features

As mentioned in section 1c, we can consider optimum clusters as both 2 and 3 based on the initial choice of centroids.

a) Now a new 4th feature 'avg_exam4' is added to the above 3 features (in section 1c) and K-Means algorithm is implemented for the best number of clusters(k=2,3) and DB Indices are calculated.
Below are the results.

DB Index with 4 features and best k=2 is: **0.8479071099001926**
DB Index with 3 features and best k=2 is: **0.7788663302582429**

DB Index with 3 features and best k=3 is: **0.8239775799329068**
DB Index with 4 features and best k=3 is: **0.932171966532608**

By comparing DB Index values for 3 features and 4 features in both cases (considering both 2 and 3 as best number of clusters as mentioned in section 1), it can be observed that DB Index for 3 features is less than that of 4 features. As lower the DBI better the clustering, adding the 4th feature did not improve the clustering results.

b) Now a new 5th feature ‘exam5_final’ is added to the above 4 features (in section 2a) and K-Means algorithm is implemented for the best number of clusters(k=2,3) and DB Indices are calculated.
Below are the results.

DB Index with 5 features and best k=2 is: **0.8985448538827289**
DB Index with 3 features and best k=2 is: **0.7788663302582429**

DB Index with 3 features and best k=3 is: **0.8239775799329068**
DB Index with 5 features and best k=3 is: **1.0363454565750736**

By comparing DB Index values for 3 features and 5 features in both cases (considering both 2 and 3 as best number of clusters as mentioned in section 1), it can be observed that DB Index for 3 features is less than that of 5 features. As lower the DBI better the clustering, adding the 5th feature did not improve the clustering results.

3. Fuzzy C-means clustering

a) Fuzzy C Means algorithm is implemented with best number of features and best number of clusters.
Best number of features obtained from section 2 is: 3
Best number of clusters obtained from section 1 is: 2 or 3 (depends on initial choice of centroids)

Centroids obtained by K-Means and Fuzzy C Means algorithms:

With 3 features and 2 clusters:

K-Means centroids:

```
[array([0.84733766, 0.68698442, 0.88402597]), array([0.74210526, 0.56568684, 0.73421053])]
```

Convergence condition is reached after 6 iterations.

Fuzzy-C-Means Centroids:

```
[array([0.8541538, 0.69919983, 0.89323003]), array([0.75066968, 0.57436903, 0.74845249])]
```

Convergence condition is reached after 15 iterations.

With 3 features and 3 clusters:

K-Means centroids:

```
[array([0.81122727, 0.63258545, 0.83418182]), array([0.87628378, 0.73162703, 0.92162162]),
array([0.71326087, 0.54484783, 0.69521739])]
```

Convergence condition is reached after 5 iterations.

Fuzzy-C-Means Centroids:

```
[array([0.80673041, 0.62830248, 0.83481904]), array([0.87501726, 0.72879362, 0.91798747]),
array([0.71533588, 0.54663643, 0.69764368])]
```

Convergence condition is reached after 16 iterations.

It can be observed that in both cases (k=2 and k=3), the difference between the centroids obtained from K-Means and Fuzzy C Means algorithms is very small. But the number of iterations to reach the convergence condition in K-Means are less compared to that of Fuzzy C Means algorithm.

b) Hard clustering is implemented for the data points based on the membership values obtained in the Fuzzy C Means algorithm. Below are the DB Indices of Fuzzy-C-Means with 3 features.

Table3:

#clusters(k)	**DB Index**
2	0.7948460395583281
3	0.831450582049154
4	1.0707567120338306
5	1.206038187998367
6	1.0058061924802597
7	1.1574216205086396
8	1.1133257604661069
9	1.0285272575460935
10	1.1075236306710674

By comparing the DB indices of K-Means (2 tables in section1) and Fuzzy C Means (table in this section) for 3 features (If k=2 is considered as best number of clusters in K-Means compare the DB Index values of k=2 in Table 1 and 3; If k=3 is considered as best number of clusters in K-Means compare the DB Index values of k=3 in Table 2 and 3), it can be observed that DB indices for K-Means are less than the DB indices of fuzzy-C-Means.
This means that K-Means algorithm will produce better clusters compared to Fuzzy -C-Means because lower the DB Index better the clustering.

c) Now a 4th feature 'avg_exam4' is added to the above best 3 features taken from section 2.
Fuzzy C Means algorithm is implemented with 4 features.
DB Indices obtained by Fuzzy C Means algorithms with 4 features are below:

#clusters(k)	**DB Index Fuzzy with 4 features**	**DB Index Fuzzy with 3 features**
2	**0.8441035573750155**	**0.7948460395583281**
3	0.9230175900844416	0.831450582049154
4	1.0814937378224256	1.0707567120338306
5	1.1942721271788956	1.186038187998367
6	1.3319662312021736	1.0058061924802597
7	1.3174860139312237	1.1574216205086396

8	1.4861301494113457	1.1133257604661069
9	1.4251174072622623	1.0285272575460935
10	1.4910371658514805	1.1075236306710674

By comparing the DB Index values of fuzzy c means algorithm with 3 features and 4 features, it can be observed that the DB index values for 3 features are less compared to corresponding DB Index values for 4 features. As lower the DB Index value better is the clustering, adding 4th feature did not improve the clustering result.