COURSE RECOMMENDATION SYSTEM

Recommendation Systems

Prof. Raghuram Bhardwaj

159	165	2023-02-25 16:23:18	2023-02-25 16:25:04	anonymous	NaN
160	166	2023-02-27 09:46:02	2023-02-27 09:46:57	anonymous	NaN
161	167	2023-02-27 09:47:04	2023-02-27 09:47:42	anonymous	NaN

162 rows × 31 columns

```
[ ]  cols_to_delete = {'ID', 'Start time','Completion
     rs_data.drop(columns=cols_to_delete,inplace=True)
```

DataPreprocessing

Missing Values of Electives Can be handelled by

1. Assigning 0
2. Assigning 0.5
3. Do nothing

```
[ ]  rs_data_final = rs_data.fillna(0) #Filling Empty Values By 0
     # rs_data_final = rs_data.fillna(0.5) #Filling Empty Values By 0.5
     #rs_data_final = rs_data #Do nothing
```

---------- DEMO TRYING SOMETHING ---------

```
[ ]  rs_data_final
```

```
                Data
```

DATA PREPROCESSING

DELETING UNNECESSARY COLUMNS

These columns were part of filling the form. We are working with purely anonymous data so hence, it is not needed

DEALING WITH NaN VALUES

We decided to go with 3 ways to deal with it

Replacing with 0
Replacing with 0.5
Ignoring them

ALGORITHMS TRIED

K MEANS & SVD

Unsupervised learning algorithm that clusters and reduces the noise

DECISION TREE

This is also unsupervised which is counter intuitive since it usually has a target variable but in this case it acts like a 32 point cluster

(5 subjects with 2 possible choices per subject)

1	0.70	0.80	0.20	0.80	0.80
2	0.70	0.00	1.00	1.00	1.00
3	0.65	0.70	0.20	0.75	0.75
4	0.70	0.60	0.20	0.00	0.80

5 rows × 25 columns

```
# Replacing NaN
#dataset = dataset.fillna(0)
```

```
#dataset.head()
```

```
c1y = dataset[dataset['Data Structures and Algorithms'] >= 0.5
c1n = dataset[dataset['Data Structures and Algorithms'] < 0.5]
dataset=[] # freeing memory
```

```
#c1y.describe()
```

```
#c1n.describe()
```

```
c1yc2y = c1y[c1y['Computer Architecture'] >= 0.5]
c1yc2n = c1y[c1y['Computer Architecture'] < 0.5]
c1y = [] # freeing memory
```

```
c1nc2y = c1n[c1n['Computer Architecture'] >= 0.5]
c1nc2n = c1n[c1n['Computer Architecture'] < 0.5]
c1n = [] # freeing memory
```

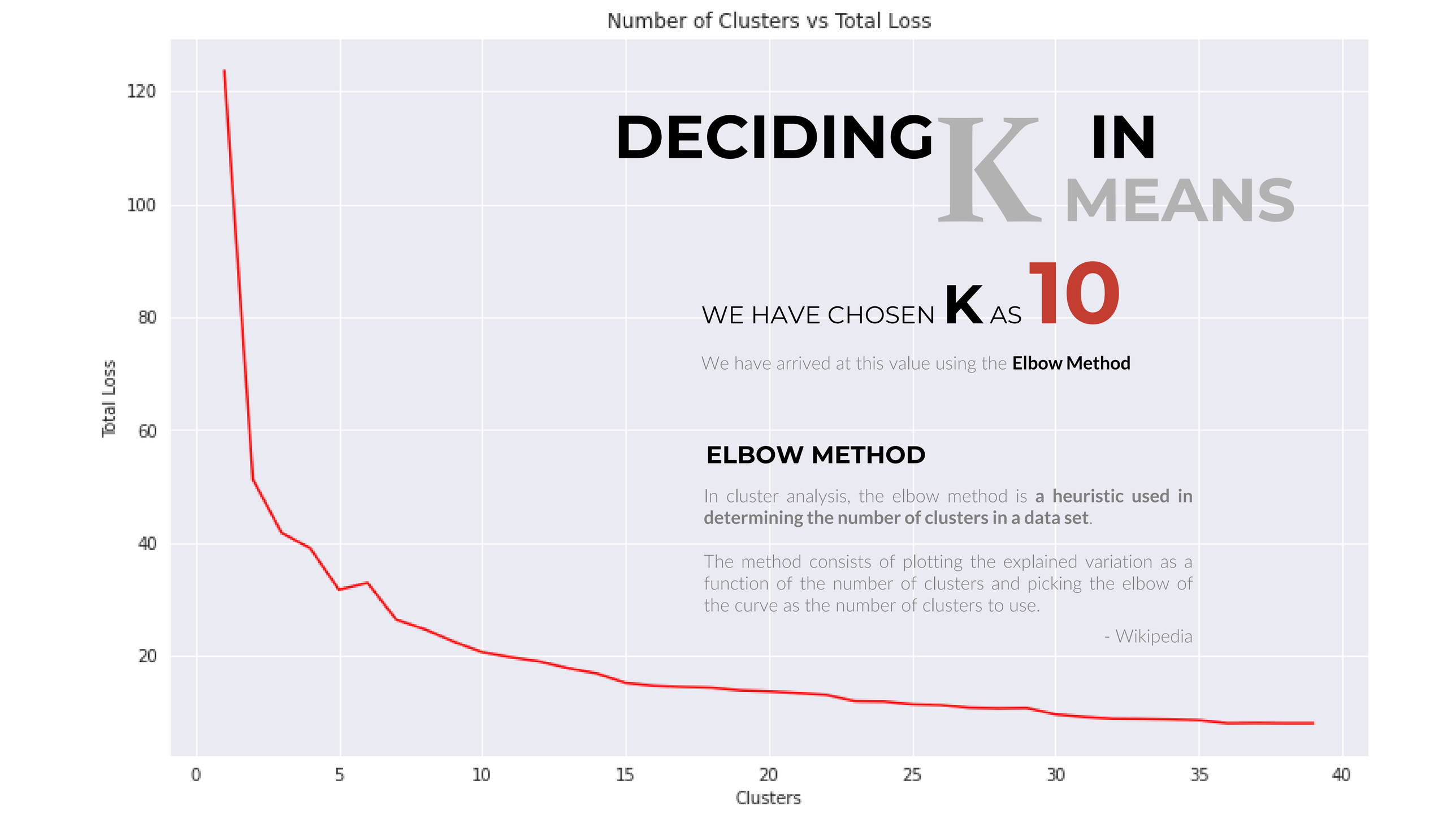
```
c1yc2yc3y = c1yc2y[c1yc2y['Discrete Mathematics\n'] >= 0.5]
c1yc2yc3n = c1yc2y[c1yc2y['Discrete Mathematics\n'] < 0.5]
```

OBSERVATIONS

DECIDING **K** FOR **K MEANS**
K MEANS WITHOUT **SVD**
K MEANS WITH **SVD**
DECISION TREE LEAFS

&

Their variations



DECIDING K IN MEANS

WE HAVE CHOSEN **K** AS **10**

We have arrived at this value using the **Elbow Method**

ELBOW METHOD

In cluster analysis, the elbow method is **a heuristic used in determining the number of clusters in a data set**.

The method consists of plotting the explained variation as a function of the number of clusters and picking the elbow of the curve as the number of clusters to use.

- Wikipedia

k-means be like:

RUNNING K MEANS WITHOUT SVD

We analyzed the performance of K Means algorithm

Coding KMeans

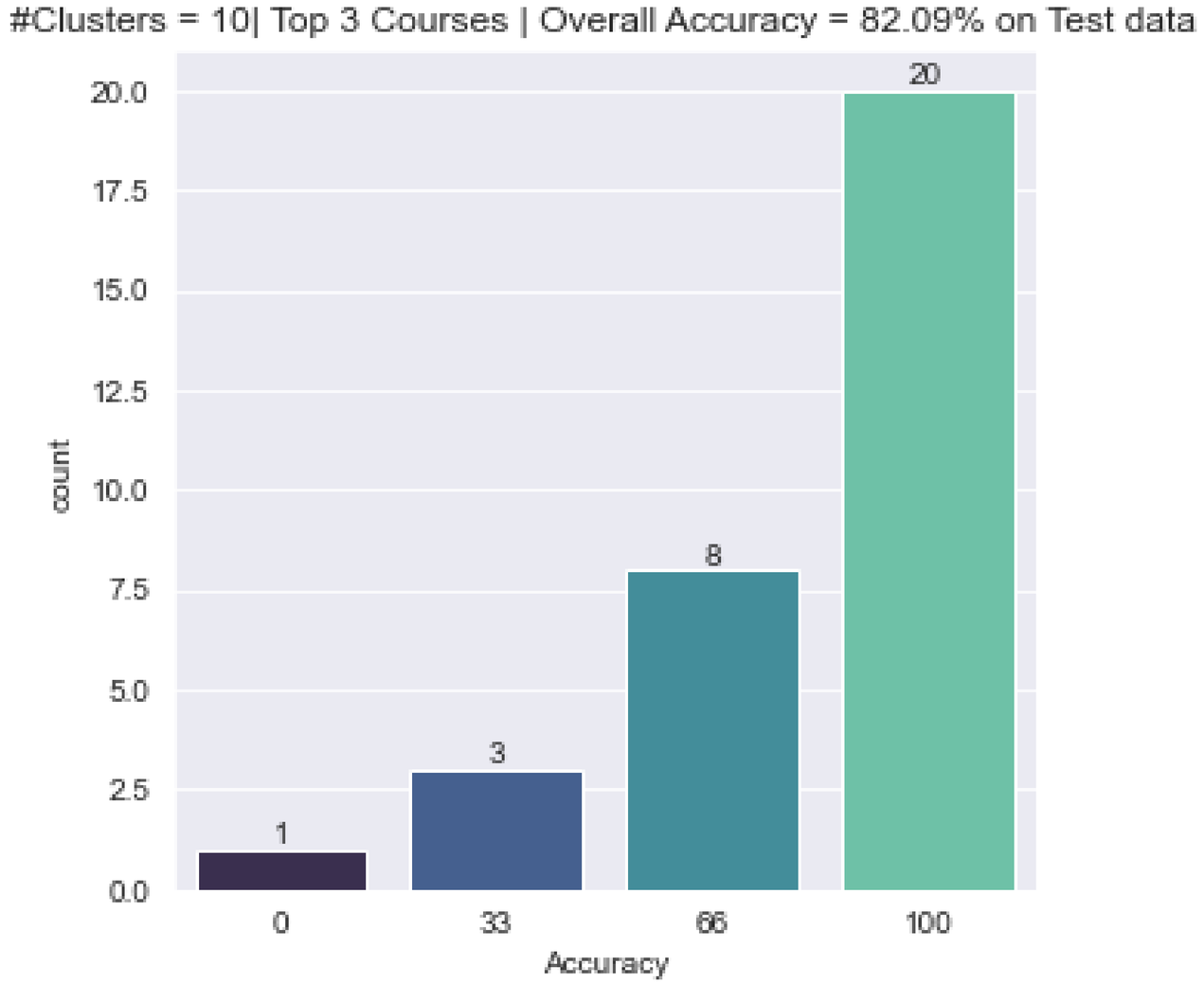
```
[ ]  # columns={'Course1','Course2','Course3','Course4','Course5'}
     # dataa = pd.DataFrame(np.random.randint(101,size=(250,5))/10,columns=columns)
     dataa = rs_data_final[:-32].copy()
     dataa_test = rs_data_final[-32:].copy()

     # dataa = rs_data_final.copy()
```

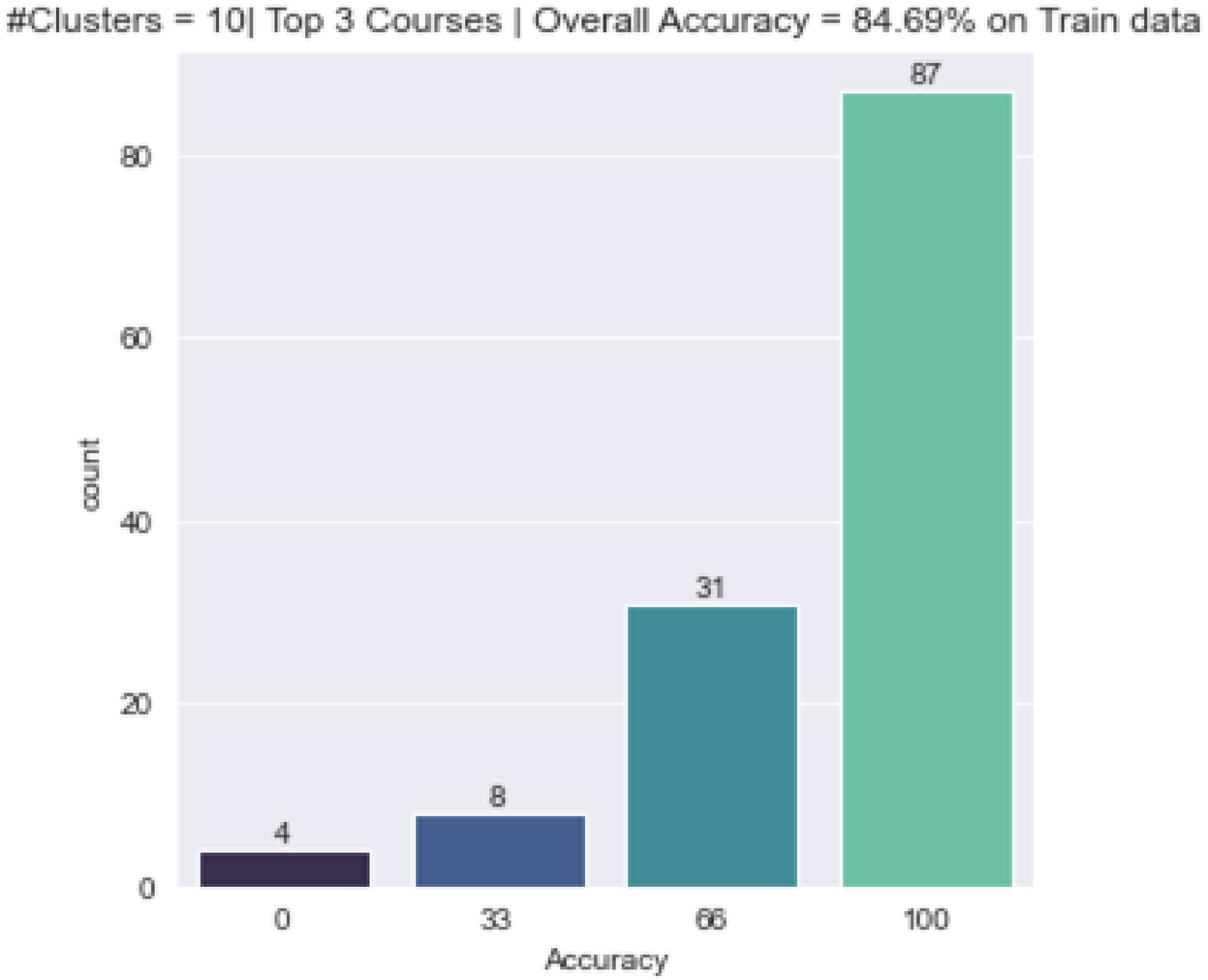
```
[ ]  dataa
```

```
[ ]  def init():
       '''1)Initialising K Clusters from the first k datapoints with the 5 core courses
          2)Assigning Random clusters to each datapoint '''
       for i in range(k):
         clusters.append(np.array(dataa.iloc[i][:5]))

       print("-----------------------------------------------INIT------------------")
       print("No of Clusters : ",len(clusters))
      # print("## LOG : init : Final Clusters : ",clusters)

       initial_clusters = np.random.randint(0,k,len(dataa))
       dataa['cluster'] = initial_clusters

     def findLoss(centroid,data):
       sum=0
```

RUNNING K MEANS *WITHOUT SVD*

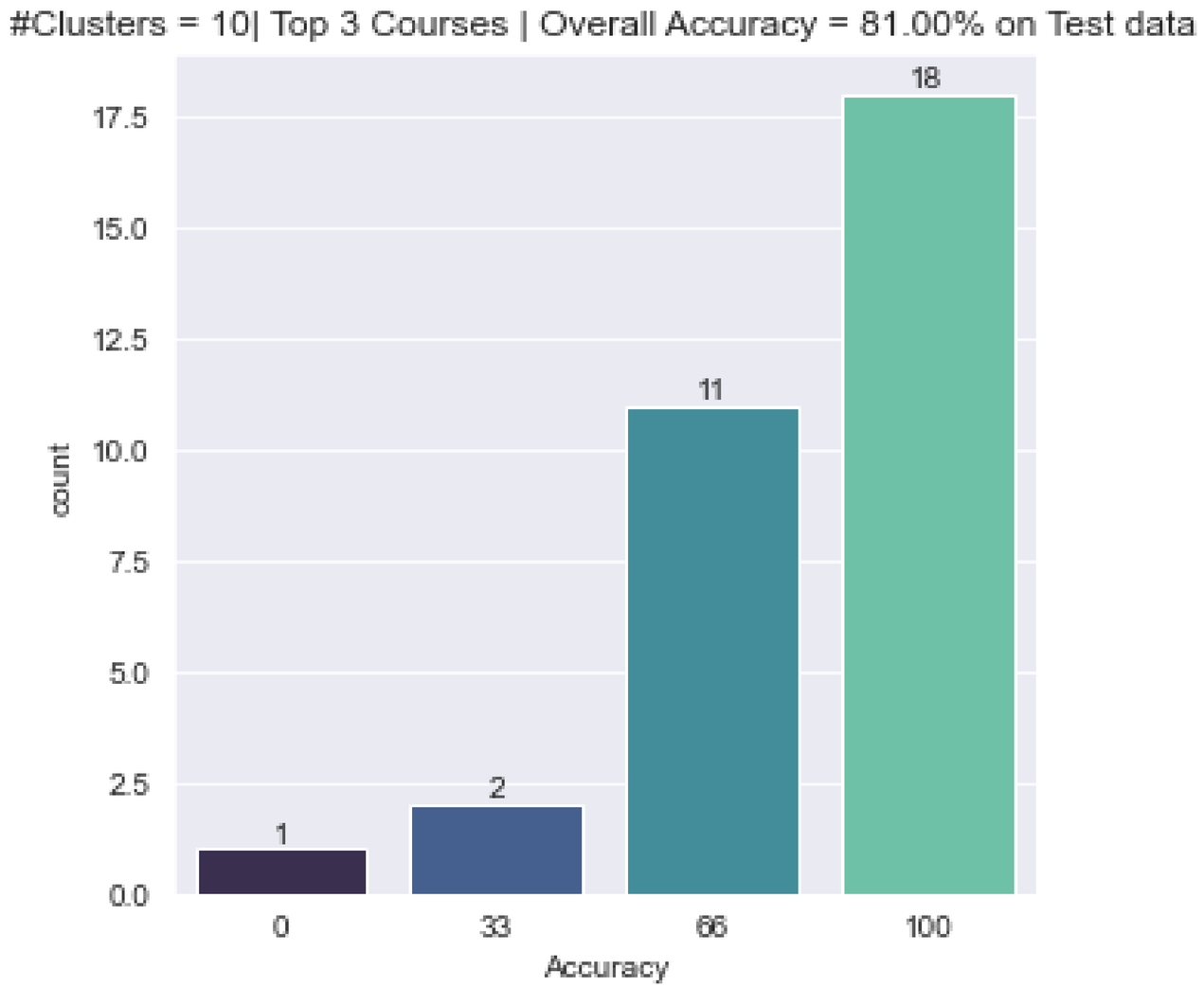
FILLED *MISSING VALUES WITH 0*

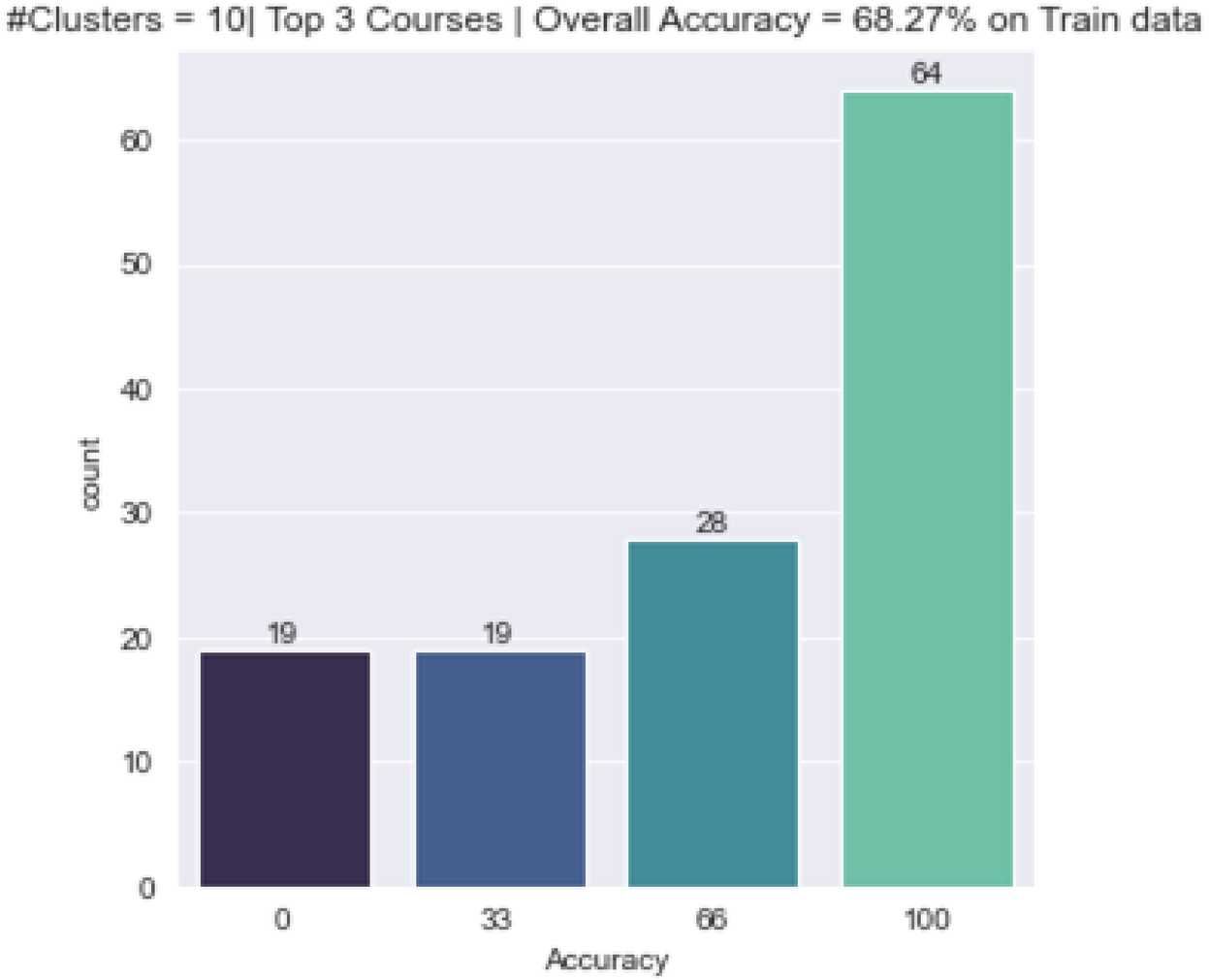




RUNNING K MEANS *WITHOUT SVD*

FILLED *MISSING VALUES WITH 0.5*





RUNNING K MEANS *WITHOUT SVD*

IGNORED THE *MISSING VALUES*

RUNNING K MEANS WITH SVD

Now we add SVD to try and improve the performance

SVD

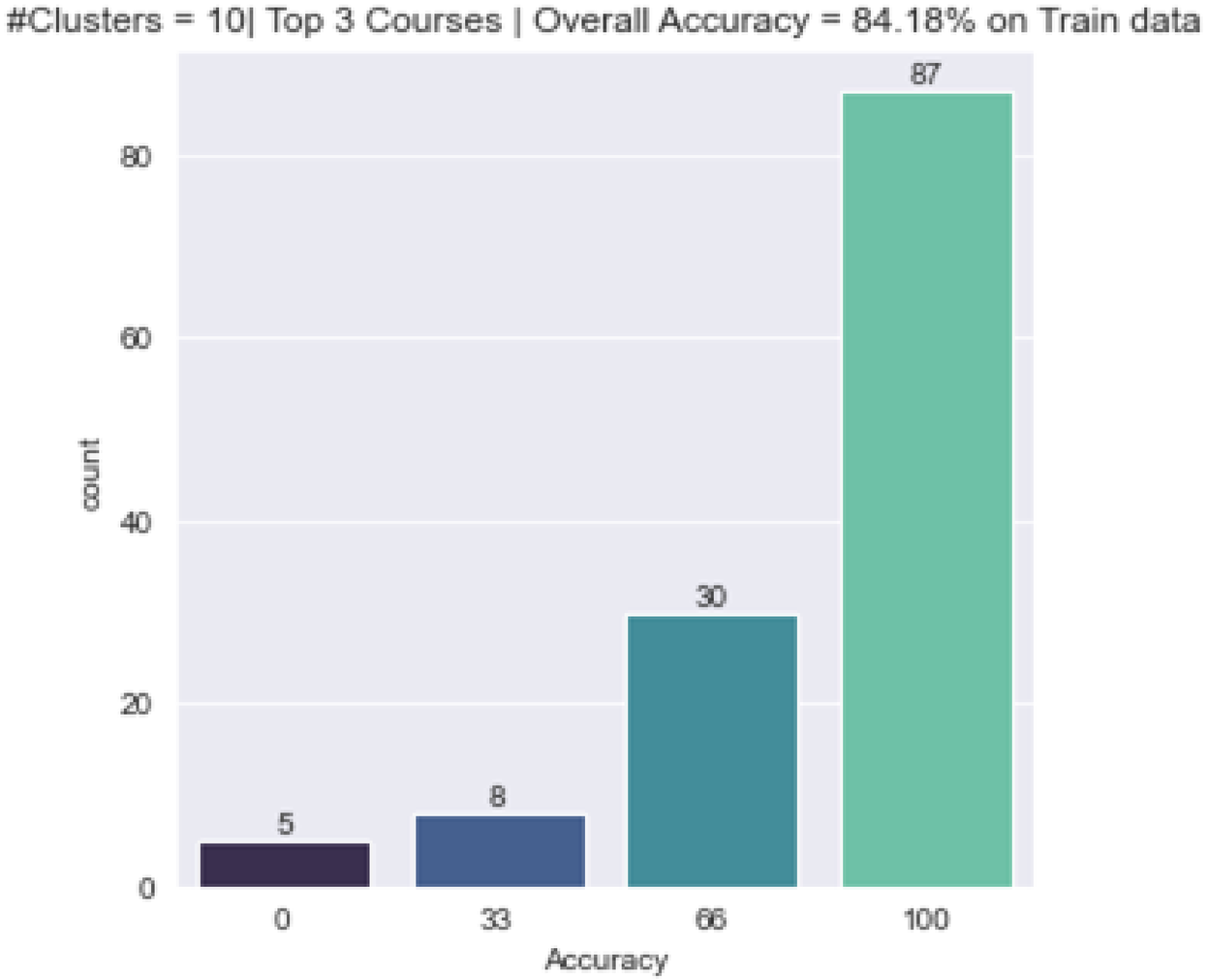
```
[ ] #Implement SVD From Here on l

[ ] u, s, vt = np.linalg.svd(mega_rec_tab)

[ ] print(u.shape)
    print(s.shape)
    print(vt.shape)

[ ] s

    array([6.79292858, 1.66094137, 1.38003871, 0.97159449, 0.669356  ,
           0.61723529, 0.48605931, 0.39372649, 0.28704915, 0.15638577])

[ ] s = s[:-2]
    s

    array([6.79292858, 1.66094137, 1.38003871, 0.97159449, 0.669356  ,
           0.61723529, 0.48605931, 0.39372649])
```

OUR SIGNMA VALUES ARE

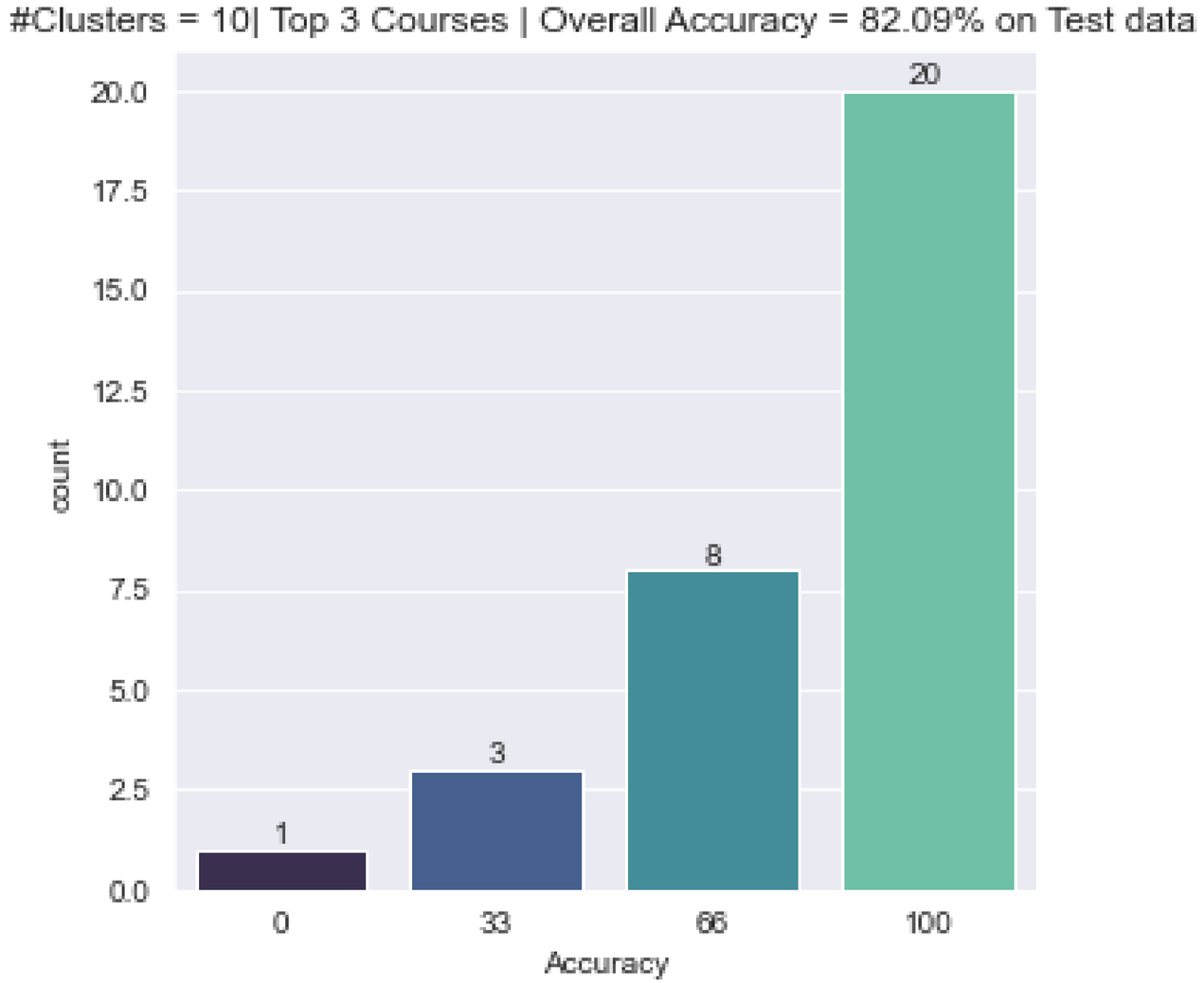
6.79633422
1.6547264
1.3751391
0.97743666
0.67857574
0.61757177
0.48787547
0.39326742
0.28305541
0.15863474

```
Sigma = np.diagflat(s)
Z1 = np.zeros((u.shape[1]-Sigma.shape[0],Sigma.shape[1]))
Sigma = np.vstack((Sigma,Z1))
Z2 = np.zeros((Sigma.shape[0],vt.shape[0]-Sigma.shape[1]))
Sigma = np.hstack((Sigma,Z2))
Sigma
```

```
array([[6.79292858, 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ],
       [0.        , 1.66094137, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ],
       [0.        , 0.        , 1.38003871, 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
```



FILLED *MISSING VALUES WITH 0*



RUNNING K MEANS

WITH SVD *1 EIGEN VALUE REMOVED*

WITHOUT SVD

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.683333
Machine Learning\n                    0.676667
Mathematics For Machine Learning\n    0.633333
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.683333
Machine Learning\n                    0.676667
Mathematics For Machine Learning\n    0.633333
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.600000
Mathematics For Machine Learning\n    0.533333
Natural Language Processing\n         0.333333
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Machine Learning\n                    1.0
Mathematics For Machine Learning\n    1.0
Visual Recognition\n                  0.8
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

Machine Learning\n                    0.661111
The Web and the Mind                  0.555556
Mathematics For Machine Learning\n    0.505556
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.683333
Machine Learning\n                    0.676667
Mathematics For Machine Learning\n    0.633333
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.683333
Machine Learning\n                    0.676667
Mathematics For Machine Learning\n    0.633333
Name: 4, dtype: float64
```

OBSERVATIONS :

Result Same for With SVD and without SVD

Overall accuracy for Train and Test is same for with SVD and without SVD

Most popular electives are recommended for majority cases.

Ex : ML, MML

WITH SVD

1 EIGEN VALUE REMOVED

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.705711
Software Production Engineering\n     0.670001
Mathematics For Machine Learning\n    0.645567
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.705711
Software Production Engineering\n     0.670001
Mathematics For Machine Learning\n    0.645567
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.591088
Mathematics For Machine Learning\n    0.537985
Software Production Engineering\n     0.328470
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Mathematics For Machine Learning\n    0.999184
Machine Learning\n                    0.975933
Visual Recognition\n                  0.745199
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

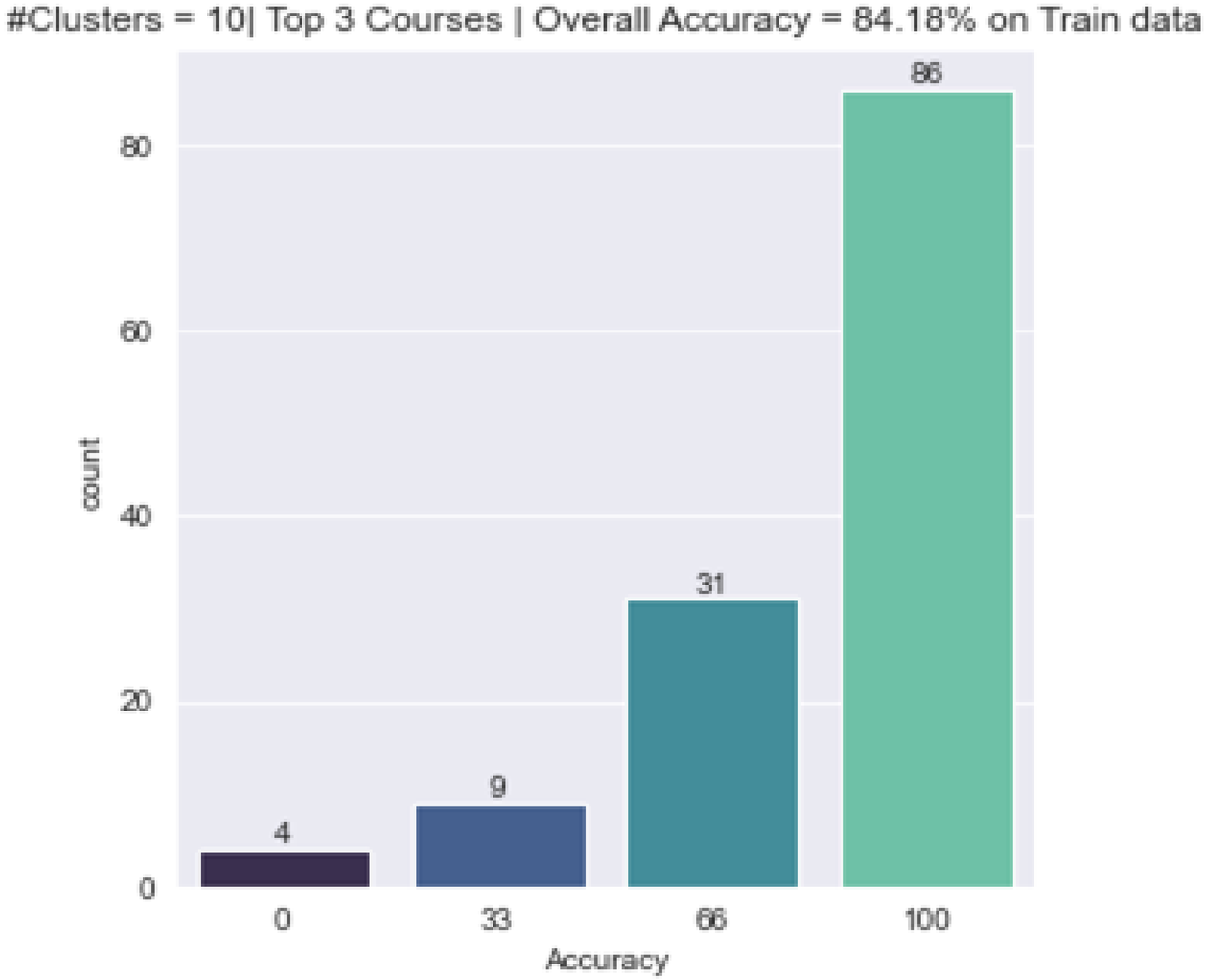
```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

Machine Learning\n                    0.669887
The Web and the Mind                  0.569028
Mathematics For Machine Learning\n    0.502772
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.705711
Software Production Engineering\n     0.670001
Mathematics For Machine Learning\n    0.645567
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.705711
Software Production Engineering\n     0.670001
Mathematics For Machine Learning\n    0.645567
Name: 4, dtype: float64
```



FILLED *MISSING VALUES WITH 0*

RUNNING K MEANS

WITH SVD *2 EIGEN VALUE REMOVED*

WITHOUT SVD

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.683333
Machine Learning\n                    0.676667
Mathematics For Machine Learning\n    0.633333
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.683333
Machine Learning\n                    0.676667
Mathematics For Machine Learning\n    0.633333
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.600000
Mathematics For Machine Learning\n    0.533333
Natural Language Processing\n         0.333333
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Machine Learning\n                    1.0
Mathematics For Machine Learning\n    1.0
Visual Recognition\n                  0.8
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

Machine Learning\n                    0.661111
The Web and the Mind                  0.555556
Mathematics For Machine Learning\n    0.505556
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.683333
Machine Learning\n                    0.676667
Mathematics For Machine Learning\n    0.633333
Name: 4, dtype: float64
```

OBSERVATIONS :

Result Same for With SVD and without SVD

Overall accuracy for Train and Test is same for with SVD and without SVD

Most popular electives are recommended for majority cases.

Ex : ML, MML

WITH SVD

2 EIGEN VALUE REMOVED

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.705711
Software Production Engineering\n     0.670001
Mathematics For Machine Learning\n    0.645567
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.705711
Software Production Engineering\n     0.670001
Mathematics For Machine Learning\n    0.645567
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.591088
Mathematics For Machine Learning\n    0.537985
Software Production Engineering\n     0.328470
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

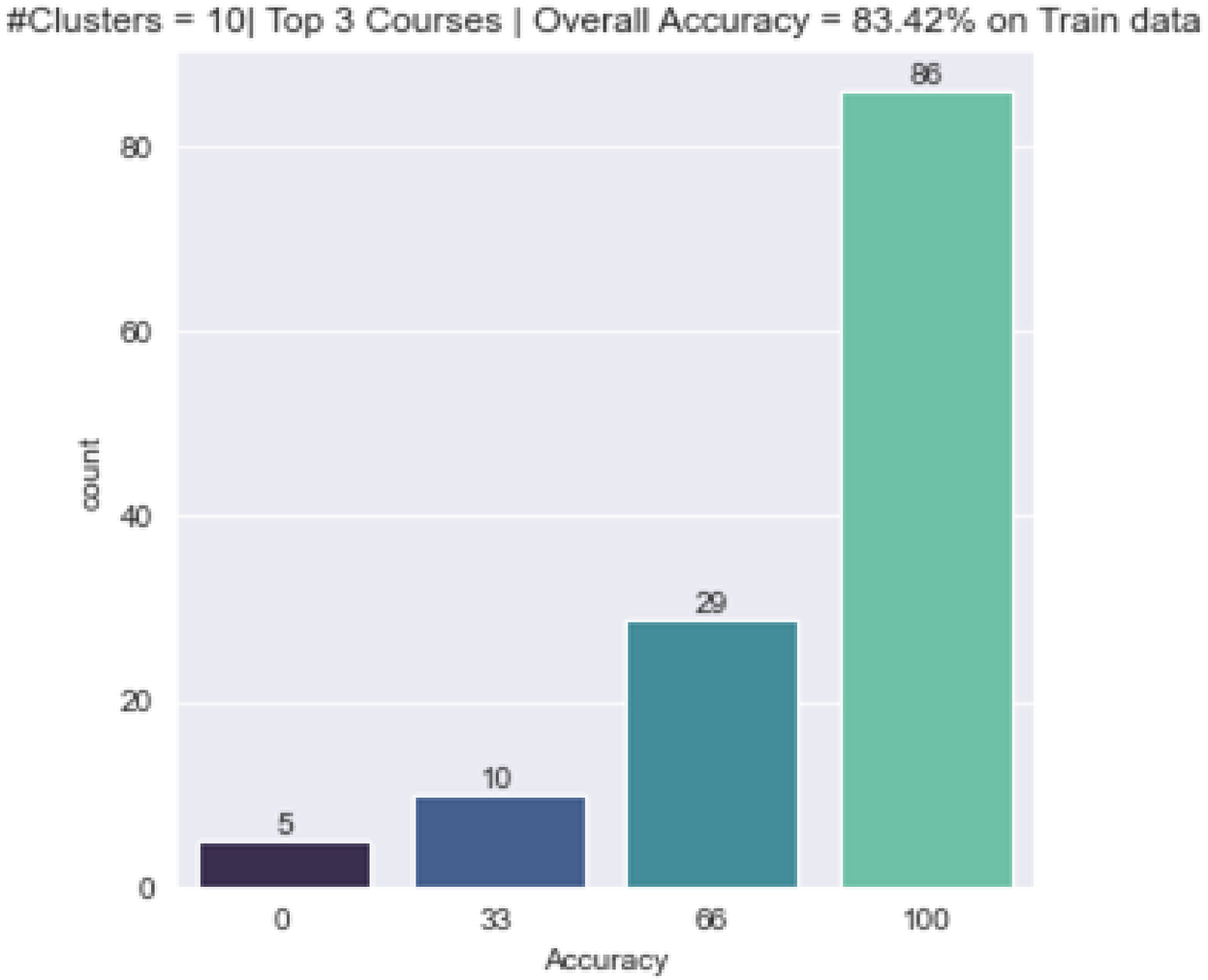
```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Mathematics For Machine Learning\n    0.999184
Machine Learning\n                    0.975933
Visual Recognition\n                  0.745199
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

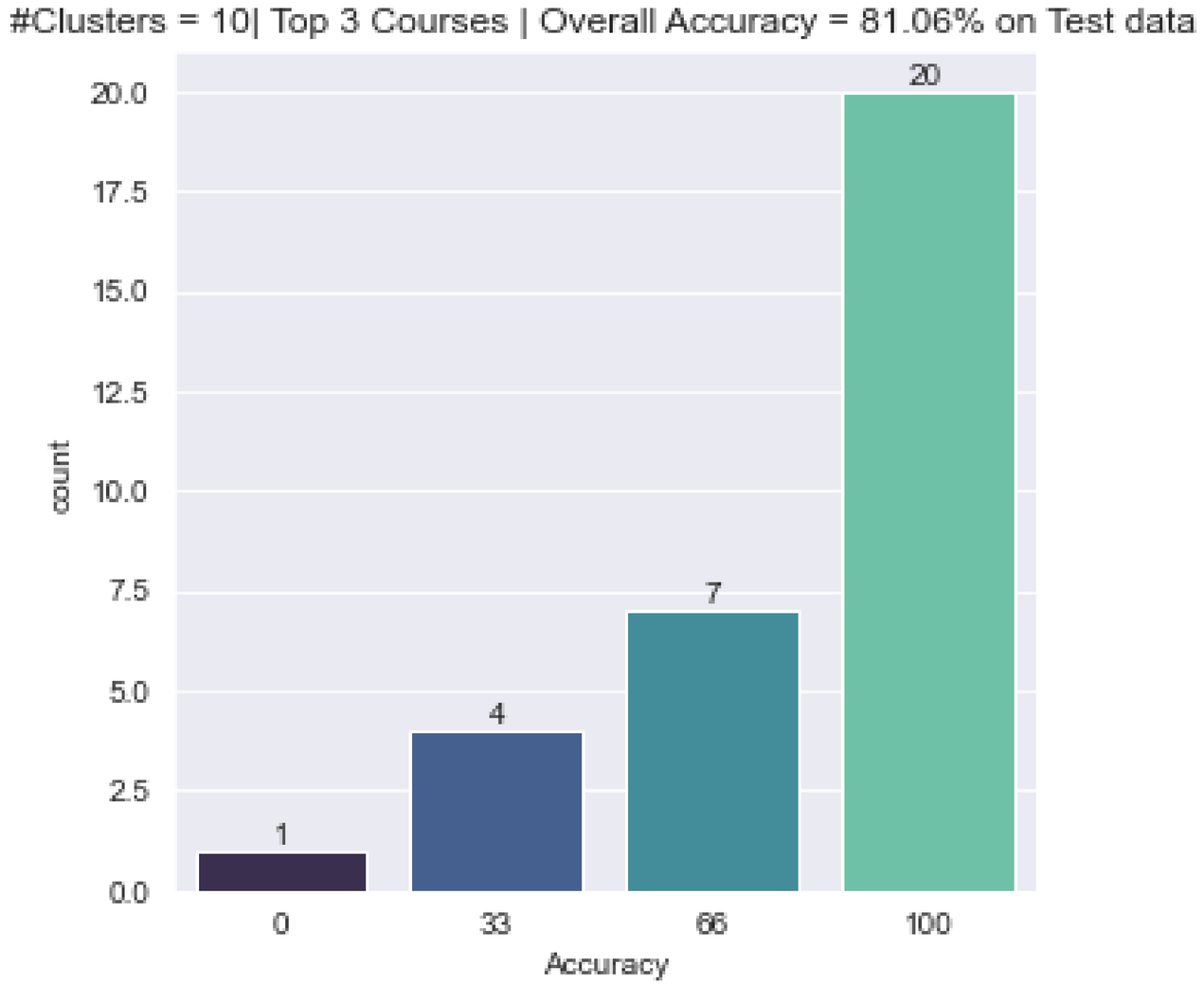
```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

Machine Learning\n                    0.669887
The Web and the Mind                  0.569028
Mathematics For Machine Learning\n    0.502772
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.705711
Software Production Engineering\n     0.670001
Mathematics For Machine Learning\n    0.645567
Name: 4, dtype: float64
```



FILLED *MISSING VALUES WITH 0*



RUNNING K MEANS

WITH SVD *4 EIGEN VALUE REMOVED*

WITHOUT SVD

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.683333
Machine Learning\n                    0.676667
Mathematics For Machine Learning\n    0.633333
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.683333
Machine Learning\n                    0.676667
Mathematics For Machine Learning\n    0.633333
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.600000
Mathematics For Machine Learning\n    0.533333
Natural Language Processing\n         0.333333
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Machine Learning\n                    1.0
Mathematics For Machine Learning\n    1.0
Visual Recognition\n                  0.8
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

Machine Learning\n                    0.661111
The Web and the Mind                  0.555556
Mathematics For Machine Learning\n    0.505556
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.683333
Machine Learning\n                    0.676667
Mathematics For Machine Learning\n    0.633333
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.683333
Machine Learning\n                    0.676667
Mathematics For Machine Learning\n    0.633333
```

OBSERVATIONS :

Result different for With SVD and without SVD

Overall accuracy has downgraded due to removing more eigen values

WITH SVD

4 EIGEN VALUE REMOVED

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.796975
Mathematics For Machine Learning\n    0.700291
Software Production Engineering\n     0.646739
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.796975
Mathematics For Machine Learning\n    0.700291
Software Production Engineering\n     0.646739
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.573159
Mathematics For Machine Learning\n    0.527234
Software Production Engineering\n     0.333040
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Mathematics For Machine Learning\n    0.984122
Machine Learning\n                    0.950813
Programming Languages\n               0.727814
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

Machine Learning\n                    0.621865
Software Production Engineering\n     0.512421
The Web and the Mind                  0.509016
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

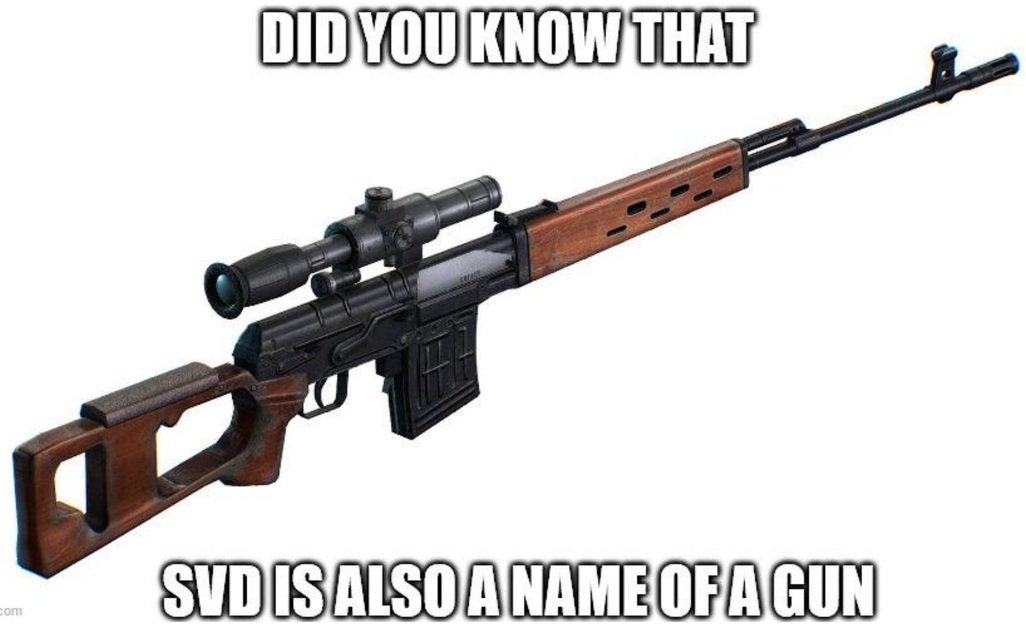
```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.796975
Mathematics For Machine Learning\n    0.700291
Software Production Engineering\n     0.646739
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.796975
Mathematics For Machine Learning\n    0.700291
Software Production Engineering\n     0.646739
Name: 4, dtype: float64
```

RUNNING K MEANS WITH SVD

Changing what we fill for NaN values in the dataset to **0.5**



SVD

```
u, s, vt = np.linalg.svd(mega_rec_tab)
```

```
print(u.shape)
print(s.shape)
print(vt.shape)
```

```
(10, 10)
(10,)
(25, 25)
```

```
s
```

```
array([8.49309849, 1.71808388, 1.23823303, 0.88985685, 0.59149522,
       0.51506093, 0.45824432, 0.35739363, 0.2254367 , 0.20019186])
```

```
s = s[:-2]
s
```

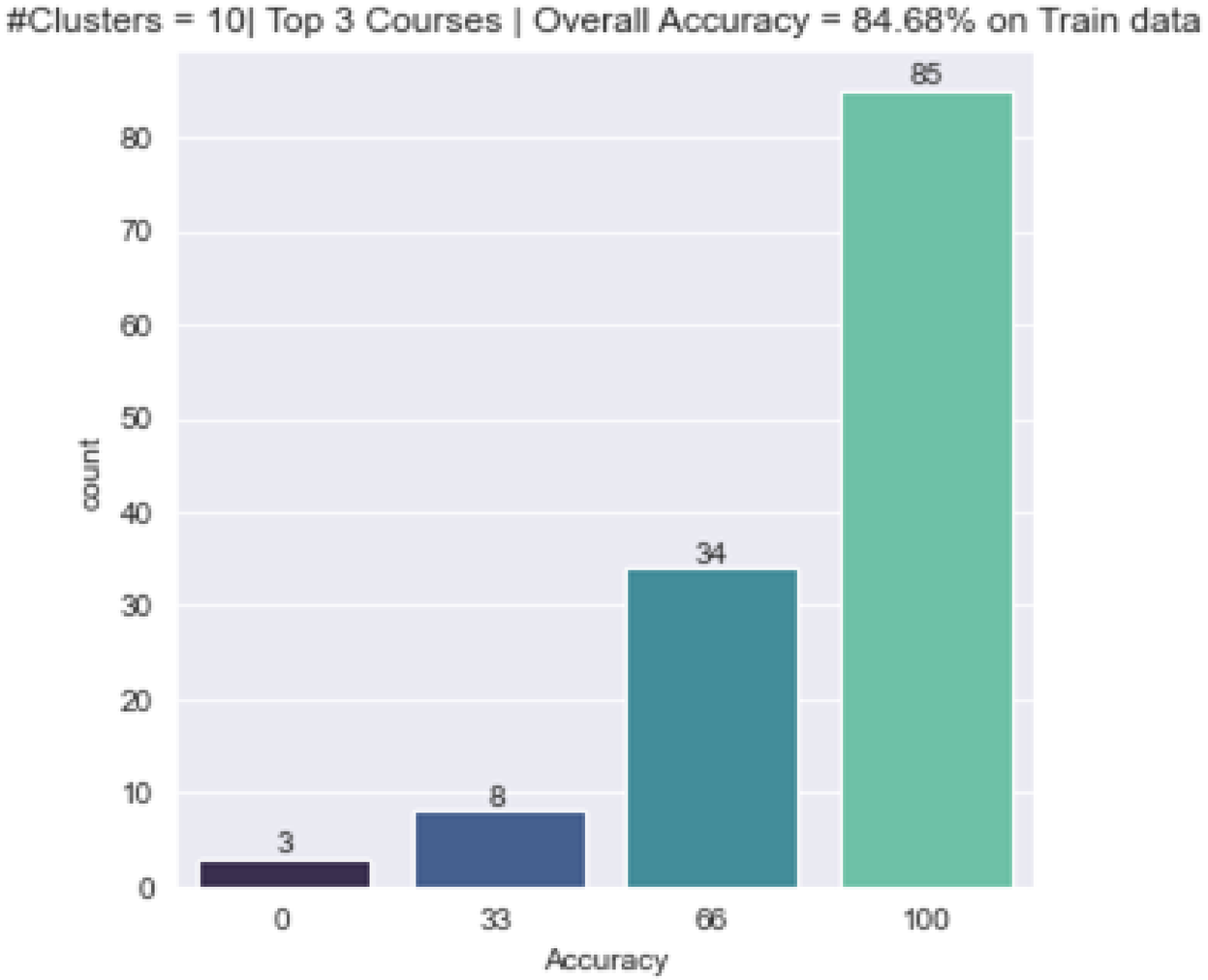
```
array([8.49309849, 1.71808388, 1.23823303, 0.88985685, 0.59149522,
       0.51506093, 0.45824432, 0.35739363])
```

```
Sigma = np.diagflat(s)
Z1 = np.zeros((u.shape[1]-Sigma.shape[0],Sigma.shape[1]))
Sigma = np.vstack((Sigma,Z1))
Z2 = np.zeros((Sigma.shape[0],vt.shape[0]-Sigma.shape[1]))
Sigma = np.hstack((Sigma,Z2))
Sigma
```

```
array([[8.49309849, 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ],
       [0.        , 1.71808388, 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ],
       [0.        , 0.        , 1.23823303, 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ],
```

OUR SIGMA VALUES ARE

8.49309849
1.71808388
1.23823303
0.88985685
0.59149522
0.51506093
0.45824432
0.35739363
0.224367
0.20019186



FILLED *MISSING VALUES WITH 0.5*

RUNNING K MEANS

WITH SVD *1 EIGEN VALUE REMOVED*

WITHOUT SVD

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.766667
Mathematics For Machine Learning\n    0.700000
Natural Language Processing\n         0.666667
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Machine Learning\n                    1.0
Mathematics For Machine Learning\n    1.0
Visual Recognition\n                  0.8
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

The Web and the Mind                  0.722222
Machine Learning\n                    0.716667
Software Production Engineering\n     0.611111
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

OBSERVATIONS :

Result Same for With SVD and without SVD

Overall accuracy for Train and Test is same for with SVD and without SVD

Most popular electives are recommended for majority cases.

Ex : ML

WITH SVD

1 EIGEN VALUE REMOVED

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.747089
Machine Learning\n                    0.720882
Mathematics For Machine Learning\n    0.665300
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.747089
Machine Learning\n                    0.720882
Mathematics For Machine Learning\n    0.665300
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.760137
Mathematics For Machine Learning\n    0.704714
Software Production Engineering\n     0.666134
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Mathematics For Machine Learning\n    1.001881
Machine Learning\n                    0.991604
Visual Recognition\n                  0.789877
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

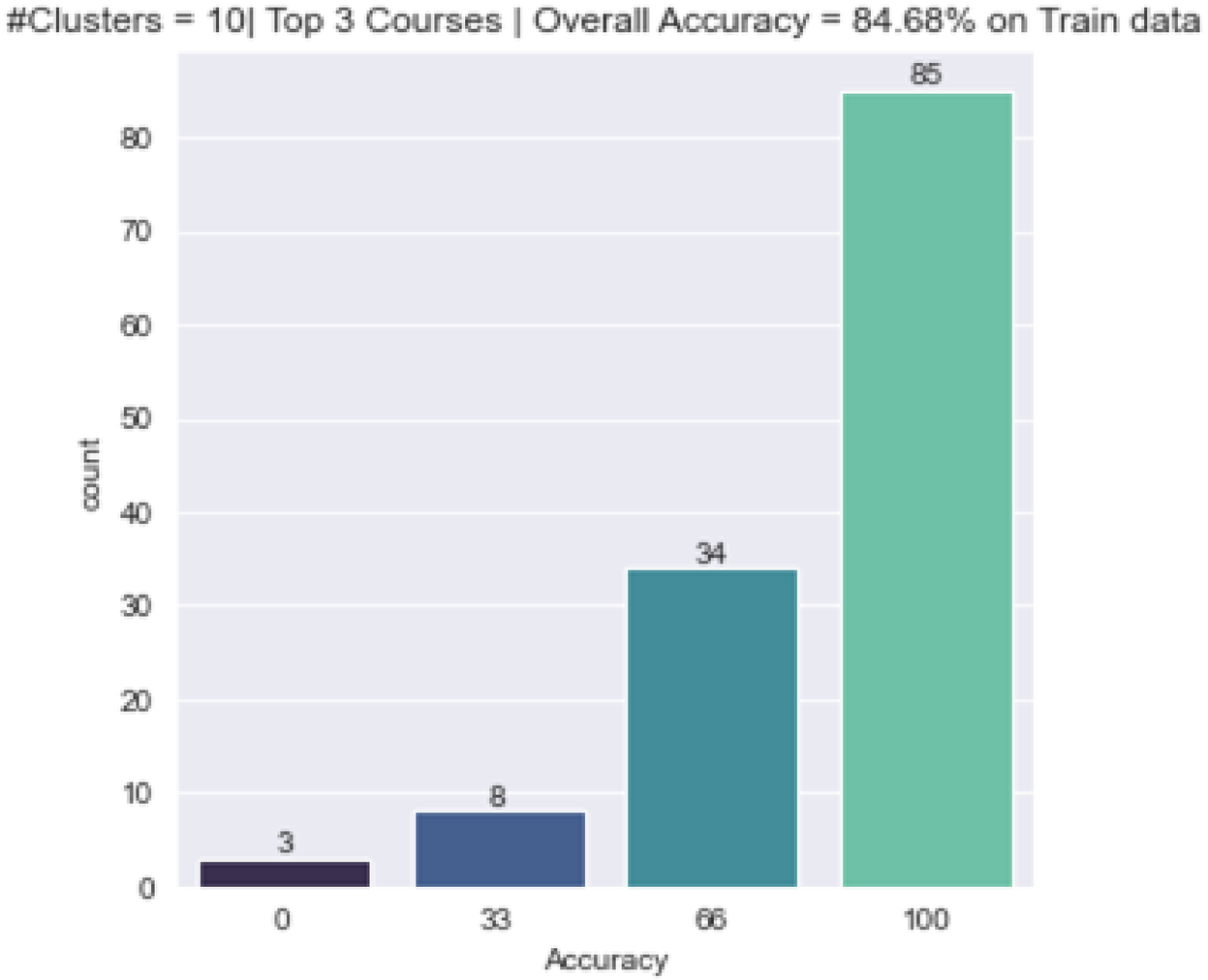
```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

Machine Learning\n                    0.735670
The Web and the Mind                  0.709805
Software Production Engineering\n     0.608822
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.747089
Machine Learning\n                    0.720882
Mathematics For Machine Learning\n    0.665300
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.747089
Machine Learning\n                    0.720882
Mathematics For Machine Learning\n    0.665300
Name: 4, dtype: float64
```



FILLED *MISSING VALUES WITH 0.5*

RUNNING K MEANS

WITH SVD *2 EIGEN VALUE REMOVED*

WITHOUT SVD

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.766667
Mathematics For Machine Learning\n    0.700000
Natural Language Processing\n         0.666667
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Machine Learning\n                    1.0
Mathematics For Machine Learning\n    1.0
Visual Recognition\n                  0.8
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

The Web and the Mind                  0.722222
Machine Learning\n                    0.716667
Software Production Engineering\n     0.611111
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

OBSERVATIONS :

Result Same for With SVD and without SVD

Overall accuracy for Train and Test is same for with SVD and without SVD

Most popular electives are recommended for majority cases.

Ex : ML

WITH SVD

2 EIGEN VALUE REMOVED

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.747089
Machine Learning\n                    0.720882
Mathematics For Machine Learning\n    0.665300
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.747089
Machine Learning\n                    0.720882
Mathematics For Machine Learning\n    0.665300
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.760137
Mathematics For Machine Learning\n    0.704714
Software Production Engineering\n     0.666134
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Mathematics For Machine Learning\n    1.001881
Machine Learning\n                    0.991604
Visual Recognition\n                  0.789877
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

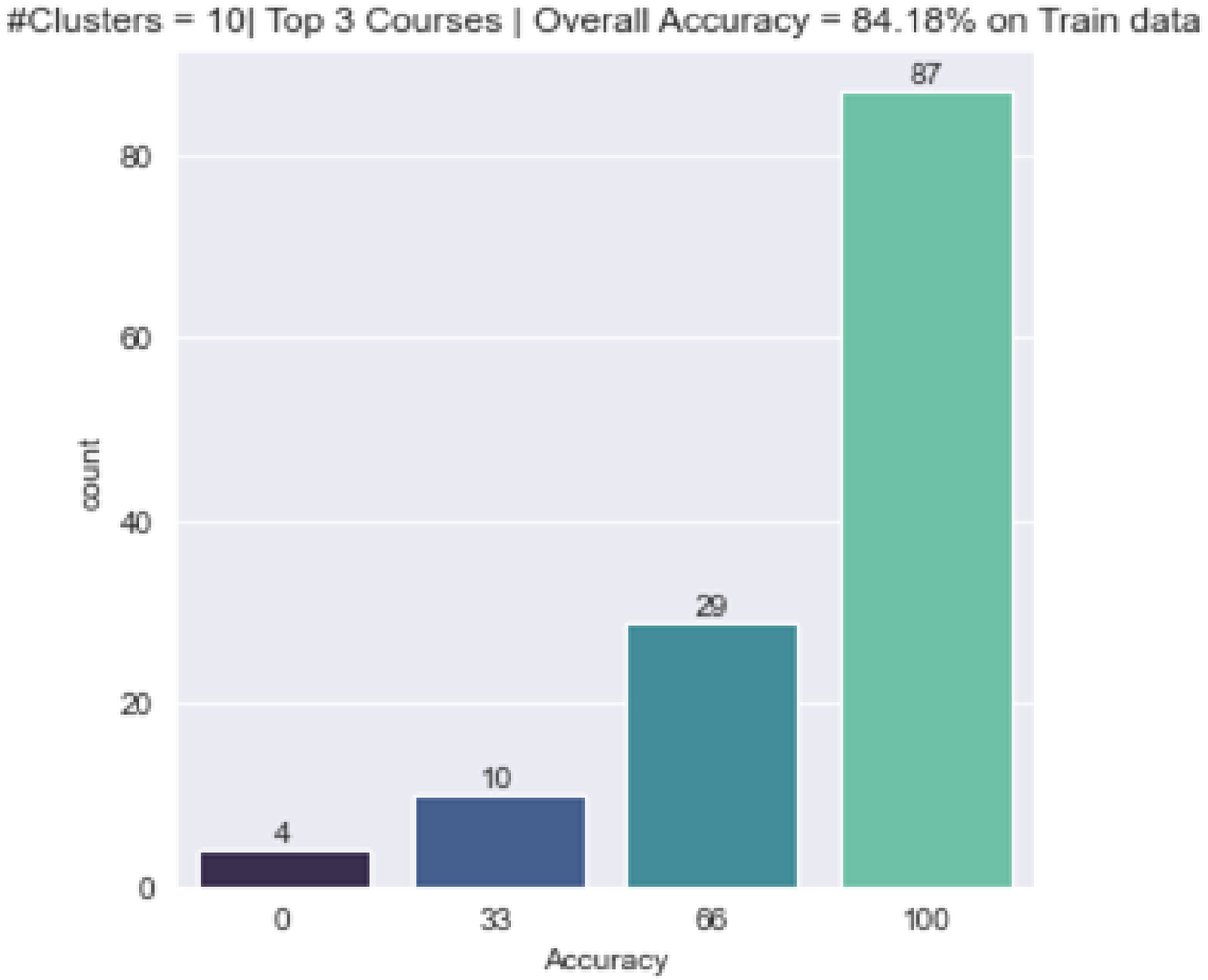
```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

Machine Learning\n                    0.735670
The Web and the Mind                  0.709805
Software Production Engineering\n     0.608822
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.747089
Machine Learning\n                    0.720882
Mathematics For Machine Learning\n    0.665300
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.747089
Machine Learning\n                    0.720882
Mathematics For Machine Learning\n    0.665300
Name: 4, dtype: float64
```



FILLED *MISSING VALUES WITH 0.5*

RUNNING K MEANS

WITH SVD *4 EIGEN VALUE REMOVED*

WITHOUT SVD

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.766667
Mathematics For Machine Learning\n    0.700000
Natural Language Processing\n         0.666667
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Machine Learning\n                    1.0
Mathematics For Machine Learning\n    1.0
Visual Recognition\n                  0.8
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

The Web and the Mind                  0.722222
Machine Learning\n                    0.716667
Software Production Engineering\n     0.611111
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.750000
Machine Learning\n                    0.710000
Mathematics For Machine Learning\n    0.666667
Name: 4, dtype: float64
```

OBSERVATIONS :

Result Same for With SVD and without SVD

Overall accuracy for Test has improved using SVD

Most popular electives are recommended for majority cases.

Ex : ML

WITH SVD

4 EIGEN VALUE REMOVED

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.802106
Mathematics For Machine Learning\n    0.697481
Software Production Engineering\n     0.669020
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.802106
Mathematics For Machine Learning\n    0.697481
Software Production Engineering\n     0.669020
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Machine Learning\n                    0.776066
Mathematics For Machine Learning\n    0.713474
Software Production Engineering\n     0.652557
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Mathematics For Machine Learning\n    0.978958
Machine Learning\n                    0.952185
Programming Languages\n               0.758476
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

Machine Learning\n                    0.693403
The Web and the Mind                  0.675482
Software Production Engineering\n     0.648665
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.802106
Mathematics For Machine Learning\n    0.697481
Software Production Engineering\n     0.669020
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.802106
Mathematics For Machine Learning\n    0.697481
Software Production Engineering\n     0.669020
Name: 4, dtype: float64
```

RUNNING K MEANS WITH SVD

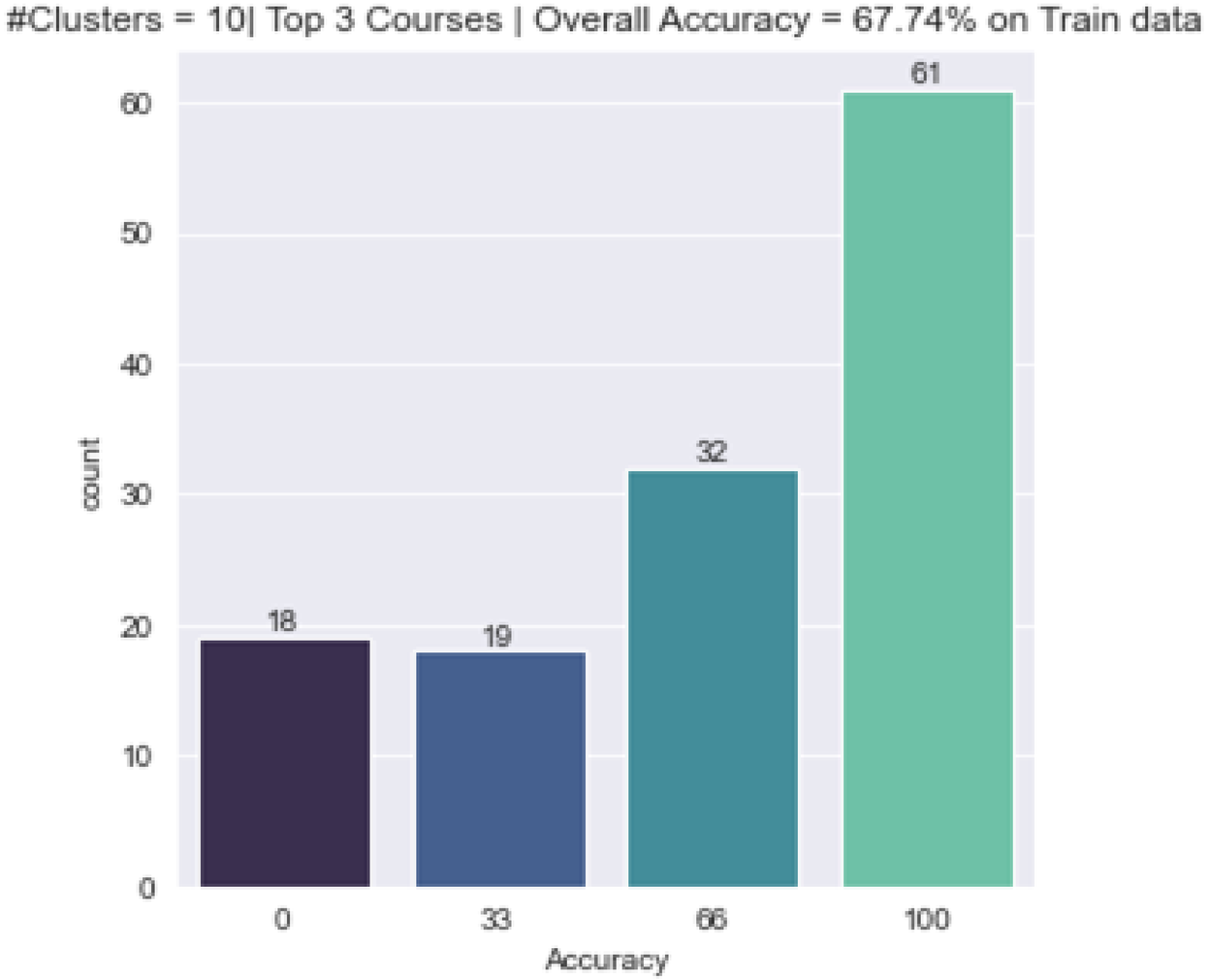
We have **ignored** NaN from the calculation

SVD

```
[88] u, s, vt = np.linalg.svd(mega_rec_tab)

[89] print(u.shape)
     print(s.shape)
     print(vt.shape)

     (10, 10)
     (10,)
     (25, 25)

[90] s

     array([8.33161985, 2.06817611, 1.49561973, 1.11953002, 0.97281152,
            0.622232  , 0.54932207, 0.45532983, 0.39007484, 0.2887603 ])

[91] s = s[:-2]
     s

     array([8.33161985, 2.06817611, 1.49561973, 1.11953002, 0.97281152,
            0.622232  , 0.54932207, 0.45532983])

     Sigma = np.diagflat(s)
     Z1 = np.zeros((u.shape[1]-Sigma.shape[0],Sigma.shape[1]))
     Sigma = np.vstack((Sigma,Z1))
     Z2 = np.zeros((Sigma.shape[0],vt.shape[0]-Sigma.shape[1]))
     Sigma = np.hstack((Sigma,Z2))
     Sigma

     array([[8.33161985, 0.        , 0.        , 0.        , 0.        ,
             0.        , 0.        , 0.        , 0.        , 0.        ,
             0.        , 0.        , 0.        , 0.        , 0.        ,
             0.        , 0.        , 0.        , 0.        , 0.        ,
             0.        , 0.        , 0.        , 0.        , 0.        ],
            [0.        , 2.06817611, 0.        , 0.        , 0.        ,
             0.        , 0.        , 0.        , 0.        , 0.        ,
             0.        , 0.        , 0.        , 0.        , 0.        ,
             0.        , 0.        , 0.        , 0.        , 0.        ,
             0.        , 0.        , 0.        , 0.        , 0.        ],
            [0.        , 0.        , 1.49561973, 0.        , 0.        ,
             0.        , 0.        , 0.        , 0.        , 0.        ,
```

OUR SIGNMA VALUES ARE

8.3316198
2.0681761
1.4956197
1.1195300
0.9728115
0.6222320
0.5493220
0.4553298
0.3900748
0.2887603



IGNORED *MISSING VALUES*

RUNNING K MEANS

WITH SVD *1 EIGEN VALUE REMOVED*

WITHOUT SVD

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.788462
Machine Learning\n                    0.725000
Mathematics For Machine Learning\n    0.678571
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.788462
Machine Learning\n                    0.725000
Mathematics For Machine Learning\n    0.678571
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Natural Language Processing\n         1.0
Software Production Engineering\n     1.0
Machine Learning\n                    0.9
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Machine Learning\n                    1.0
Mathematics For Machine Learning\n    1.0
Visual Recognition\n                  0.8
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

The Web and the Mind      0.833333
Machine Learning\n        0.743750
Reinforcement Learning    0.650000
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.788462
Machine Learning\n                    0.725000
Mathematics For Machine Learning\n    0.678571
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.788462
Machine Learning\n                    0.725000
Mathematics For Machine Learning\n    0.678571
Name: 4, dtype: float64
```

OBSERVATIONS :

Overall accuracy has decreased compared to the previous 2 methods

Ratings from a smaller department is also considered for Recommendation

WITH SVD

1 EIGEN VALUE REMOVED

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.805118
Machine Learning\n                    0.737831
Mathematics For Machine Learning\n    0.678989
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.805118
Machine Learning\n                    0.737831
Mathematics For Machine Learning\n    0.678989
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Natural Language Processing\n         0.995144
Software Production Engineering\n     0.990886
Machine Learning\n                    0.892980
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Mathematics For Machine Learning\n    0.999943
Machine Learning\n                    0.998237
Visual Recognition\n                  0.798867
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

The Web and the Mind                  0.826440
Machine Learning\n                    0.753678
Software Production Engineering\n     0.655746
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.805118
Machine Learning\n                    0.737831
Mathematics For Machine Learning\n    0.678989
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.805118
Machine Learning\n                    0.737831
Mathematics For Machine Learning\n    0.678989
Name: 4, dtype: float64
```

IGNORED *MISSING VALUES*

RUNNING K MEANS

WITH SVD *2 EIGEN VALUE REMOVED*

WITHOUT SVD

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.788462
Machine Learning\n                    0.725000
Mathematics For Machine Learning\n    0.678571
Programming Languages\n               0.661538
Cloud Computing\n                     0.641667
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.788462
Machine Learning\n                    0.725000
Mathematics For Machine Learning\n    0.678571
Programming Languages\n               0.661538
Cloud Computing\n                     0.641667
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Natural Language Processing\n         1.00
Software Production Engineering\n     1.00
Machine Learning\n                    0.90
Mathematics For Machine Learning\n    0.80
Visual Recognition\n                  0.25
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Machine Learning\n                    1.00
Mathematics For Machine Learning\n    1.00
Visual Recognition\n                  0.80
Reinforcement Learning                0.75
Programming Languages\n               0.72
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

The Web and the Mind                0.833333
Machine Learning\n                  0.743750
Reinforcement Learning              0.650000
Techno-economics of networks        0.650000
Software Production Engineering\n   0.642857
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.788462
Machine Learning\n                    0.725000
Mathematics For Machine Learning\n    0.678571
Programming Languages\n               0.661538
Cloud Computing\n                     0.641667
```

OBSERVATIONS :

Overall accuracy has decreased compared to the previous 2 methods

Improvement from removing single eigen value

Ratings from a smaller department is also considered for Recommendation

WITH SVD

2 EIGEN VALUE REMOVED

```
test_data = [0,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.801636
Machine Learning\n                    0.736849
Mathematics For Machine Learning\n    0.676004
Cloud Computing\n                     0.652045
Programming Languages\n               0.650608
Name: 4, dtype: float64
```

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.801636
Machine Learning\n                    0.736849
Mathematics For Machine Learning\n    0.676004
Cloud Computing\n                     0.652045
Programming Languages\n               0.650608
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Natural Language Processing\n         1.004489
Software Production Engineering\n     0.980683
Machine Learning\n                    0.890102
Mathematics For Machine Learning\n    0.791024
Visual Recognition\n                  0.256728
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

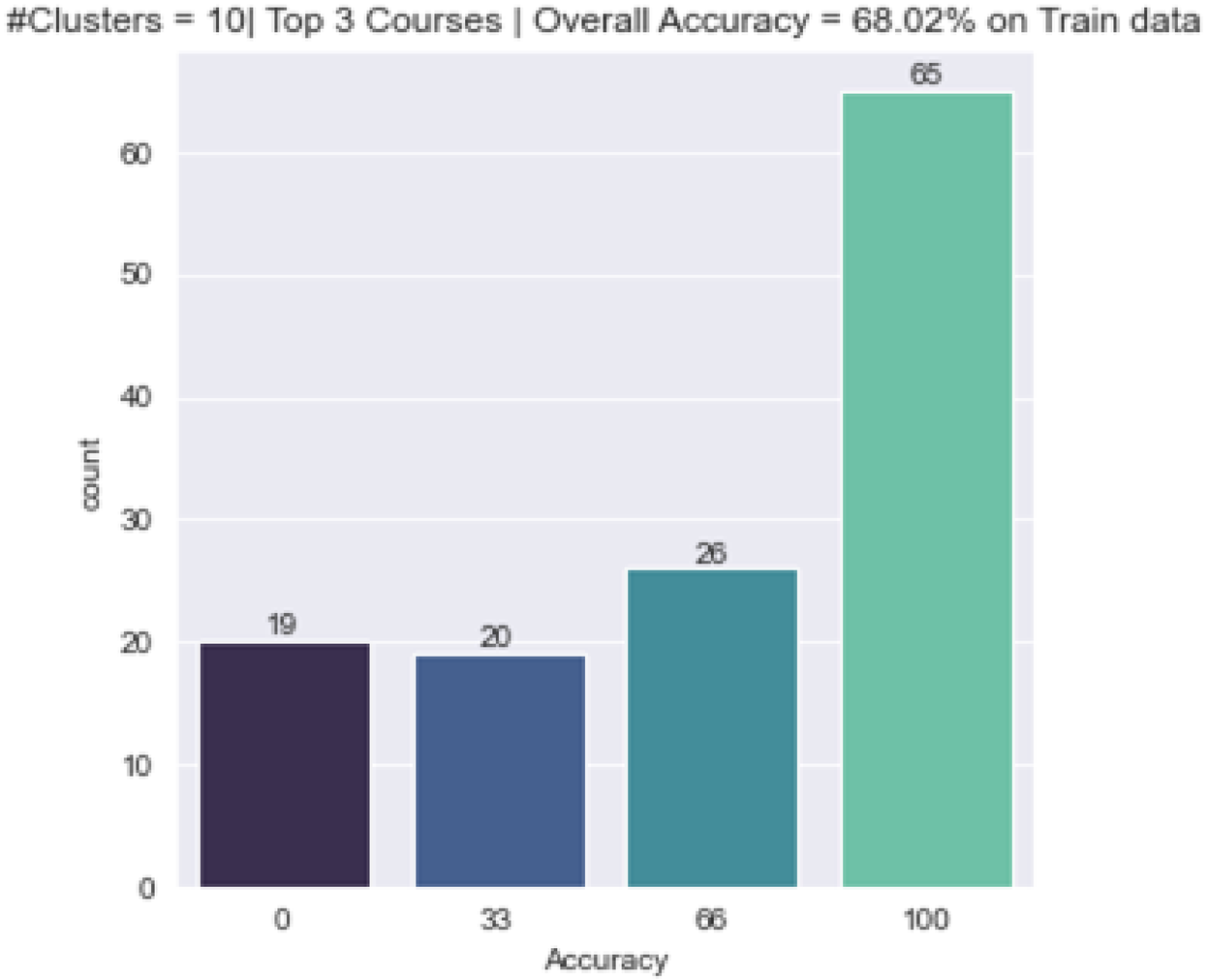
```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Mathematics For Machine Learning\n    1.016372
Machine Learning\n                    1.003642
Visual Recognition\n                  0.777755
Reinforcement Learning                0.750462
Programming Languages\n               0.743941
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

The Web and the Mind                0.810577
Machine Learning\n                  0.746005
Reinforcement Learning              0.644270
Privacy in the Digital Age          0.638473
Techno-economics of networks        0.633423
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.801636
Machine Learning\n                    0.736849
Mathematics For Machine Learning\n    0.676004
Cloud Computing\n                     0.652045
Programming Languages\n               0.650608
```



IGNORED *MISSING VALUES*

RUNNING K MEANS

WITH SVD *4 EIGEN VALUE REMOVED*

WITHOUT SVD

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.788462
Machine Learning\n                    0.725000
Mathematics For Machine Learning\n    0.678571
Programming Languages\n               0.661538
Cloud Computing\n                     0.641667
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Natural Language Processing\n         1.00
Software Production Engineering\n     1.00
Machine Learning\n                    0.90
Mathematics For Machine Learning\n    0.80
Visual Recognition\n                  0.25
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Machine Learning\n                    1.00
Mathematics For Machine Learning\n    1.00
Visual Recognition\n                  0.80
Reinforcement Learning                0.75
Programming Languages\n               0.72
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

The Web and the Mind                  0.833333
Machine Learning\n                    0.743750
Reinforcement Learning                0.650000
Techno-economics of networks          0.650000
Software Production Engineering\n     0.642857
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.788462
Machine Learning\n                    0.725000
Mathematics For Machine Learning\n    0.678571
Programming Languages\n               0.661538
Cloud Computing\n                     0.641667
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Software Production Engineering\n     0.788462
Machine Learning\n                    0.725000
Mathematics For Machine Learning\n    0.678571
Programming Languages\n               0.661538
```

OBSERVATIONS :

Overall accuracy of Test remains the same but Train has slightly reduced compared to removal of single and dual EVs.

Ratings from a smaller department is also considered for Recommendation

More Personalisation – SVD providing

Negating Input Values also showing good results

WITH SVD

4 EIGEN VALUE REMOVED

```
test_data = [1,0,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.849625
Mathematics For Machine Learning\n    0.745959
Software Production Engineering\n     0.745798
Reinforcement Learning                0.672602
Visual Recognition\n                  0.551963
Name: 4, dtype: float64
```

```
test_data = [0,1,0,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  9

Natural Language Processing\n         0.997894
Software Production Engineering\n     0.985826
Machine Learning\n                    0.880298
Mathematics For Machine Learning\n    0.787000
Visual Recognition\n                  0.256193
Name: 9, dtype: float64
```

```
test_data = [0,0,1,0,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  2

Mathematics For Machine Learning\n    0.961101
Machine Learning\n                    0.923920
Programming Languages\n               0.787475
Visual Recognition\n                  0.710643
Cloud Computing\n                     0.704532
Name: 2, dtype: float64
```

```
test_data = [0,0,0,1,0]
recommendTopX(test_data,topX)
```

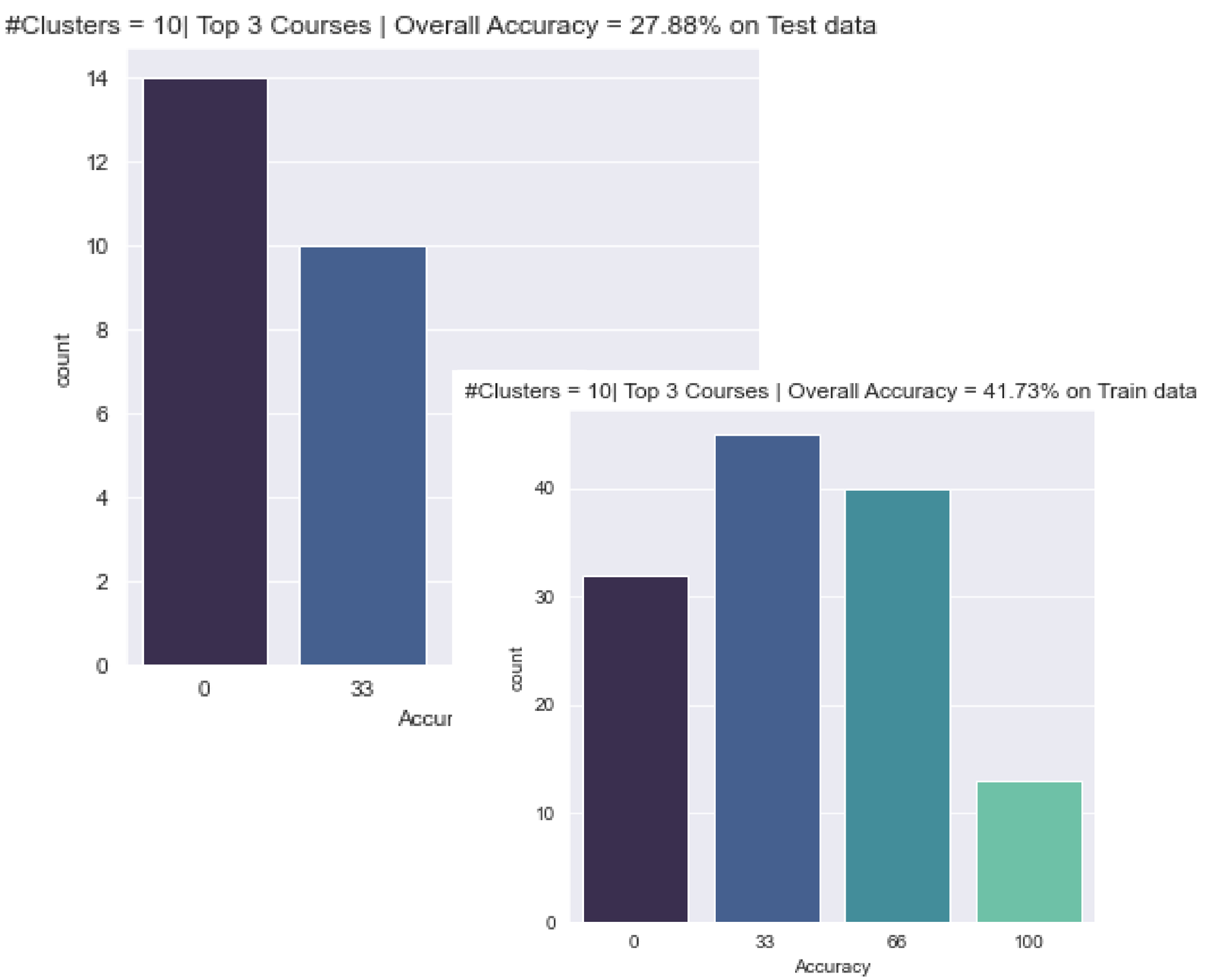
```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  3

The Web and the Mind                  0.869088
Machine Learning\n                    0.702703
Software Production Engineering\n     0.648645
Techno-economics of networks          0.621488
Privacy in the Digital Age            0.615487
Name: 3, dtype: float64
```

```
test_data = [0,0,0,-1,0]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.849625
Mathematics For Machine Learning\n    0.745959
Software Production Engineering\n     0.745798
Reinforcement Learning                0.672602
Visual Recognition\n                  0.551963
Name: 4, dtype: float64
```

```
test_data = [0,0,0,0,1]
recommendTopX(test_data,topX)
```

```
##--------------------LOG INFO : assignCentroid : Assigned Cluster :  4

Machine Learning\n                    0.849625
Mathematics For Machine Learning\n    0.745959
Software Production Engineering\n     0.745798
Reinforcement Learning                0.672602
```



APPLYING SVD BEFORE K MEANS

Sometimes there are wrong answers in ML as well :)

DECISION TREE

OBSERVATIONS :

Works like an extreme version of K Means clustering.

We have ideally given 32 cluster points to generate on (based on 5 subject and 2 choices per subject)

However, there are few subsets which are empty as well. For example rating all core courses less than .5 has no student and hence no recommendation can be provided by the decision tree algorithm (Total of 13 leaf nodes are empty out of the 32)

This is the downfall of decision tree. While it is much easier to visualise, the fact that we don't have a target label means we are left with a difficult way of calculating the purity of a leaf and whether we need to split it or not.

```
test = [1,1,1,1,1]

if test[0]<0.5 and test[1]<0.5 and test[2]<0.5 and test[3]<0.5 and test[4]<0.5:
    print(c1nc2nc3nc4nc5n.mean().sort_values())
elif test[0]<0.5 and test[1]<0.5 and test[2]<0.5 and test[3]<0.5 and test[4]>=0.5:
    print(c1nc2nc3nc4nc5y.mean().sort_values())
elif test[0]<0.5 and test[1]<0.5 and test[2]<0.5 and test[3]>=0.5 and test[4]<0.5:
    print(c1nc2nc3nc4yc5n.mean().sort_values())
elif test[0]<0.5 and test[1]<0.5 and test[2]<0.5 and test[3]>=0.5 and test[4]>=0.5:
    print(c1nc2nc3nc4yc5y.mean().sort_values())
elif test[0]<0.5 and test[1]<0.5 and test[2]>=0.5 and test[3]<0.5 and test[4]<0.5:
    print(c1nc2nc3yc4nc5n.mean().sort_values())
elif test[0]<0.5 and test[1]<0.5 and test[2]>=0.5 and test[3]<0.5 and test[4]>=0.5:
    print(c1nc2nc3yc4nc5y.mean().sort_values())
elif test[0]<0.5 and test[1]<0.5 and test[2]>=0.5 and test[3]>=0.5 and test[4]<0.5:
    print(c1nc2nc3yc4yc5n.mean().sort_values())
elif test[0]<0.5 and test[1]<0.5 and test[2]>=0.5 and test[3]>=0.5 and test[4]>=0.5:
    print(c1nc2nc3yc4yc5y.mean().sort_values())
elif test[0]<0.5 and test[1]>=0.5 and test[2]<0.5 and test[3]<0.5 and test[4]<0.5:
    print(c1nc2yc3nc4nc5n.mean().sort_values())
elif test[0]<0.5 and test[1]>=0.5 and test[2]<0.5 and test[3]<0.5 and test[4]>=0.5:
    print(c1nc2yc3nc4nc5y.mean().sort_values())
elif test[0]<0.5 and test[1]>=0.5 and test[2]<0.5 and test[3]>=0.5 and test[4]<0.5:
    print(c1nc2yc3nc4yc5n.mean().sort_values())
elif test[0]<0.5 and test[1]>=0.5 and test[2]<0.5 and test[3]>=0.5 and test[4]>=0.5:
    print(c1nc2yc3nc4yc5y.mean().sort_values())
elif test[0]<0.5 and test[1]>=0.5 and test[2]>=0.5 and test[3]<0.5 and test[4]<0.5:
    print(c1nc2yc3yc4nc5n.mean().sort_values())
elif test[0]<0.5 and test[1]>=0.5 and test[2]>=0.5 and test[3]<0.5 and test[4]>=0.5:
    print(c1nc2yc3yc4nc5y.mean().sort_values())
elif test[0]<0.5 and test[1]>=0.5 and test[2]>=0.5 and test[3]>=0.5 and test[4]<0.5:
    print(c1nc2yc3yc4yc5n.mean().sort_values())
```

```
c1nc2y = c1n[c1n['Computer Architecture'] >= 0.5]
c1nc2n = c1n[c1n['Computer Architecture'] < 0.5]
c1n = [] # freeing memory
```

```
c1yc2yc3y = c1yc2y[c1yc2y['Discrete Mathematics\n'] >= 0.5]
c1yc2yc3n = c1yc2y[c1yc2y['Discrete Mathematics\n'] < 0.5]
c1yc2y = [] # freeing memory
```

```
c1yc2nc3y = c1yc2n[c1yc2n['Discrete Mathematics\n'] >= 0.5]
c1yc2nc3n = c1yc2n[c1yc2n['Discrete Mathematics\n'] < 0.5]
c1yc2n = [] # freeing memory
```

```
c1nc2yc3y = c1nc2y[c1nc2y['Discrete Mathematics\n'] >= 0.5]
c1nc2yc3n = c1nc2y[c1nc2y['Discrete Mathematics\n'] < 0.5]
c1nc2y = [] # freeing memory
```

```
c1nc2yc3n = c1nc2y[c1nc2y['Discrete Mathematics\n'] < 0.5]
c1nc2y = [] # freeing memory
```

```
c1nc2nc3y = c1nc2n[c1nc2n['Discrete Mathematics\n'] >= 0.5]
c1nc2nc3n = c1nc2n[c1nc2n['Discrete Mathematics\n'] < 0.5]
c1nc2n = [] # freeing memory
```

```
c1yc2yc3yc4y = c1yc2yc3y[c1yc2yc3y['Economics'] >= 0.5]
c1yc2yc3yc4n = c1yc2yc3y[c1yc2yc3y['Economics'] < 0.5]
c1yc2yc3y = [] # freeing memory
```

```
c1yc2yc3nc4y = c1yc2yc3n[c1yc2yc3n['Economics'] >= 0.5]
c1yc2yc3nc4n = c1yc2yc3n[c1yc2yc3n['Economics'] < 0.5]
c1yc2yc3n = [] # freeing memory
```

```
c1yc2nc3yc4y = c1yc2nc3y[c1yc2nc3y['Economics'] >= 0.5]
c1yc2nc3yc4n = c1yc2nc3y[c1yc2nc3y['Economics'] < 0.5]
c1yc2nc3y = [] # freeing memory
```

KEY OBSERVATIONS

What are our takeaways from the dataset & the models we applied

If Dataset contains less amount of a group data, SVD with Non replacement of NaN can be applied – That provides more personalization

If Dataset has large amount of data, replacing NaN with zero provides better Accuracy

If Dataset has huge amount data of one class and small section of the other class, and missing values are filled with 0, the smaller class would be ignored

Given enough data that is representative of the entire world of data, decision tree will be able to generalize. Else we will have empty leaf nodes

Filling empty values by 0 is better than 0.5 as if a student hasn't rated a course, he/she is more likely not interested in it.

ML is prevalent is almost all test inputs as it is rated by most students across all types of students (CS, ECE & DT)

CS Courses dominate the ratings so it has lead to poor recommendation of ECE & DT electives. Further exacerbated by the poor ratings given by the students as well

Without SVD, replacing 0 I best since taking a educated & analyzed guess is better

With SVD, .5 is best since SVD takes the neutral value and is able to reduce the noise giving a better output

THANK YOU

Meet Nitin Mandhane MT2022061
Narasimhan N MT2022062
Jacob Mathew MT2022150