Unsupervised Learning Project

Dor Boker - 209271279

April 2025

Abstract

In this project, I processed Customer Segmentation data from kaggle.com[?] to find the optimal clustering. The data was pre-processed, including dropping unique columns and rows that included NaN. In addition, categorial columns are converted into numerical columns using MCA. I ran a dimension reduction algorithm to reduce the 26-feature data. The optimal number of dimensions to reduce, along with the optimal clustering number found using the silhouette score comparing between PCA and ICA dimensions reducing algorithms, corresponding to K-Means, DBScan, and GMM clustering algorithms. The optimal reduction was to 7 dimensions using ICA, and clustering to 5 clusters. Then I performed a comparison between K-Means and GMM for the clustering, and K-Means performed better by the silhouette score. The results are plotted in a T-SNE 2D graph and a UMap 2D graph, trying to find a hidden manifold. The code for this project can be found at github.com[2].

Contents

1 Introduction

The clustering of Customer Segmentation data explored in this project might have some very useful uses. It can help define a few types of customers, what their purchasing habits are, and which commercials they like more. The clustering can help define the groups of customers, and conclude each group's behavior, then the publication will be more accurate, in the places and in the contents that reach the edge customer. It is a common assumption that people will behave similarly to others, and the goal here is to help find those groups of people.

2 Methods

2.1 DataSet

2.1.1 Customer Segmentation Data

For this project, I used customer segmentation data: "vishakhdapat/customer-segmentation-clustering" taken from kaggle.com[1]. This dataset includes 26 different customer features, including personal and economic data for 2240 customers. The data contains 2 categorical features: Education and Marial_Status.

2.2 Pre Processing

2.2.1 MCA

Multiple correspondence analysis. This algorithm is used to convert the categorical features into numerical ones. The MCA algorithm used two categorical columns and replaced them with 1 numerical column.

2.2.2 Standardization

Used with *sklearn.preprocessing.StandardScaler*[3] to normalize the values of the features, so the highly scaled ones will have less effect on clustering algorithms based on distances.

2.3 Dimensions Reduction

2.3.1 PCA

- Principal component analysis - Used to reduce the number of dimensions of the data and keep the most varianced in a new coordinate system. PCA was used in this project with different numbers of dimensions to keep.

2.3.2 ICA

- Independent component analysis - Used compared to PCA to reduce the dimension of the data, using a statistically independent assumption. ICA was used in this project with different numbers of dimensions to keep.

2.4 Clustering

2.4.1 K-Means

K-Means is an iterative, centroid-based clustering algorithm that partitions a dataset into similar groups based on the distance between their centroids. Used to decide which number of dimensions to keep, and in different number of clusters, and then for the final clustering with 5 clusters.[4]

2.4.2 GMM

Gaussian mixture is used to find the optimal number of clusters and dimension to reduce.

2.4.3 DBScan

- Clustering based on density. Used with epsilon from 1 to 4.5, and minimum samples from 1 to 4. At each run, the result was converted to the number of clusters mapped to silhouette score. The default score to fill the heatmap was 0.

2.5 Tests

2.5.1 Silhouette Score

Used to choose the best clustering and dimension reduction algorithm[5].

2.5.2 Paired T Test

Checks the null hypothesis that the means of the two groups of samples are equal. Used to compare the clustering algorithms. Alpha is set to 0.05, so the confidence level is 95

2.6 Visualization

2.6.1 T-SNE

This algorithm is used to visualize the data in a 2-dimensional graph. Called with perplexity of 30 and maximum iterations of 250 to limit the run time.

2.6.2 UMap

This algorithm is used to visualize the data in a 2-dimensional graph, trying to find a manifold pattern for better view.

3 Results

3.1 Pre Processing

3.1.1 Data Manipulation

The data needs to have some pre-processing before it can be used with the common algorithms. The following was dropped from the dataset:

- Unique columns: ID, $Dt\_Customer$.
- Rows contained NaN.

3.1.2 Standardization

After that, the data was standardized so high-scaled columns won't get extra weight in the computations.

3.2 Finding Optimal Dimension Reduction and Clusters Number

To find the optimal dimension reduction to get the best clustering, a multi variable comparison has been made. The comparison included PCA and ICA dimension reduction algorithms, along with $K-Means$, $DBScan$, and GMM clustering algorithms. Dimensions tested from 5 to 18, and cluster number tested from 2 to 18, except $DBScan$, which runs differently as described in the methods section. Each combination was tested with $silhouettescore$, which is set to 0 by default in combinations that didn't appear in the $DBScan$ run. Figure 1 contains the heatmap of the result of every combination. The optimal results, means the maximum silhouette score, received for 7 dimensions, 5 clusters using ICA algorithm for dimensions reduction and GMM for clustering.

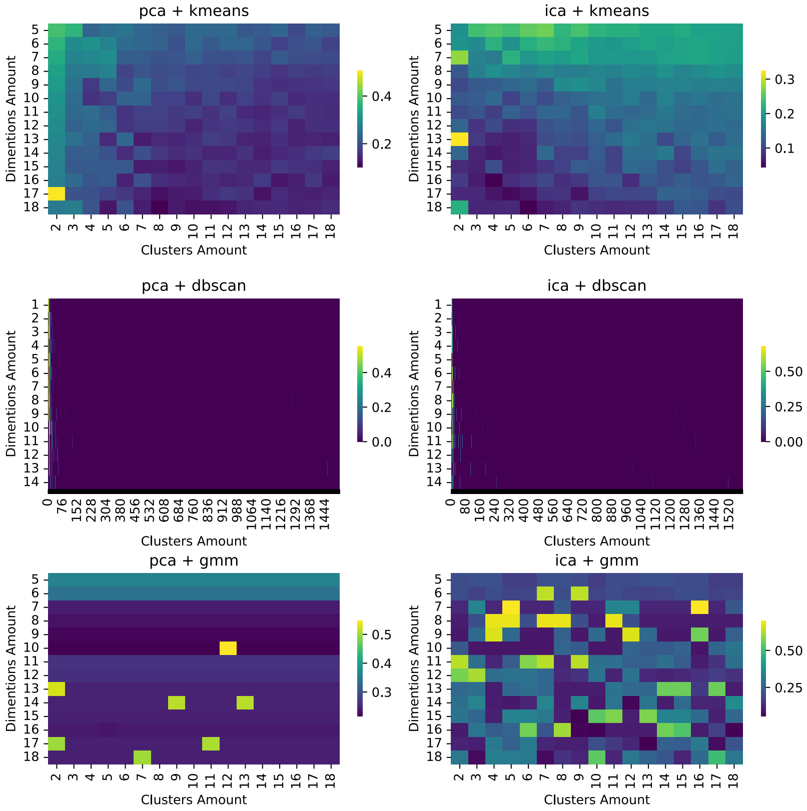

Figure 1: Combinations of Dimension Reduction: PCA / ICA with Clustering Algorithms: K-Means / DBScan / GMM

3.3 Comparing Clustering Using Paired T-Test

After choosing the optimal number of dimensions after reduction, optimal number of clusters, and optimal dimension reduction algorithm, I checked the optimal algorithm GMM in comparison with K-Means over 128 runs each on the same processed dataset. With the 2 groups of samples, scores for each algorithm on the same dataset, the null hypothesis was that the difference between the means of each group of samples in 0. For this, I ran the paired T-Test and rejected the null hypothesis with p-value $5.422998556940823e-41$, which is lower than alpha (0.05), than there is a significant difference between the means. The mean of K-Means scores was higher than the mean of GMM scores, as can be observed from Table 1. Then I can conclude that it will be the optimal clustering in this case.

P-Value	5.422998556940823e-41
K-Means scores Mean	0.2018806140581344
GMM scores Mean	0.14514109029726902

Table 1: Paired T-Test

3.4 Visualization

3.4.1 T-SNE

To view the results, the T-SNE algorithm was used to reduce the 7 dimensions to 2 dimensions. The labels that got the best silhouette result of K-Means are shown in Figure 2 on the T-SNE reduced dimensions.

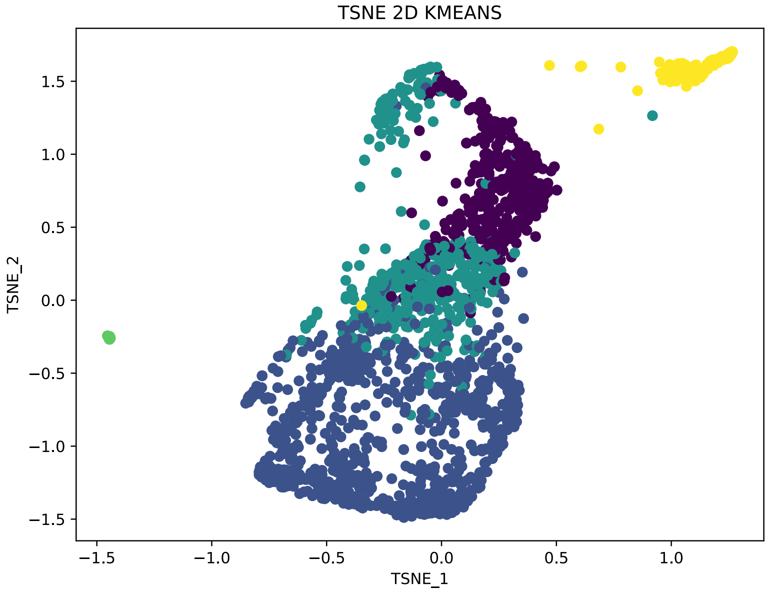

Figure 2: K-Means high score labels on TSNE 2D

3.4.2 UMap

The T-SNE plot as shown in Figure 2 might have some hidden manifold. To check this I run the UMap algorithm on the 7 dimensions data (the previous ICA Result) and plotted it with the K-Means high score labels as can be seen in Figure 3.

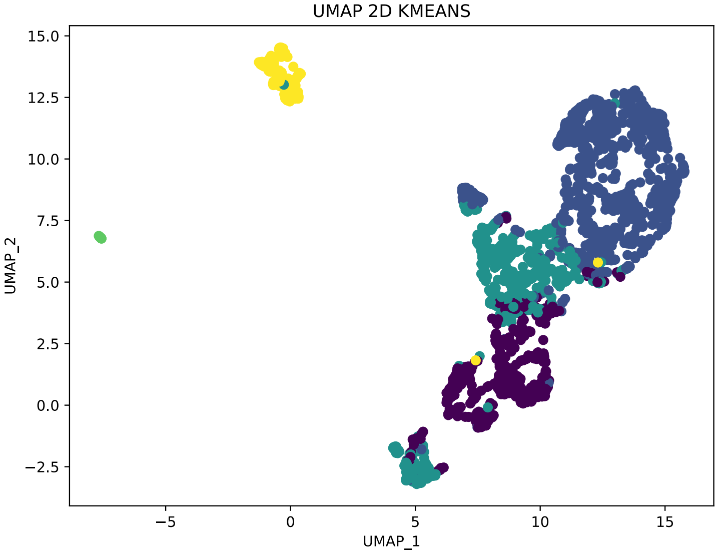

Figure 3: K-Means high score labels on UMap 2D

4 Discussion

In this project, the optimal dimensions were reduced using the ICA algorithm to 7 dimensions, then using K-Means with 5 clusters. For future improvements, I would suggest checking more clustering algorithms and running the $DBScan$ algorithm in a wider range of epsilon and minimum samples, with a combination of different score types. For the pre-processing, a combination of PCA and ICA might have led to better clustering results. In addition, it looks like there is a hidden manifold that the data relies on, so UMap might also be used even in the pre-processing stage before the clustering, and not only at the visualization.

References

[1] Vishakh Patel, *Customer Segmentation : Clustering*, https://www.kaggle.com/datasets/vishakhdapat/customer-segmentation-clustering, 2024.

[2] Boker Dor, *Unsupervised Learning Project*, https://github.com/dorbo/Unsupervised-Learning-Project, 2025.

[3] scikit-learn API Reference, *StandardScaler*, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

[4] Eda Kavlakoglu, Vanna Winland, IBM, *What is k-means clustering?*, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html https://www.ibm.com/think/topics/k-means-clustering.

[5] scikit-learn API Reference, *Silhouette Score*, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette$_score.html$