Student Declaration of Authorship

HERIOT WATT UNIVERSITY
UK | DUBAI | MALAYSIA

Course code and name:	F21DL: Data Mining and Machine Learning
Type of assessment:	**Group**
Coursework Title:	Part 2. Clustering
Student Name:	Arshati Ajay Marchande, Asmitha Krishnakumar, Gauri Revankar, Pooja Sheni Meledath, Prasitha Naidu
Student ID Number:	H00382093, H00376043, H00373987, H00386700, H00379641

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Arshati, Asmitha, Gauri, Pooja, Prasitha*

Date: *05/11/2023*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

PART 2: CLUSTERING

TABLE OF CONTENTS

1. GITHUB LINK

GitHub Link: https://github.com/poojameledath/F20DL_Group_5_DMML_Portfolio

2. CONTRIBUTIONS

Arshati

- Algorithms to compute optimal number of clusters
- Comparing clustering algorithms with Bayesian classification
- Report

Asmitha

- Exploring Various Clustering Algorithms
- Pros and Cons of different clustering algorithm
- Report

Gauri

- K-Means Clustering and Accuracy Evaluation
- Additional Preprocessing and Visualization
- Report

Pooja

- Exploring Various Clustering Algorithms
- Pros and Cons of different clustering algorithms
- Report

Prasitha

- Algorithms to compute optimal number of clusters
- Comparing clustering algorithms with Bayesian classification
- Report

3. K MEANS CLUSTERING

3.1 Additional Preprocessing

Apart from the preprocessing methods applied in part 1, Principal Component Analysis (PCA) was applied to reduce the dimensions of the dataset. Expected variance score was used to get an estimate of the number of dimensions to reduce the datasets. PCA was initialized with a value of 5 components.
Dimension reduction was performed on 2 datasets: X train all dataset (complete dataset containing all samples) and the dataset with top 5 features from each class.

3.2 K Means Clustering

The K means clustering model was fit on the X_train_all dataset which contained all class samples and dataset with top 5 features from each class. Following are the hyperparameter values that were set for the models:
n_clusters=10, init='k-means++', n_init=50.

The scores on evaluation of model on the complete train dataset were:

```
1 evaluatingModel(clusters_all, pd.DataFrame(X1_pca), kmeans, X1_train_array)
```

```
Silhouette Score: 0.22916449284958584
Davies-Bouldin Index: 1.3144167886700848
Calinski-Harabasz Index: 2783.4708928384357
```

The scores on evaluation of model on the top 5 features dataset were:

```
2 evaluatingModel(clusters_5, pd.DataFrame(X_pca), kmeans_5, X5_train_array)
```

```
Silhouette Score: 0.26591986211494945
Davies-Bouldin Index: 1.244254862559205
Calinski-Harabasz Index: 3628.972812144454
```

The complete train dataset achieved an accuracy of 35.72% whereas the top 5 features dataset achieved an accuracy of 53.08%.

Visualization of Clusters:
Below are the methods using which the clusters formed by K means clustering were visualized:

1. PCA (in 1D, 2D and 3D space)
2. Scatter Plots (with and without cluster centres)
3. Histogram
4. Violin plot – it is a hybrid of Box plot and Kernel Density plot.

3.3 Conclusions

Conclusions: (Complete train dataset)

- Cluster 1 contained maximum number of samples (interpreted from the histogram)
- Cluster 9 contained the least number of samples (interpreted from the histogram)

Conclusions: (Top 5 features dataset)

- Cluster 5 contained maximum number of samples (interpreted from the histogram)
- Clusters 0 and 2 contained the least number of samples (interpreted from the histogram)

Cluster centres in Top 5 dataset are more spread out compared to the ones in complete train dataset (observed from the scatter plots).
The number of outliers greatly reduced after performing PCA. This can be observed from the violin plots (The ends of the plots aren't stretched).

4. EXPLORING VARIOUS CLUSTERING ALGORITHMS

We have tried the below clustering algorithm:

Hard Clustering

- *DBSCAN*
- *BIRCH Algorithm (Hierarchical Algorithm)*
- *Agglomerative Clustering Algorithm (Hierarchical Algorithm)*

Soft Clustering

- Gaussian Mixture Model (GMM)
- Fuzzy C Means Clustering
- Self-Organized Map Model (SOM)

COMPARISON OF THE SILHOUETTE SCORE OF DIFFERENT HARD CLUSTERING ALGORITHMS:

	DBScan	BIRCH	Agglomerative
WITHOUT PCA	-0.074	0.226	0.161
WITH PCA	0.368	0.408	0.249

COMPARISON OF THE SILHOUETTE SCORE OF DIFFERENT SOFT CLUSTERING ALGORITHMS:

	GMM	FUZZY C MEANS	SOM
WITHOUT PCA	0.463	0.482	0.017
WITH PCA	0.551	0.552	0.055

4.1 Comparison of Different Clustering Algorithm

We evaluated the different hard and soft clustering algorithms on dataset types: one without PCA reduction and another with PCA reduction. For the hard clustering without PCA, DBScan displayed a low silhouette score of -0.074, implying that DBScan had difficulty forming well-defined clusters. At the same time, BIRCH and Agglomerative showed relatively better clustering performance with silhouette scores of 0.226 and 0.161, respectively. The variation in the result could be because DBScan heavily relies on the density of the data point. However, with PCA applied, there is a significant change in all the algorithms, with BIRCH changing to 0.408, surpassing DBScan and Agglomerative with 0.368 and 0.249 each.

Similarly, for the soft clustering algorithm without PCA, Fuzzy C Means had the highest silhouette score of 0.482, followed by GMM at 0.463. In contrast, SOM had a lower score of 0.017 suggesting weak clustering. With PCA, the performance improved for all three

algorithms. While Fuzzy C Means still leads with the highest silhouette score of 0.552, GMM's score increased to 0.551 and SOM to 0.055. We can infer that there is an improvement after performing PCA, that the effectiveness of the overall clustering algorithm increased.

4.2 Pros and Cons of the Algorithms

For hard clustering, DBScan can automatically detect the number of clusters, which is convenient when we do not know the number of clusters beforehand. DBScan may struggle with high-dimensional datasets as the data becomes more sparsely distributed, which can lead to the fragmentation of clusters. Similarily, another hard clustering algorithm, BIRCH, is efficient for managing large datasets as it follows an incremental and iterative approach. One drawback of BIRCH is that it can produce unbalanced clusters when dealing with some datasets. Similarly, for the clustering method Agglomerative Clustering, the algorithm determines how many clusters you want as it runs, making it suitable for unsupervised data. However, it is slow and computationally intensive for large datasets.

For soft clustering, SOM Clusters offer several advantages, such as their effectiveness in visualizing high-dimensional data and their ability to reveal intricate data patterns. They are well-suited for unsupervised learning tasks. However, using SOM clusters has its challenges. It often requires tuning of hyperparameters, and the results can be sensitive to data scaling and initialization. Additionally, SOMs can be computationally intensive, especially when dealing with large datasets, as it is initialized using random weights; thus, the performance varies with each computation. Another soft clustering method, Fuzzy C-Means, is known for dealing with uncertainty and ambiguity by assigning degrees of membership to clusters. It offers a flexible and computationally efficient clustering approach that is well-suited for large datasets. But FCM can be sensitive to the placement of clusters, and it relies on parameter tuning, such as the fuzziness parameter. Lastly, GMM offers flexibility in capturing different clusters, making it suitable for complex datasets. It provides probability information for cluster assignments and avoids assumptions about cluster shape and size, which benefits non-uniform data distributions. One of the cons of GMM is that it is computationally intensive.

5. ALGORITHMS TO COMPUTE OPTIMAL NUMBER OF CLUSTERS

5.1 Silhouette Score:

The silhouette score is a measure of how similar an object is to its own cluster compared to other clusters.
For each data point, the silhouette score ranges from -1 to 1, where a higher score indicates that the object is well-matched to its own cluster and poorly-matched to neighboring clusters.

5.2 Calinski-Harabasz Index:

A higher Calinski-Harabasz index suggests that the data points are well separated into distinct clusters.

5.3 Davies-Bouldin Index:

The Davies-Bouldin index measures the average similarity between each cluster and its most similar cluster.
A lower Davies-Bouldin index indicates that clusters are well separated and have a distinct boundary

Below is the output for all the three scores after running it for the normalized dataset:

```
Silhouette Score: 0.22974457053309918
Davies-Bouldin Index: 1.3135304824863572
Calinski-Harabasz Index: 2783.4353578941327
```

Below is the output for all the three scores after running it for the top 5 dataset:

```
Silhouette Score: 0.2660658083346455
Davies-Bouldin Index: 1.2465743993576968
Calinski-Harabasz Index: 3628.8783362268277
```

From the above outputs we conclude that top 5 has the highest silhouette score. When checking for the Davies-Bouldin Index, the lowest value is for top 5 dataset. When we check for the Calinski-Harabasz Index, the highest score is for top 5 dataset.

6. COMPARING CLUSTERING ALGORITHMS WITH BAYESIAN CLASSIFICATION

6.1 Accuracy Calculation With K-Means Clustering

After loading the x train and y train dataset we specified the number of clusters and number of iterations for k means algorithm.

The K-means clustering algorithm is initiated with these parameters and the dataset is fitted to it using the kmeans.fit().

The cluster labels assigned to each data point are stored in the cluster_labels_all variable. For each cluster, the code calculates the most common class label from the y_train dataset among the data points within that cluster.

```
Cluster 0: Most Common Class - [2]
Cluster 1: Most Common Class - [1]
Cluster 2: Most Common Class - [3]
Cluster 3: Most Common Class - [2]
Cluster 4: Most Common Class - [1]
Cluster 5: Most Common Class - [2]
Cluster 6: Most Common Class - [0]
Cluster 7: Most Common Class - [4]
Cluster 8: Most Common Class - [1]
Cluster 9: Most Common Class - [3]
```

It also computes the class distribution within each cluster, displaying the count and percentage of each class within the cluster.

The cluster labels are assigned to each data point in the original dataset, and a new column 'Cluster' is added to represent the cluster number, and another column 'Assigned_Class' is added to represent the most common class label in that cluster.

The code calculates the accuracy of the clustering results by comparing the 'Assigned_Class' to the original class labels in the y_train dataset. It counts how many data points were correctly assigned to their respective clusters and calculates the accuracy.

Accuracy score for complete dataset : **35.72%**
Accuracy score for Top_5 dataset : **53.08%**

6.2 Accuracy Using Bayesian Classification In Part-1

Gaussian Naive Bayes gives an accuracy of **43.59%** for train data when trained on the complete dataset
Multinominal Naive Bayes gives an accuracy of **60.80%** for train data when trained on the **Top_5** dataset

6.3 Conclusion After Comparison

The accuracy obtained from the **complete** dataset after doing **k-means clustering** is **35.72%.** After comparing the accuracy with the Bayesian classification models accuracy, we can conclude that **gaussian naive bayes** has the higher accuracy with **43.59%** on the **complete dataset**.

The accuracy obtained from the **Top_5** dataset after doing **k-means clustering** is **53.08%.** After comparing the accuracy with the Bayesian classification models accuracy, we can conclude that **Multinominal naive bayes** has the higher accuracy with **60.80%** on the **Top_5** dataset.