

AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Data Warehousing and Data Mining [Section B]

Date: 23.12.2023

|  |  |
| --- | --- |
| **Topic: Implementing k-means clustering on the dataset Cervical Cancer Behavior Risk** | |
| **Name** | **ID** |
| ABDULLAH AL NASIR | 19-41052-2 |
| TONOY CHANDRA SARKAR | 20-43804-2 |
| NAZMUS SAKIB | 20-44113-2 |

## Submitted To

Dr. Akinul Islam Jony

Associate Professor & Head (UG) CS

Faculty of Science and Technology

# **Introduction:**

The goal of this project is to create a consumer segmentation model by utilizing data mining techniques. A key component of contemporary corporate strategy is customer segmentation, which divides consumers into discrete groups according to their traits, inclinations, and actions. Businesses can improve overall happiness and loyalty by customizing their marketing strategies, product offers, and customer experiences to cater to the specific needs of various client segments. In this study, we used the well-liked unsupervised machine learning technique K- Means Clustering to explore the field of customer segmentation.

# **K-Means clustering Algorithm:**

There many methods of clustering but for this project we are going to implement K-means clustering algorithm. K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which mean reassign each data point to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH. Step-7: The model is ready.

## Advantages:

* K-Means is computationally efficient and easy to implement, making it suitable for large datasets and real-time applications.
* It can be applied to a variety of data types, including numerical and categorical, making it versatile for different types of datasets.
* K-Means scales well with the size of the dataset and is efficient in high-dimensional spaces.
* The results of K-Means clustering are easy to interpret, and the algorithm is intuitive in identifying natural groupings in the data.
* Given the same initial conditions, K-Means tends to converge to a similar solution, providing consistency in results.

## Disadvantages:

* K-Means is sensitive to the initial selection of centroids, and different initializations may lead to different final clusters.
* K-Means assumes that clusters are spherical and equally sized, which may not be suitable for datasets with non-uniformly shaped or elongated clusters.
* The choice of the number of clusters, 'k,' is crucial but not always clear-cut. It often requires domain knowledge or the use of additional techniques for optimal selection.
* Outliers can significantly affect the clustering results in K-Means. The algorithm is sensitive to noise and may assign outliers to incorrect clusters.
* K-Means assumes that clusters have equal variance, which might not hold in real-world scenarios where clusters exhibit different variances.

# **Dataset Description:**

Cancer is a condition when the body's cells grow out of control. Even if cancer later spreads to other bodily areas, it is always given the name of the portion of the body where it first manifests. Cervical cancer is the name given to cancer that first appears there. The cells of the cervix are where cervical cancer begins. The cervix is the uterus' (the womb's) lower, thin end. The uterus and vagina, or birth canal, are joined by the cervix. In most cases, cervical cancer grows gradually over time. Before cancer develops in the cervix, the cervical tissue undergoes a process called dysplasia, during which aberrant cells start to emerge in the tissue. Over time, if not destroyed or removed, the abnormal cells may become cancer cells and start to grow and spread more deeply into the cervix and to surrounding areas. Anyone with a cervix is at risk for cervical cancer. It occurs most often in people over age 30. Long-lasting infection with certain types of human papillomavirus (HPV) is the main cause of cervical cancer. HPV is a common virus that is passed from one person to another during sex. At least half of sexually active people will have HPV at some point in their lives, but few women will get cervical cancer. Screening tests and the HPV vaccine can help prevent cervical cancer. When cervical cancer is found early, it is highly treatable and associated with long survival and good quality of life.

<https://www.kaggle.com/datasets/senapatirajesh/cervical-cancer>

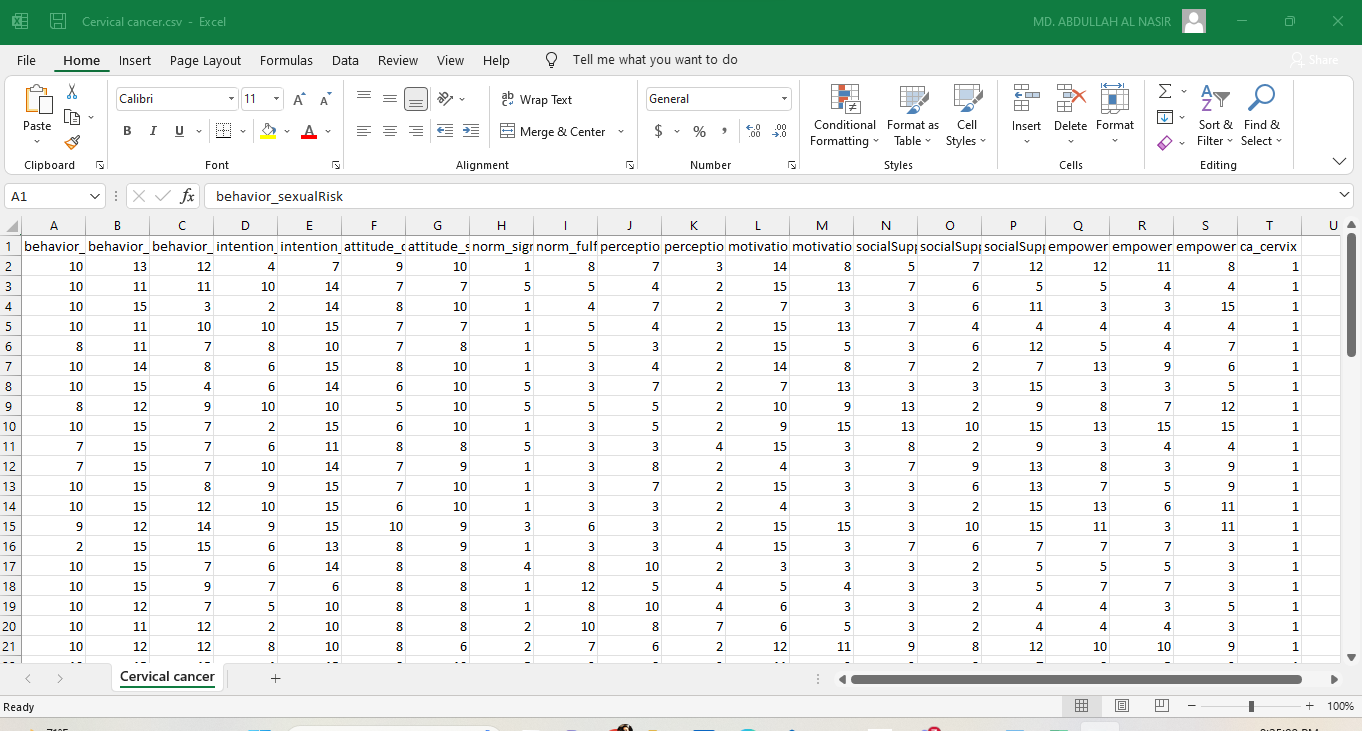
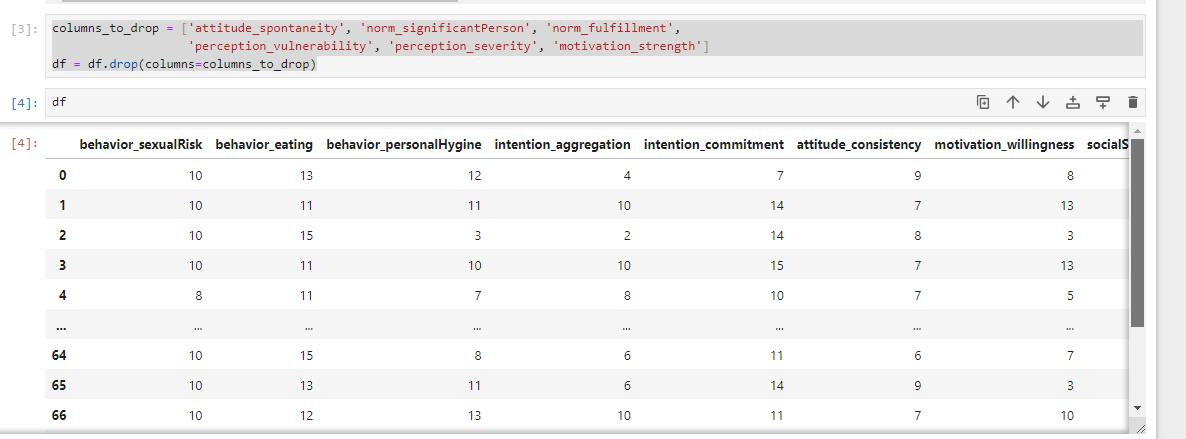


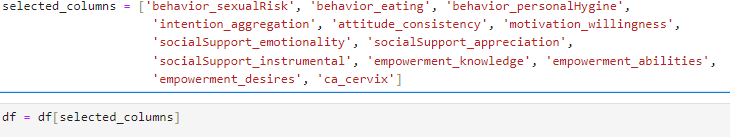
Fig: CSV file of the dataset

1. **Implementation:**
   1. **-** Data Preproccesing

This Python code drops specific columns from a Pandas DataFrame named df. The columns to be removed are specified in the list columns\_to\_drop, including 'attitude\_spontaneity', 'norm\_significantPerson', 'norm\_fulfillment', 'perception\_vulnerability', 'perception\_severity', and 'motivation\_strength'. The drop method is then applied to the DataFrame, removing the specified columns. The resulting DataFrame is stored back in the variable df without the dropped columns, effectively eliminating those factors from the analysis.



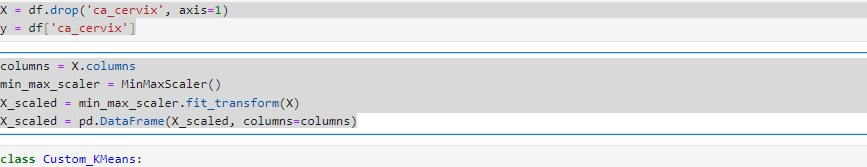
This Python code creates a list named selected\_columns containing specific columns from a DataFrame. These columns pertain to various behavioral, attitudinal, and support factors related to cervical cancer. The list is likely intended for selecting and analyzing a subset of relevant features in the dataset.



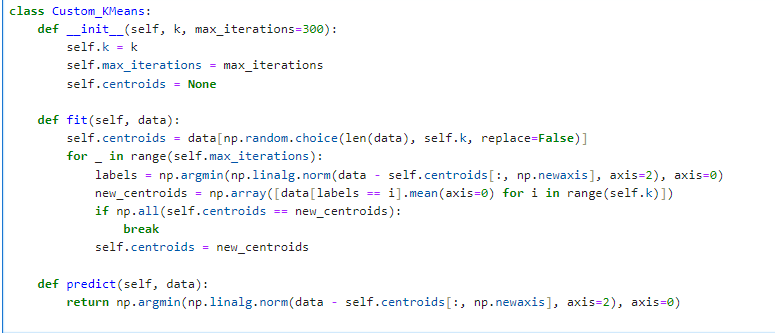
This Python code uses scikit-learn's LabelEncoder to convert the categorical values in the 'ca\_cervix' column of a Pandas DataFrame (df) into numerical labels. The fit\_transform method fits the encoder to the unique values in the column and transforms those values, replacing the original categorical data with corresponding numerical labels.



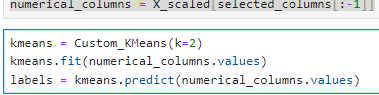
This Python code prepares the dataset for machine learning. It separates the target variable ('ca\_cervix') from the features (X), scales the feature values using Min-Max scaling, and replaces the original feature values with their scaled counterparts in a new DataFrame called **X\_scaled**.



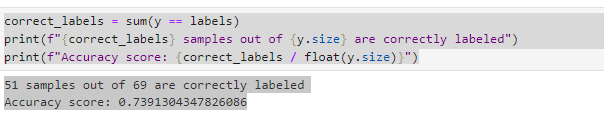
This Python code defines a simple implementation of the K-Means clustering algorithm in a class called Custom\_KMeans. The class is initialized with the desired number of clusters (k) and a maximum number of iterations. The fit method fits the model to the input data by iteratively updating cluster centroids based on the mean of assigned data points. The predict method assigns data points to the nearest cluster. The algorithm stops iterating when the centroids no longer change. This custom implementation provides a basic understanding of how the K-Means algorithm works, using NumPy for efficient array operations



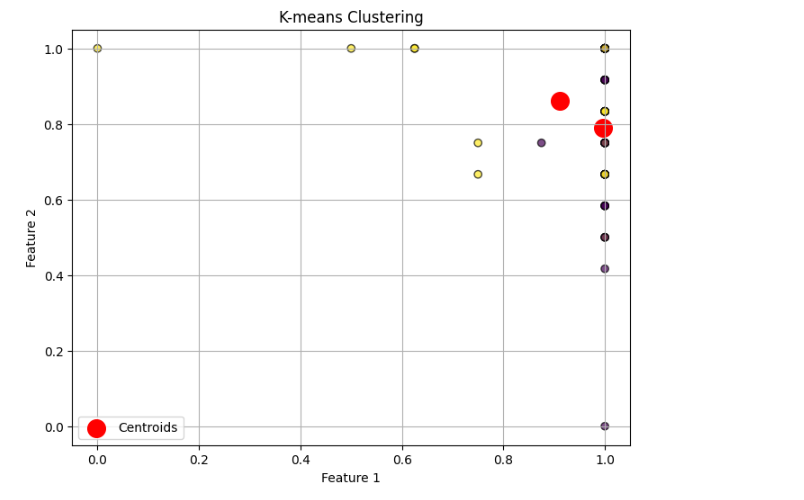
In the first line (1), a subset of the scaled features (numerical\_columns) is selected, excluding the last column. Then, an instance of the custom K-Means class (Custom\_KMeans) with two clusters is created (2). The fit method is applied to the scaled numerical features, determining cluster assignments based on centroid updates. Finally, the predict method is used to obtain cluster labels for each data point in the scaled numerical features, and these labels are stored in the 'labels' variable.



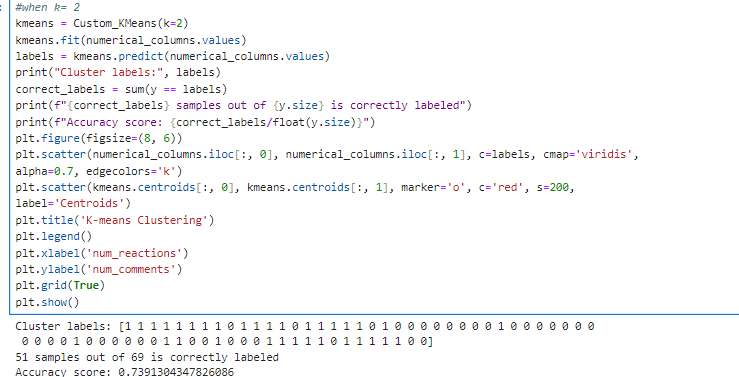
The code evaluates the accuracy of a clustering algorithm by comparing predicted cluster labels (labels) with the true labels (y). The number of correctly labeled samples is computed using sum (y == labels), and the results are printed. In this specific example, 51 out of 69 samples are correctly labeled, resulting in an accuracy score of approximately 73.91%. This accuracy score indicates the proportion of correctly assigned samples in the clustering compared to the total number of samples.

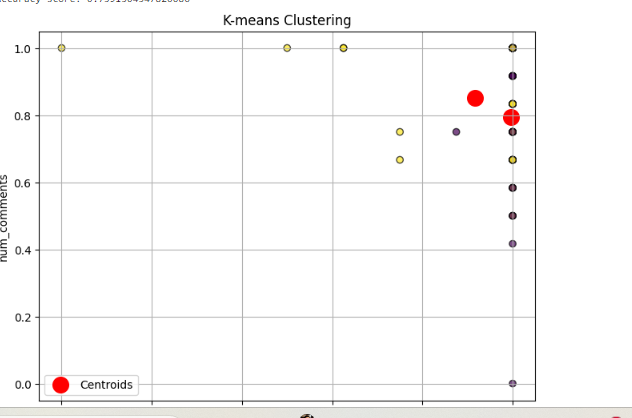


This code creates a scatter plot to visualize the results of the K-Means clustering. The scatter function is used to plot points from the first two features of the scaled dataset (numerical\_columns). Points are colored according to their cluster labels (labels) using the 'viridis' colormap, and centroids are marked in red. The plot provides a visual representation of how the algorithm has grouped data points. The plot includes labels, a legend, and grid lines for clarity. This visualization helps assess the separation and distribution of clusters in the feature space.



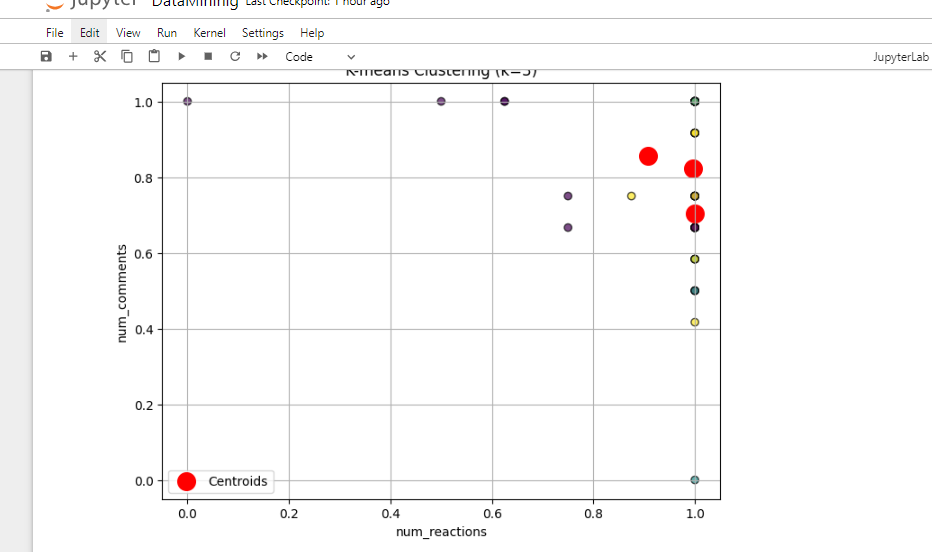
In this code, a custom K-Means clustering algorithm is applied to the scaled numerical features (numerical\_columns) with k=2. The resulting cluster labels are printed, and the accuracy of the clustering is evaluated by comparing the predicted labels with the true labels (y). The scatter plot is then generated to visualize the clustering results. Data points are colored according to their cluster assignments, and centroids are marked in red. The output shows that 51 out of 69 samples are correctly labeled, yielding an accuracy score of approximately 73.91%. The scatter plot provides a visual representation of how the algorithm has separated the data points into two clusters based on the specified features.



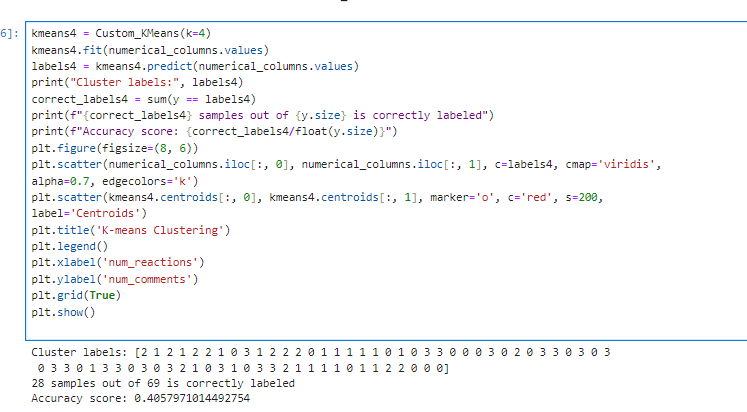


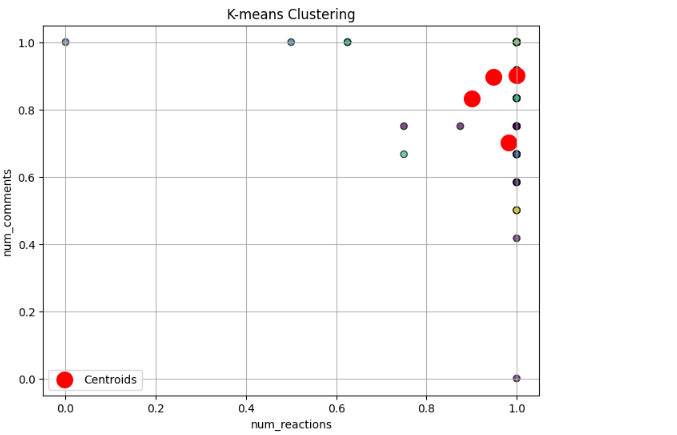
In this code, a custom K-Means clustering algorithm is applied with k=3 to the scaled numerical features (numerical\_columns). The resulting cluster labels are printed, and the accuracy of the clustering is evaluated by comparing the predicted labels with the true labels (y). A scatter plot is generated to visualize the clustering results, with data points colored according to their cluster assignments and centroids marked in red. The output shows that 15 out of 69 samples are correctly labeled, resulting in an accuracy score of approximately 21.74%. The scatter plot illustrates the separation of data points into three clusters based on the specified features. The lower accuracy indicates that the data may not naturally form distinct clusters with k=3.





In this code, a custom K-Means clustering algorithm is applied with k=4 to the scaled numerical features (numerical\_columns). The resulting cluster labels are printed, and the accuracy of the clustering is evaluated by comparing the predicted labels with the true labels (y). A scatter plot is generated to visualize the clustering results, with data points colored according to their cluster assignments and centroids marked in red. The output shows that 28 out of 69 samples are correctly labeled, resulting in an accuracy score of approximately 40.58%. The scatter plot illustrates the separation of data points into four clusters based on the specified features. The improved accuracy suggests that k=4 provides a better fit for this dataset compared to previous values of k.





The code calculates the Silhouette Score and Inertia for different values of k in the K-Means clustering. Silhouette Score measures the cohesion and separation of clusters, with a higher score indicating better-defined clusters. Inertia represents the sum of squared distances of samples to their cluster centroids.

For k=2, the Silhouette Score is 0.36, suggesting moderate cluster quality. The Inertia is 53.75.

For k=3, the Silhouette Score decreases to 0.30, indicating less distinct clusters. The Inertia is 50.26.

For k=4, the Silhouette Score further decreases to 0.16, and Inertia is 47.65. These values suggest that k=3 provides a better balance between cluster cohesion and separation in this dataset.

