DATA MINING
ASSIGNMENT-3

AASTHA 2019224
SATWIK TIWARI 2019100

1

Preprocessing

1. There were no missing values in the dataset.
2. The categorical features were labeled using OneHotEncoding.
3. X and target column y were separated.
4. Principal Component Analysis was done for dimensionality reduction. The n_components were selected as 3 to better visualize the clusters in Q1. A cumulative variance of 97.8% was achieved with 3 principal components.

Clustering

Clustering was done with k=7, the number of classes in the dataset. The following models were used for clustering

1. Gaussian Based Clustering Model-Gaussian Mixture
2. K-Means
3. Birch
4. Mini Batch K-Means
5. Fuzzy C-Means

1. Centroid/representative object/prototype of each cluster

1. Gaussian Based Clustering Model

```
Means
```

```
 [[-14.33048854   0.32707579  -0.07316152]
 [ 12.72648744   0.10057065   0.05330428]
 [ -4.853531     0.11161925   1.50922941]
 [ -4.89035321  -1.10293942   0.23471094]
 [  1.71058991  -0.33009248   0.06795308]
 [ 20.80973755   0.6794516   -0.43413735]
 [ -7.82036662   0.91755591  -0.58787067]]

Covariances
 [[[ 1.06737761e+00 -2.49459955e-01 -1.32449487e-01]
  [-2.49459955e-01  1.69810563e+00  5.43643594e-01]
  [-1.32449487e-01  5.43643594e-01  1.06058787e+00]]

 [[ 3.09910153e+00  4.97610217e-01 -3.68684921e-01]
  [ 4.97610217e-01  1.10888940e+00 -3.41826175e-01]
  [-3.68684921e-01 -3.41826175e-01  1.37541045e+00]]

 [[ 1.23198617e-01 -2.43556996e-02  1.05088708e-03]
  [-2.43556996e-02  2.99353959e-01  2.52288624e-01]
  [ 1.05088708e-03  2.52288624e-01  3.23418712e-01]]

 [[ 1.71585194e-01 -2.46634685e-03  2.02766619e-02]
  [-2.46634685e-03  1.50033564e-01  1.14968838e-01]
  [ 2.02766619e-02  1.14968838e-01  1.80005716e-01]]

 [[ 2.21029216e+00 -5.80270252e-01  2.38559064e-01]
  [-5.80270252e-01  1.06315316e+00  9.58945418e-03]
  [ 2.38559064e-01  9.58945418e-03  8.66055196e-01]]

 [[ 2.32665617e+00 -1.66347792e-01  1.30552243e-01]
  [-1.66347792e-01  4.31403463e-01  1.52780436e-01]
  [ 1.30552243e-01  1.52780436e-01  4.31467986e-01]]

 [[ 6.28037531e-01 -6.04134910e-02  1.60190727e-02]
  [-6.04134910e-02  7.78187225e-01  7.12327469e-01]
  [ 1.60190727e-02  7.12327469e-01  8.41425643e-01]]]
```

Using the means and covariances the density was calculated. This was done by calculating the multivariate normal distribution and taking its log pdf. Using the density the point with maximum density was chosen as the center. The centers are represented as follows:

```
[[-1.46436462e+01,  5.07037585e-01,  1.05055270e-01],
       [ 1.34027414e+01,  2.24116943e-01, -1.94197900e-01],
```

```
       [-4.71319310e+00,  9.74704334e-02,  1.57581317e+00],
       [-4.67130192e+00, -1.23842545e+00,  1.92657373e-01],
       [ 1.34104970e+00, -2.53641898e-01,  1.97510349e-02],
       [ 2.04192990e+01,  1.07782557e+00, -4.13110364e-01],
       [-7.62647943e+00,  8.73596533e-01, -5.98146787e-01]]
```

2. K-Means

Cluster Centers

```
[[ -4.85704434,  -0.78323496,   0.52653102],
       [ 13.04557621,   0.13417694,   0.08398096],
       [-14.44753369,   0.34015623,  -0.08795273],
       [ -7.84855395,   0.90799137,  -0.57849004],
       [  5.72335799,  -0.75290146,   0.16244481],
       [ 20.79081282,   0.67814001,  -0.42823241],
       [  1.39076703,  -0.28967763,   0.02725263]]
```

3. Birch

Centroids of subclusters

```
[[-4.70533906e+00, -1.76155897e-02,  1.40994010e+00],
       [-4.68231818e+00, -7.65250965e-01,  7.11995207e-01],
       [-4.65453436e+00, -1.41048808e+00, -5.35422269e-02],
       [-5.64192698e+00, -1.37652970e+00, -1.63812716e-01],
       [-2.65064446e+00,  1.62537345e+00,  5.51629470e-03],
       [-5.67011115e+00, -6.55284273e-01,  5.71910593e-01],
       [ 3.19227575e-01,  2.77184325e+00,  1.30176099e+00],
       [-4.84708396e+00,  1.82740943e+00,  3.06300541e+00],
     ...
       [ 1.93696157e+01,  2.99653901e+00,  7.24462579e-01],
       [ 1.72759121e+01, -9.14952720e-01,  1.89970971e+00],
       [ 2.03138274e+01,  2.33819591e+00,  1.77344855e+00],
       [ 2.13974807e+01,  2.01969444e+00,  3.01287827e-01],
       [ 1.83781967e+01,  2.71893952e+00,  9.72680059e-01],
       [ 2.33906769e+01,  2.11719590e+00,  2.62809807e-01]]
```

4. Mini Batch K-Means

Cluster Centers

```
[[  5.82123639,  -0.82152819,   0.18063948],
       [ -4.85627731,  -0.7543809 ,   0.5478819 ],
       [ 20.66437034,   0.67813789,  -0.44754259],
       [-14.41002785,   0.43195749,  -0.07359517],
       [  1.38746098,  -0.29566984,   0.0380987 ],
       [ -7.82845518,   0.90318514,  -0.57438691],
       [ 13.10106239,   0.14729236,   0.03780597]]
```

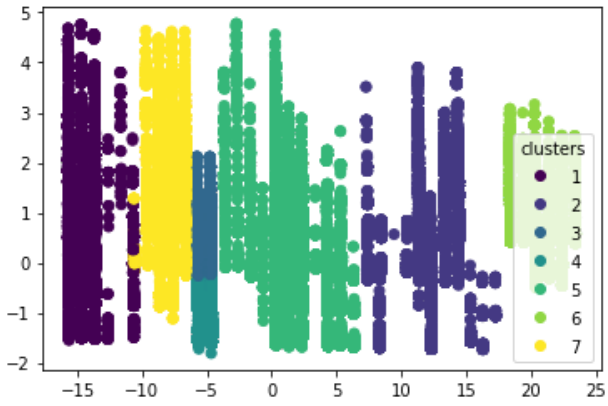
5. Fuzzy C-Means

```
[[  1.41475161,  -0.40427143,  -0.05189059],
       [ -7.84915597,   0.83890094,  -0.63917906],
       [  4.80725234,  -0.7651775 ,   0.27333742],
       [ -4.87022361,  -0.84239476,   0.44062268],
       [ 13.05855267,   0.09328717,   0.04635546],
       [-14.37181486,   0.24460986,  -0.14290269],
       [ 20.83005582,   0.62593769,  -0.41531885]]
```

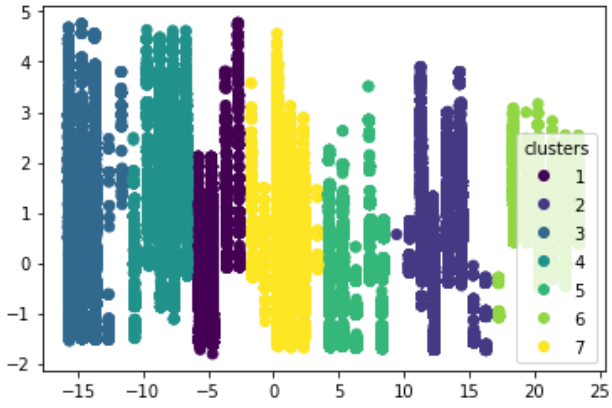
2. Visualization of the clusters

Visualizing clusters in 2D using PC1 and PC2

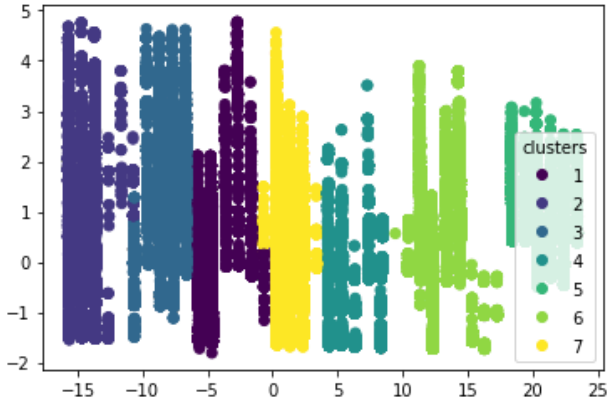
1. Gaussian Based Clustering Model



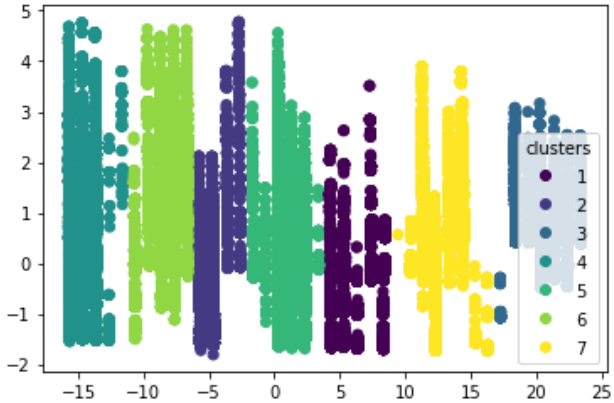
2. K-Means



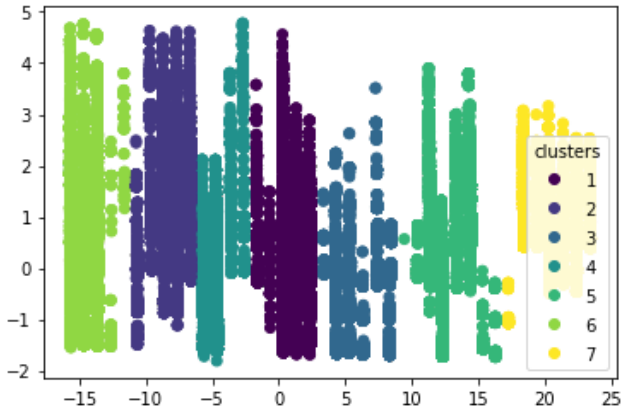
3. Birch



4. Mini Batch K-Means

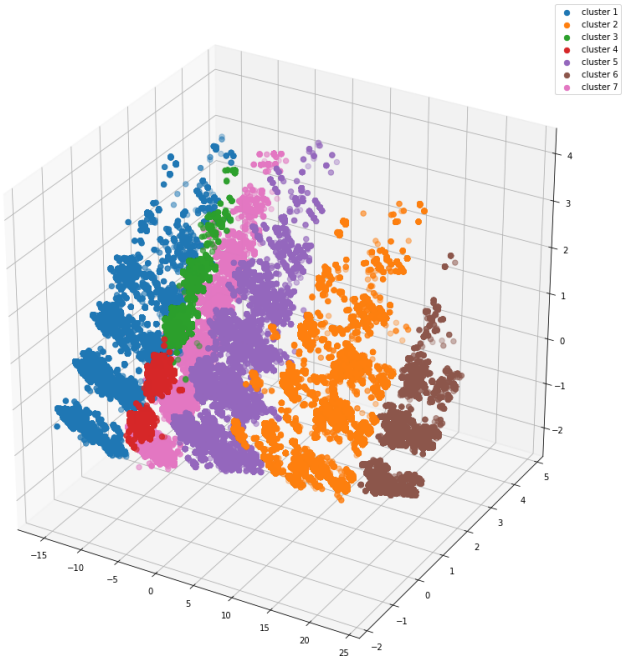


5. Fuzzy C-Means

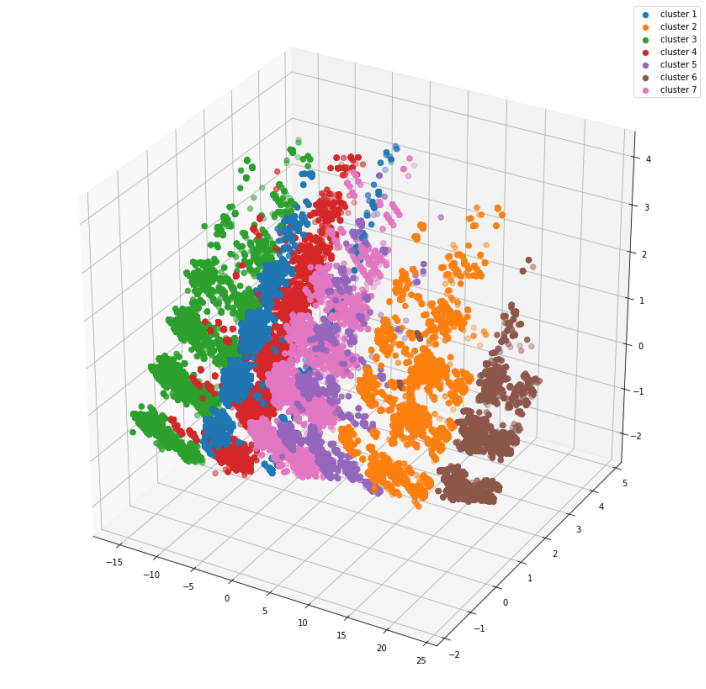


Visualizing clusters in 3D using all the 3 dimensions

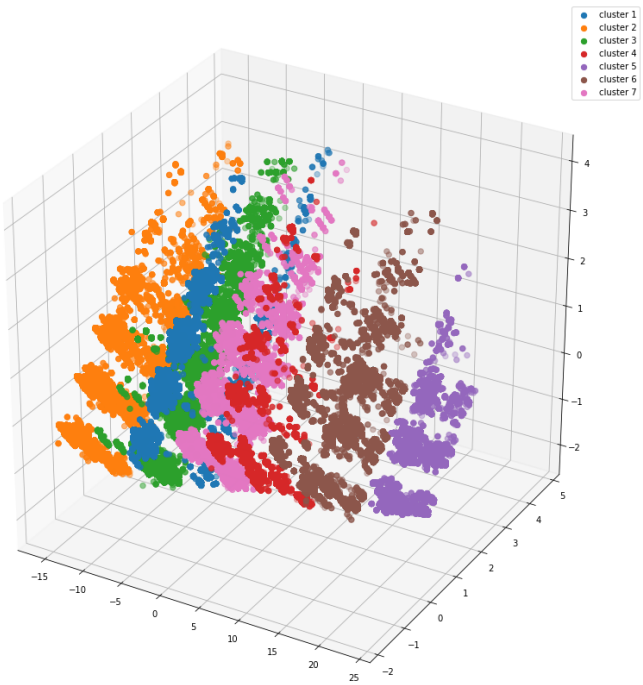
1. Gaussian Based Clustering Model



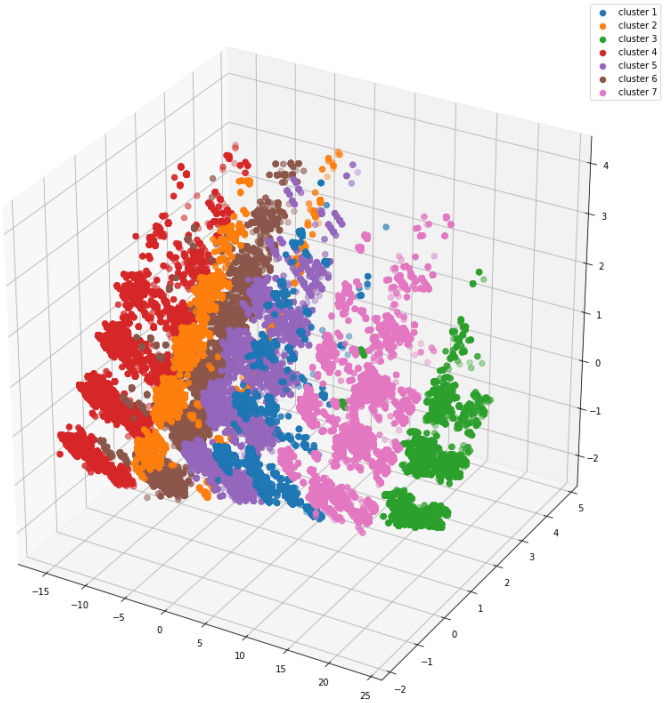
2. K-Means



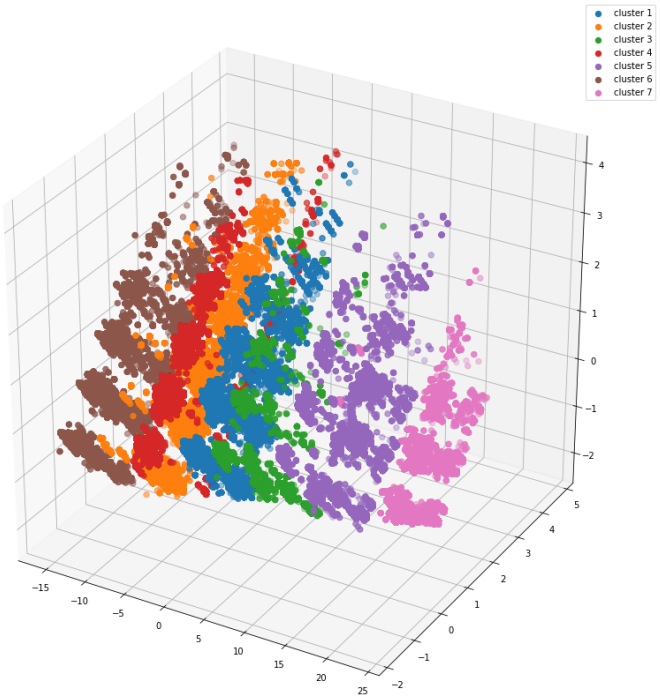
3. Birch



4. Mini Batch K-Means



5. Fuzzy C-Means



3. Comparing cluster distribution with true label count

Below are

1. The clustering performance metric scores
2. The number of data points in a cluster distribution and the true label count
3. The proportion of points in each cluster in true label count vs predicted cluster labels
4. The clusters that are overrepresented and underrepresented in our cluster distribution compared to true label counts.

1. Gaussian Based Clustering Model

```
V-measure score 0.214883344611975
Homogeneity score 0.26828591642345473
Rand score 0.6177580731034045

True Label Counts
 [148288 198310  25028   1923   6645  12157  14357]
Cluster Distribution
 [27815 70583 21525 80059 93730 25130 87866]


Proportion of points in each cluster:

For True Labels
 [0.36460556 0.48759798 0.06153801 0.00472821 0.0163385  0.02989122
 0.03530051]
For predicted
 [0.06839059 0.17354711 0.05292495 0.19684639 0.23046018 0.0617888
 0.21604198]


Cluster Representation compared to true labels:

Cluster 1 is underrepresented by -29.62149748714065 %
Cluster 2 is underrepresented by -31.4050866961063 %
Cluster 3 is underrepresented by -0.8613059000560592 %
Cluster 4 is overrepresented by 19.21181781523845 %
Cluster 5 is overrepresented by 21.41216794358606 %
Cluster 6 is overrepresented by 3.1897577623257964 %
Cluster 7 is overrepresented by 18.0741465621527 %
```

2. K-Means

```
V-measure score 0.2160347498535246
Homogeneity score 0.26682307273516237
Rand score 0.6208212467310872

True Label Counts
 [148288 198310  25028   1923   6645  12157  14357]
Cluster Distribution
 [103022  66133  26981  88681  15026  25198  81667]


Proportion of points in each cluster:

For True Labels
 [0.36460556 0.48759798 0.06153801 0.00472821 0.0163385  0.02989122
 0.03530051]
For predicted
 [0.25330704 0.1626056  0.06633998 0.21804587 0.03694543 0.061956
 0.20080008]


Cluster Representation compared to true labels:

Cluster 1 is underrepresented by -11.129852375660182 %
Cluster 2 is underrepresented by -32.499237782389336 %
Cluster 3 is overrepresented by 0.4801970947215205 %
Cluster 4 is overrepresented by 21.331766279492907 %
Cluster 5 is overrepresented by 2.060692191941147 %
Cluster 6 is overrepresented by 3.2064773744307953 %
Cluster 7 is overrepresented by 16.54995721746314 %
```

3. Birch

```
V-measure score 0.21838163229718102
Homogeneity score 0.26964003125463565
Rand score 0.6218413875175778

True Label Counts
 [148288 198310  25028   1923   6645  12157  14357]
```

```
Cluster Distribution
 [105004  27152  88510  15026  25130  66201  79685]


Proportion of points in each cluster:

For True Labels
 [0.36460556 0.48759798 0.06153801 0.00472821 0.0163385  0.02989122
 0.03530051]
For predicted
 [0.25818032 0.06676043 0.21762542 0.03694543 0.0617888  0.1627728
 0.19592681]


Cluster Representation compared to true labels:

Cluster 1 is underrepresented by -10.642524858129176 %
Cluster 2 is underrepresented by -42.08375542158011 %
Cluster 3 is overrepresented by 15.608741406611133 %
Cluster 4 is overrepresented by 3.221721726644177 %
Cluster 5 is overrepresented by 4.54502984942514 %
Cluster 6 is overrepresented by 13.288157597096689 %
Cluster 7 is overrepresented by 16.06262969993214 %
```

4. Mini Batch K Means

```
V-measure score 0.2160347498535246
Homogeneity score 0.26682307273516237
Rand score 0.6208212467310872

True Label Counts
 [148288 198310  25028   1923   6645  12157  14357]
Cluster Distribution
 [ 15026 103022  25198  26981  81667  88681  66133]


Proportion of points in each cluster:

For True Labels
 [0.36460556 0.48759798 0.06153801 0.00472821 0.0163385  0.02989122
 0.03530051]
For predicted
 [0.03694543 0.25330704 0.061956   0.06633998 0.20080008 0.21804587
 0.1626056 ]
```

```
Cluster Representation compared to true labels:

Cluster 1 is underrepresented by -32.76601394612351 %
Cluster 2 is underrepresented by -23.429094092075893 %
Cluster 3 is overrepresented by 0.041799030262498066 %
Cluster 4 is overrepresented by 6.161177060692192 %
Cluster 5 is overrepresented by 18.44615793148893 %
Cluster 6 is overrepresented by 18.81546465769053 %
Cluster 7 is overrepresented by 12.730509358065245 %
```

5. Fuzzy C-Means

```
V-measure score 0.1836340353612263
Homogeneity score 0.22573617342888677
Rand score 0.6082720976991959

True Label Counts
 [148288 198310  25028   1923   6645  12157  14357]
Cluster Distribution
 [ 66133 104222  38005 114462  15587  43101  25198]


Proportion of points in each cluster:

For True Labels
 [0.36460556 0.48759798 0.06153801 0.00472821 0.0163385  0.02989122
 0.03530051]
For predicted
 [0.1626056  0.25625756 0.09344542 0.28143533 0.03832479 0.10597529
 0.061956  ]


Cluster Representation compared to true labels:

Cluster 1 is underrepresented by -20.199996065973625 %
Cluster 2 is underrepresented by -23.134042113752372 %
Cluster 3 is overrepresented by 3.190741268920209 %
Cluster 4 is overrepresented by 27.67071215712501 %
Cluster 5 is overrepresented by 2.1986289918073902 %
Cluster 6 is overrepresented by 7.608407014369031 %
Cluster 7 is overrepresented by 2.6655487475043516 %
```

4 Comparing cluster formation of Gaussian Based Clustering With Other Clustering Methods

1. In Gaussian-based clustering, the representative objects are the means and covariances, and the cluster centers are calculated as points with maximum density using the means and covariances. The density is calculated by finding the multivariate normal distribution and taking its log pdf.

   In K-Means and Mini Batch K-Means, the representative object is the cluster centers found using the mean of all points belonging to the cluster. In Birch clustering, the representative object is the subcluster centers. In Birch, a tree is constructed with cluster centroids being at the leaf. Subcluster centers are the centroids of the subclusters read from the leaves. In Fuzzy c-means the representative object is the cluster center and each data point is assigned a probability to belong to a cluster.

2. The clusters were visualized in 2D(using principal components 1 and 2) and in 3D using all the 3 principal components. As we can see in the cluster visualizations, the label mapping is changed when comparing the cluster visualization of Gaussian-based clustering with the other three clustering methods.

   The shapes of the clusters were the same in the other 3 clustering methods and were different for the Gaussian-based clustering method. As we can see in the 2D visualization, on the x-axis from coordinates -5 to 5 we have a single green cluster whereas, in the other 3 clustering graphs, we have 2 clusters within those coordinates. This difference in cluster formation is because algorithms like K-means take only means into account while forming clusters whereas Gaussian Mixture considers means and covariances.

3. The actual cluster distribution and the true label counts were also different in Gaussian distribution and the other three clustering algorithms(K-Means, Birch, Mini Batch K-Means) while Fuzzy C-Means had a slightly different cluster distribution. As we can observe in the table below the lengths and proportions are the same for the other three clustering algorithms (with only the label mappings being different) and different for Gaussian and slightly different for Fuzzy C-Means.

Cluster Lengths

```
True Label Counts
 [148288 198310  25028   1923   6645  12157  14357]

Cluster Distribution in Gaussian
 [27815 70583 21525 80059 93730 25130 87866]

Cluster Distribution in K-Means
 [103022  66133  26981  88681  15026  25198  81667]

Cluster Distribution in Birch
 [105004  27152  88510  15026  25130  66201  79685]

Cluster Distribution in Mini Batch K-Means
 [ 15026 103022  25198  26981  81667  88681  66133]

Cluster Distribution in Fuzzy C-Means
 [ 66133 104222  38005 114462  15587  43101  25198]
```

Proportion of data in each cluster

```
For True Labels
 [0.36460556 0.48759798 0.06153801 0.00472821 0.0163385
0.02989122 0.03530051]

For Gaussian
 [0.06839059 0.17354711 0.05292495 0.19684639 0.23046018 0.0617888
 0.21604198]

For K-Means
 [0.25330704 0.1626056  0.06633998 0.21804587 0.03694543 0.061956
 0.20080008]

For Birch
 [0.25818032 0.06676043 0.21762542 0.03694543 0.0617888  0.1627728
 0.19592681]

For Mini Batch K-Means
 [0.03694543 0.25330704 0.061956   0.06633998 0.20080008
0.21804587 0.1626056 ]

For Fuzzy C-Means
 [0.1626056  0.25625756 0.09344542 0.28143533 0.03832479
0.10597529 0.061956  ]
```

4. Results for performance metrics

In Gaussian, different values are obtained for V-measure score, Homogeneity score, and rand score. The V-measure score and rand score values were slightly lower for Gaussian as compared to the other 3 and higher than fuzzy c-means. Since the rand score is less in Gaussian it shows lesser similarity between clusters from true labels, compared to the other 3 clustering algorithms and higher compared to fuzzy c-means.

For Gaussian

```
V-measure score 0.214883344611975
Homogeneity score 0.26828591642345473
Rand score 0.6177580731034045
```

For K-Means

```
V-measure score 0.2160347498535246
Homogeneity score 0.26682307273516237
Rand score 0.6208212467310872
```

For Birch

```
V-measure score 0.21838163229718102
Homogeneity score 0.26964003125463565
Rand score 0.6218413875175778
```

For Mini Batch K-Means

```
V-measure score 0.2160347498535246
Homogeneity score 0.26682307273516237
Rand score 0.6208212467310872
```

For Fuzzy C-Means

```
V-measure score 0.1836340353612263
Homogeneity score 0.22573617342888677
Rand score 0.6082720976991959
```

2

The following preprocessing and optimization has been done on the dataset.

1. One-Hot Encoding -

Converting categorical data to the numerical data in the given dataset.

Encoding Categorical Values

```
[4] df = pd.get_dummies(df)
```

```
[5] df.head()
```

	Horizontal_Distance_To_Hydrology	Vertical_Distance_To_Hydrology	Soil_Type	Wilderness	target	Elevation_elevation_high	Elevation_elevation_low	Elevation_elevation_medium
0	0	1	22	0	2	0	0	1
1	1	1	32	2	1	1	0	0
2	1	1	10	2	2	0	0	1
3	2	1	23	2	1	1	0	0
4	2	1	28	0	2	1	0	0

2. Feature Scaling -

In clustering, feature scaling is very important to give equal priority to each feature as we're concerned with the distance between two points. Thus we have standardScaler() to scale all the features in our dataset.

Feature Scaling

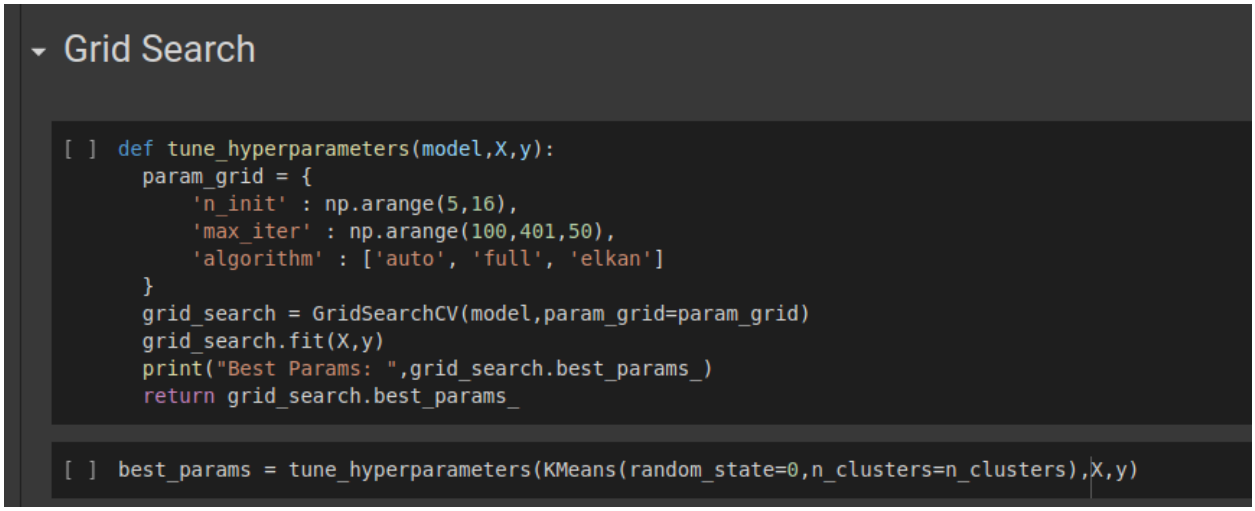
```
[ ] standardScaler = StandardScaler()

    X_train = standardScaler.fit_transform(X_train)
    X_test = standardScaler.transform(X_test)
    X = standardScaler.fit_transform(X)
```

3. Hyperparameter Tuning -

To get the best hyperparameters in k-means we have used Grid Search to tune our model. The best Parameters we got were -

algorithm	Auto
max_Iter	100
n_init	8



4. Mapping Clusters -

So finally after training our model, we have the labels of the clusters from 0 to 6 and each point lies in one of these clusters. So it was really important to find the correct mapping to the target values (from 1 to 7) to increase the accuracy of the model.

So we have used the number of True Positive count for each of the clusters and for each of the target mapping, Based on that we have prioritized the target mapping to each cluster and then assigned the mapped numbers so that the most prioritized target number is mapped to each of the clusters.

Mapping clusters

```
def find_mapping(labels, X, y):
  # labels - contain values from 0 to 6
  # y contain values from 1 to 7

  pref = {}
  for clus in range(7):
    temp = {}
    for j in range(1,8):
      cnt = 0
      for k in range(len(y)):
        cnt += (y[k] == j and labels[k] == clus)
      temp[j] = cnt
    temp = dict(sorted(temp.items(), key=lambda item: -item[1]))
    order = []
    for i in temp: order.append(i)

    pref[clus] = order

  perm = []
  vis = [0] * (10)

  for i in range(7):
    for j in pref[i]:
      if(vis[j]): continue
      else:
        perm.append(j);
        vis[j] = 1
        break

  return perm;
```

5. Predictions and Results -

So after mapping the clusters to the target numbers, We tried to make some predictions on the given dataset. Below are the results -

Dataset	F1 - Score
Full Dataset	0.4207065081867018
Training Dataset	0.42083496813276416
Testing Dataset	0.4203184202811393

Learnings -

1. Learned about different clustering techniques and other optimizations.
2. Compared Gaussian with different algorithms such as kmeans, Birch and fuzzy C-means and learned about the pros and cons of each of them.
3. Calculated center/representative object for each of the clusters and compared their distribution with true label count.
4. Visualization of clusters obtained from different clustering techniques.
5. Preprocessed the data and learned how to tune the hyperparameters of a machine learning model using Grid Search. Also, explored different cluster mapping techniques.

References -

1. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
2. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html
3. https://pypi.org/project/fuzzy-c-means/
4. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
5. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html
6. https://medium.com/analytics-vidhya/why-is-scaling-required-in-knn-and-k-means-8129e4d88ed7
7. https://stackoverflow.com/questions/34611038/grid-search-for-hyperparameter-evaluation-of-clustering-in-scikit-learn