Dimensionality Reduction Project Report

Amr MOHAMED
CY Tech/ CY Cergy Paris University
mohamedamr@cy-tech.fr

Abstract

In this paper, we are going to see the different techniques applied to perform Dimensionality Reduction or dimension reduction to transform data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. In addition, we will be working with K-means and Hierarchical Clustering techniques to divide datasets into different clusters.

Keywords: Dimensionality-Reduction, PCA, MCA, K-means, HC

1 Introduction

During the past decade, Data Science and Machine Learning have become very important fields in all industries as they show us the real insights behind the different sources of data. As a result of this, a technique called Dimensionality Reduction has aroused to ease the processes of Data Science and Machine Learning as it reduces the processing power used to process the data by transforming it from a high-dimensional space to a low-dimensional space so that the low-dimensional representation retains some significant properties of the original data. In this paper, we are going to see the different techniques applied to perform Dimensionality Reduction, and see how their results can be interpreted.

2 Presentation of the dataset

The data that was used in this study is artificial data created to simulate the users' activity on dating applications. The dataset consists of 16 columns and 3000 rows. The columns of the dataset are represented below in Table 1.

Column	Description
userid	Integer, id of the user
date.crea	Date, date of the creation of the account
score	Float, score of the profile
n.matches	Integer, total number of matches the user has had since account creation (with conversation)
n.photos	Integer, number of photos on the profile
last.up.photo	Date, last time the user updated profile pictures
last.pr.update	Date, last time the user updated profile text
last.connex	Date, last time the user was connected
gender	Integer, gender of the user
sent.ana	Float, sentiment score for the text of the profile
length.prof	Integer, number of words in the profile text
voyage	Boolean, Keyword voyage found in the profile text
laugh	Boolean, Keyword laugh found in the profile text
photo.keke	Boolean, one of the profile pics comports a photo without a T-shirt /with sunglasses / selfie in an elevator
photo.beach	Boolean, one of the profile pics comports a photo taken on the beach

Table 1: Columns of the dataset

3 Identifying correlations in the variables

We started by inspecting the distributions of our numerical variable score.

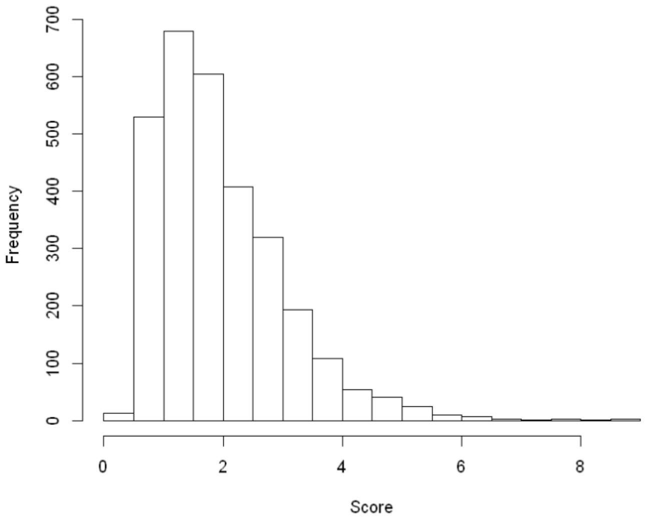

Figure 1: The score of users distribution

As we can see from Figure 1, the users' score data doesn't follow the normal distribution, for further verification, we applied the quantile-quantile plot to verify our assumption.

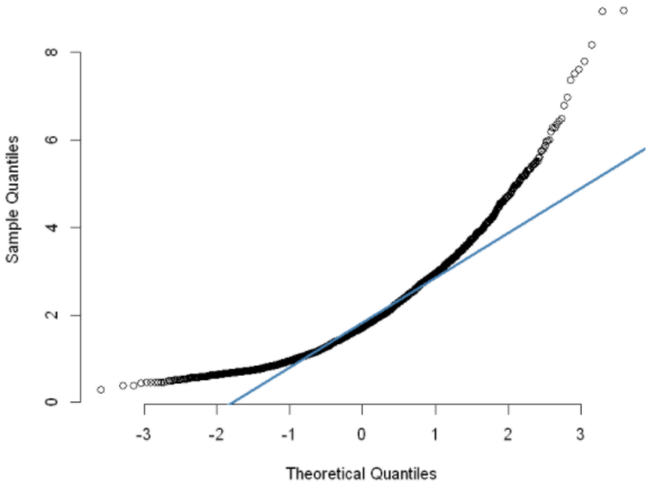

Figure 2: quantile-quantile plot of users' scores

From Figure 2, we can see that the data doesn't follow the normal distribution as the data points don't fit the line, and our assumption about the abnormality of the scores data was verified. Therefore, we proceeded to perform a log-transformation for the scores data.

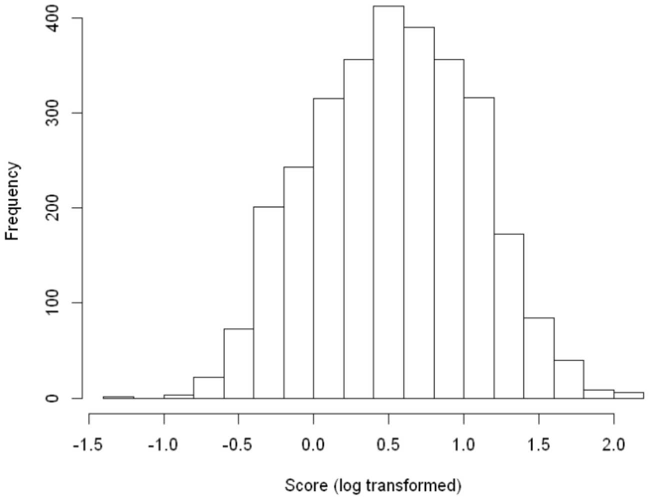

Figure 3: Log-transformed users' scores distribution

As we can see in Figure 3 after the log-transformation was performed on the data, the data now looks to follow the normal distribution, to verify our observation, we plotted a quantile-quantile plot for the log-transformed data.

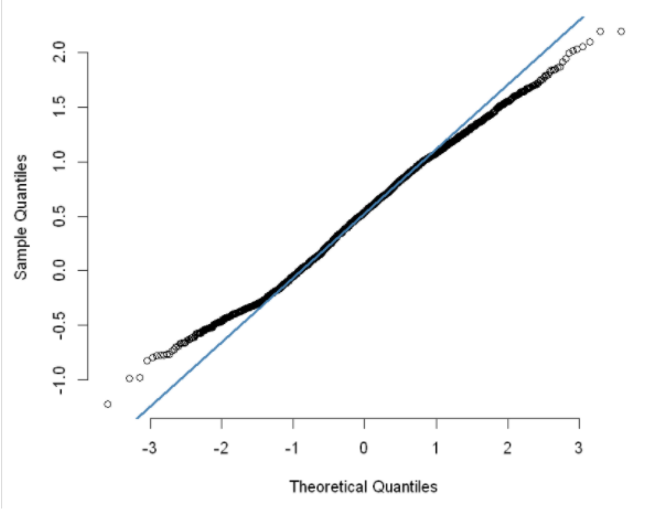

Figure 4: quantile-quantile plot of log-transformed users' scores

From Figure 4, we can see that the scores data points almost fit the line perfectly, and the log-transformed scores data is normally distributed, and now we can add the log-transformed data of the scores data as a new column to our dataset.

Afterward, We proceeded to check the distribution of the numerical variable number of matches.

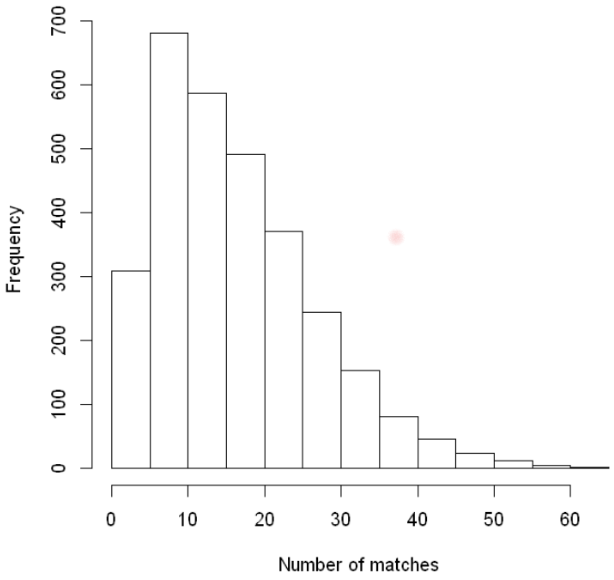

Figure 5: Number of matches of users distribution

As we can see from Figure 5, the users' number of matches data doesn't follow the normal distribution, to verify this assumption, we applied the quantile-quantile plot.

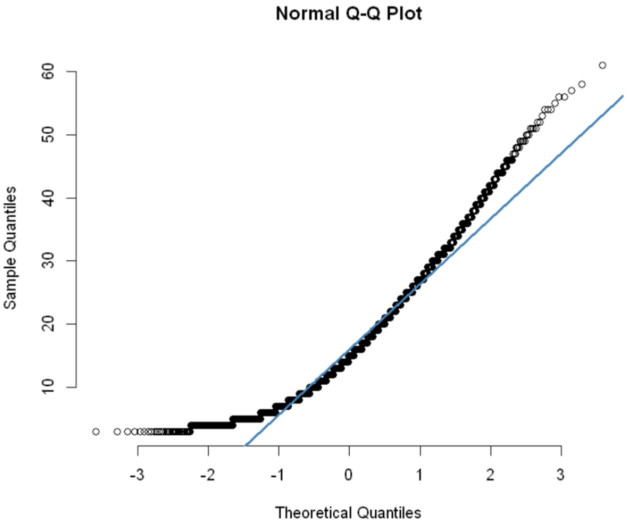

Figure 6: the quantile-quantile plot of users' number of matches

From Figure 6, we can see that the data doesn't follow the normal distribution as the data points don't fit the line, and our assumption about the abnormality of the number of matches data was verified. Therefore, we proceeded to perform a log-transformation for the number of matches data.

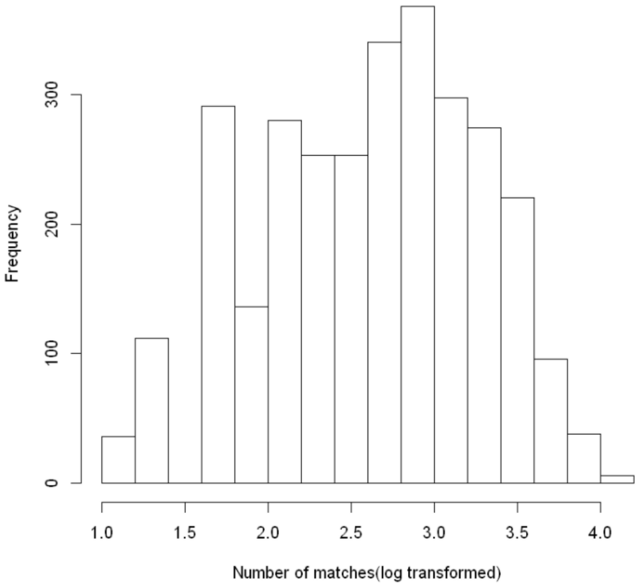

Figure 7: Log-transformed users' number of matches distribution

As we can see in Figure 7 after the log-transformation was performed on the data, the data now looks to follow the normal distribution, but with a little spike on the left-hand side of the histogram, to verify our observation, we plotted a quantile-quantile plot for the log-transformed data.

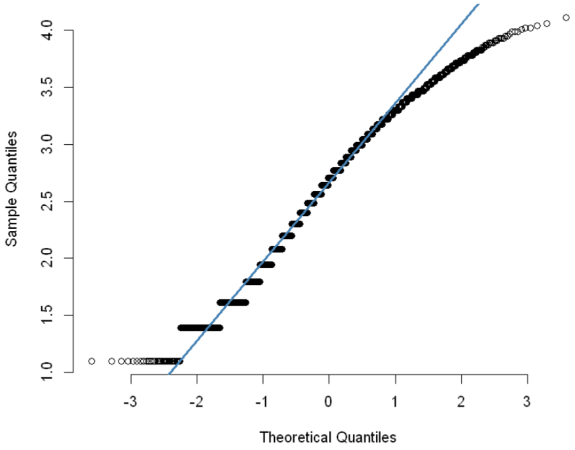

Figure 8: the quantile-quantile plot of log-transformed users' number of matches

From Figure 8, we can see that the number of matches data points better fit the line than in Figure 6, and the log-transformed number of matches data is almost normally distributed, and now we can add the log-transformed data of the number of matches data as a new column to our dataset

Afterward, we started looking for the different correlations between the columns of the dataset, so we started by plotting a heatmap as shown in the figure below.

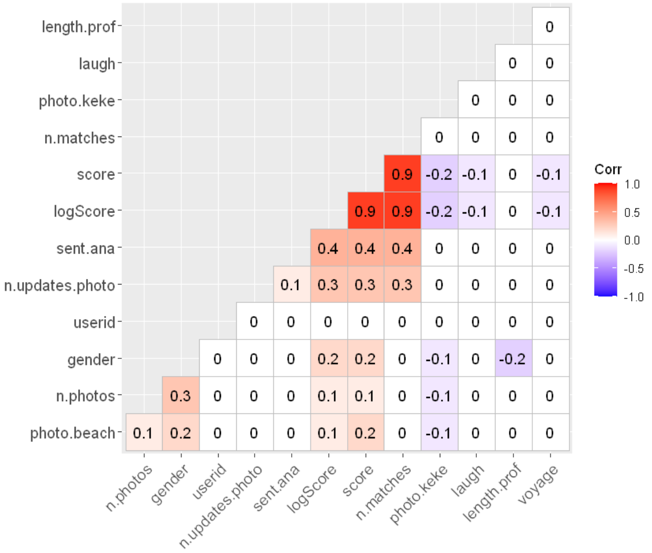

Figure 9: Correlation between the numerical variables

From Figure 9, we can see a very strong positive correlation between the score of the users and the number of matches with a correlation coefficient of 0.9, and a weak negative correlation between the photo.keke and the score variables with a correlation coefficient of -0.2, which shows us that there might be a little indirect influence of photo.keke variable on the number of matches through the score of the user.

To verify our assumptions about the strong correlation between the number of matches of a user and his/her score, we performed a Spearman's rank correlation test on the two variables. We found that our assumption is correct and there is a very strong correlation between the user's number of matches and score with rho(ρ) of 0.92 (p-value $<$ 2.2e-16).

For further verification, we performed Pearson's product-moment correlation test on the same variables and found that the user's number of matches and score have a very strong correlation with a correlation coefficient of 0.9 (p-value $<$ 2.2e-16).

4 Dimensionality Reduction

4.1 Principal Component Analysis

Moreover, to have a better of our different variables and how do they relate to each other, we proceeded to perform dimensionality reduction techniques. We started with the continuous numerical variables logScore, logNofMatches, sent.ana, length.prof, and n.updates.photo from the dataset and performing on them Principal component analysis.

Firstly, we proceeded to plot the variables' circle of correlation. In a correlation circle, all of the original variables included in the PCA are plotted against two of the principal components (the 2 principal components explaining the highest percentages of the variance), which are represented as the x- and y-axes.

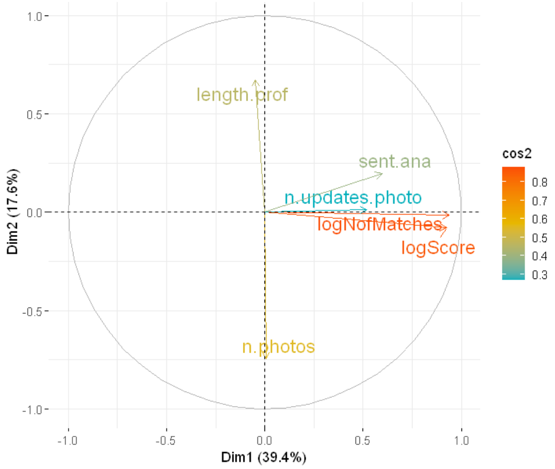

Figure 10: PCA circle of correlations of the variables

As we can see in Figure 10, the log-transformed number of matches and the log-transformed score are positively correlated as found earlier, and since both of the variables have high cos2, they are highly represented by PC1, and they have no relation with length.prof variable which is almost perpendicular to both of them. Moreover, we can that length.prof is highly represented by PC2 because of its high cos2, but is negatively correlated with the n.photos variable. Finally, we can see that both sent.ana and n.updates.photo are not perfectly represented by PC1 because of their low cos2.

Afterward, we proceeded by plotting the scree plot to determine the number of principal components to retain after PCA.

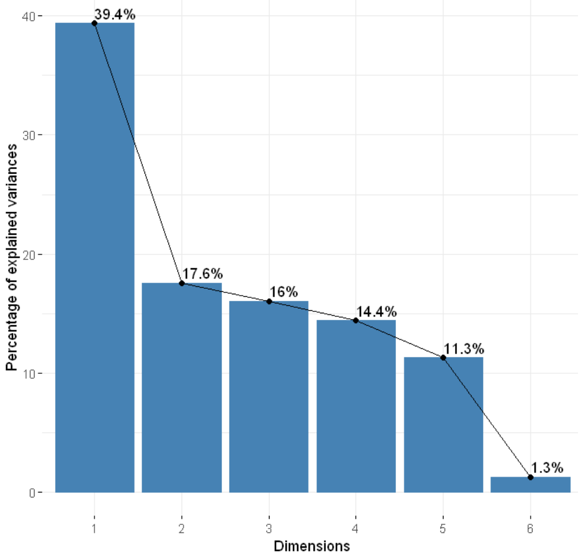

Figure 11: PCA scree plot with the percentage of explained variance.

From Figure 11, we might want to stop at the Third principal component as 73% of the information (variances) contained in the data are retained by the first three principal components, and the aim is to retain the first n principal components explaining at least 70-80% of the variance.

Moreover, we were interested in seeing the individual map of the PCA and how the individuals are represented by the PCs.

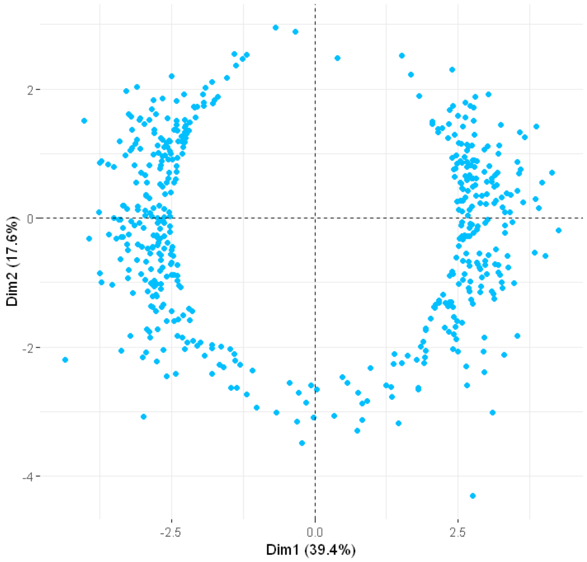

Figure 12: PCA individuals map by a sample of individuals

As we can see in Figure 12, most of the individuals have high cos2, which means that they are highly represented by both of the PCs, and that's because that PC1, PC2, and PC3 represent 73% of the variance contained in the data.

Moreover, we were interested in seeing the PCA biplot to see how is the distribution of the individuals and the variables together in the PCA with the first 2 PCs.

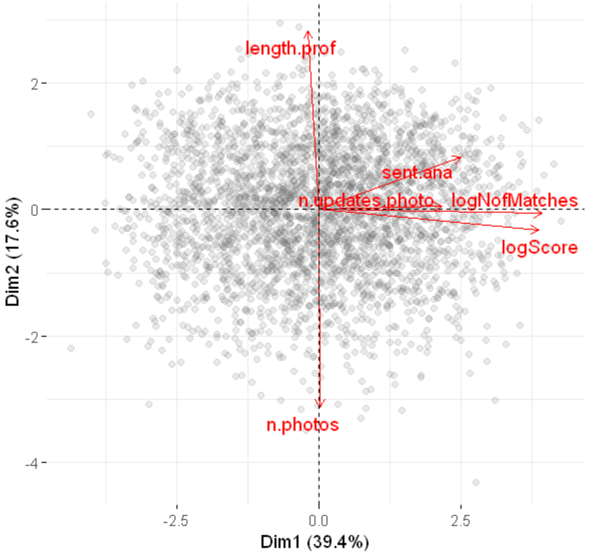

Figure 13: PCA Biplot of variables and individuals

As we can see in Figure 13, it is clear that the variables logNofMatches and logScore are strongly correlated and that they have no correlation with the length.prof variable.

To sum up our interpretations of the PCA, we were interested in verifying all the interpretations throughout the table of loadings.

	PC1	PC2	PC3	PC4	PC5	PC6
logScore	0.6	-0.1	0	0	-0.4	-0.7
logMatches	0.6	0	0	0	-0.3	0.7
sent.ana	0.4	0.2	0	-0.5	0.7	0
length.prof	0	0.7	-0.7	0.1	-0.1	0
n.upd.photo	0.3	0	0.1	0.8	0.4	0
n.photos	0	-0.7	-0.7	0	0.2	0

Table 2: PCA table of loadings

From Table 2, we can see that PC1 has high positive loadings of the variables logScore, logMatches, and sent.ana , while PC2 has high positive loading of the variable length.prof and high negative loading of n.photos.

4.2 Multiple Correspondence Analysis

Moreover, we proceeded to perform dimensionality reduction techniques on the categorical variables of the dataset. We used the variables gender, voyage, laugh, photo.keke, and photo.beach from the dataset and performed Multiple Correspondence Analyses.

Firstly, we start by plotting the scree plot to determine the number of dimensions to retain after the MCA.

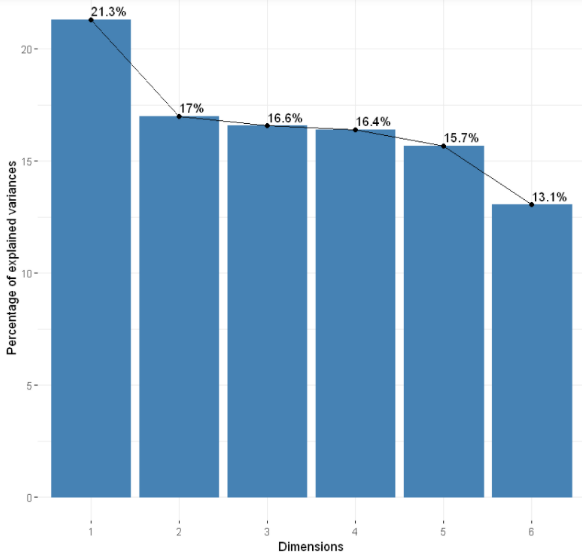

Figure 14: The percentages of inertia explained by each dimension

From Figure 14, we might want to stop at the fifth dimension. 87% of the information(variances) contained in the data are retained by the first five dimensions.

Moreover, we were interested in seeing the individual map along with the variables of the MCA, so we plotted the MCA biplot below.

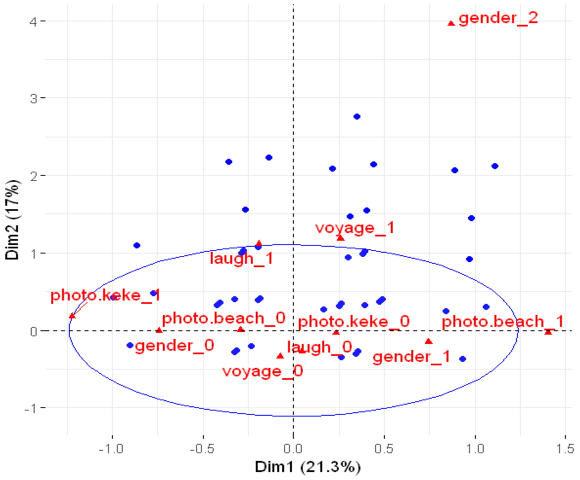

Figure 15: MCA biplot of individuals and variable categories

As we can see in Figure 15, the distance between any row points(blue points) or column points (red points) gives a measure of their similarity. Row points with a similar profile are closed on the factor map. The same holds for column points. In addition, we can see that 21.3% of the variance in the data can be represented through Dim1 while 17% of the variance in the data can be represented by Dim2.

5 k-means Clustering

5.1 k-means clustering on principal components of the analysis

In continuation to our Principal component analysis, we moved forward to k-means clustering to find groups in the data that weren't explicitly defined. It was decided to use 2 clusters (subsection 5.3).

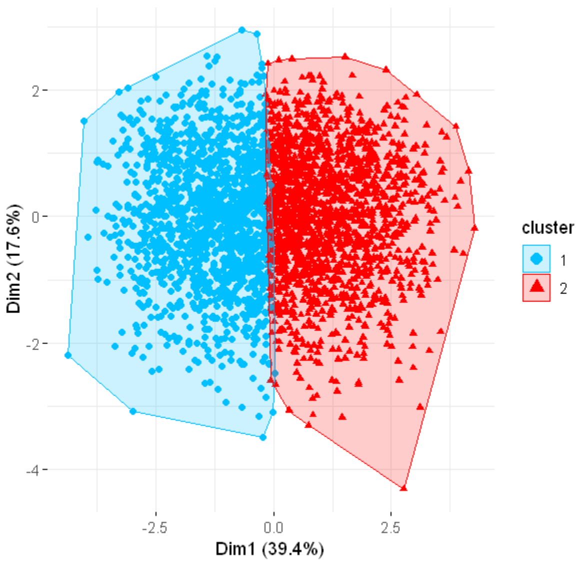

Figure 16: K-means clustering on principal components one and two

From Figure 16, we can see that the data was clustered into 2 clusters as pre-specified in the algorithm labeled as Cluster 1 and Cluster 2. Recalling Figure 10, it was seen that 60% of the variance in each of the number of matches of the user and his score was explained by the first principal component, and since Cluster 1 lies on the positive part of the first principal component, this implies that the users that were clustered to Cluster 1 have higher scores and higher number of matches that the users that were clustered to Cluster 2.

5.2 How do k-means work?

K-means clustering is an unsupervised machine learning technique for dividing given dataset clusters, where k is the pre-specified number of clusters. To cluster the data points, K different randomly-initiated points in the data called centroids, and assigns every data point to the nearest centroid, once the centroids stop moving between the data points and are centered in each of the different clusters, our clustering algorithm stops.

K-means classifies objects into multiple groups, such that objects in the same cluster are as similar as possible, while objects in different clusters are as dissimilar as possible. Each cluster is represented by its center which corresponds to the average of the points assigned to the cluster.

The main advantages of k-means clustering are the guaranteed convergence and its specialization in clusters of different sizes and shapes. The main disadvantages of k-means clustering are its sensitivity for outliers and the difficulty of k-value prediction.

5.3 Number of clusters choice

5.3.1 Elbow Method

The Elbow method is a very popular technique that works on performing k-means clustering for a range of clusters k and each value, we calculate the sum of squared distances from each point to its assigned centroid.

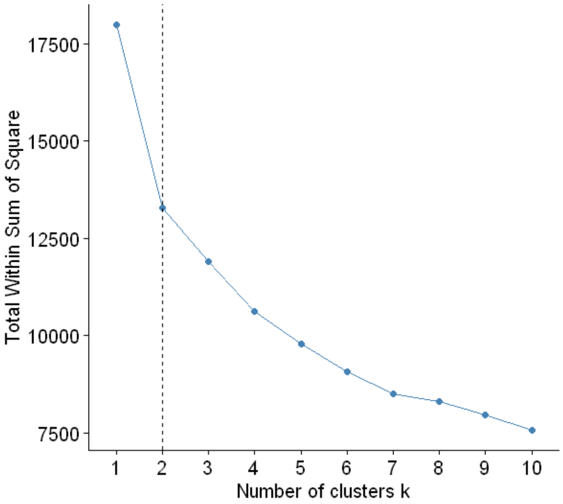

Figure 17: Elbow method to find the optimal number of clusters for k-means

From Figure 17, according to the Elbow method we can argue that the optimal number of clusters is 2, and therefore we can use 2 centroids to cluster the data points.

5.3.2 Silhouette Method

Silhouette analysis allows you to calculate how similar each observation is with the cluster it is assigned relative to other clusters. This metric ranges from -1 to 1 for each observation in your data and can be interpreted as follows:

- Values close to 1 suggest that the observation is well matched to the assigned cluster.

- Values close to 0 suggest that the observation is borderline matched between two clusters.

- Values close to -1 suggest that the observations may be assigned to the wrong cluster.

We can determine the number of clusters K using the average silhouette width. We pick the K which maximizes that score.

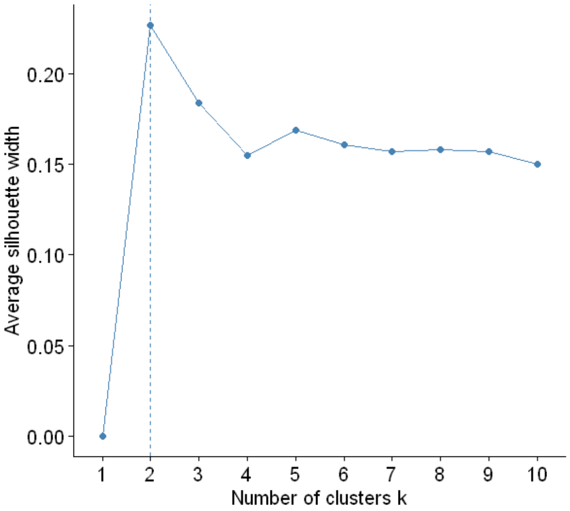

Figure 18: Silhouette method to find the optimal number of clusters for k-means

From Figure 18, according to the Silhouette method we can argue that the optimal number of clusters is 2, and therefore we can use 2 centroids to cluster the data points.

5.3.3 Gap statistic Method

The gap statistic is a method for approximating the correct number of clusters k for k-means clustering.This is done by assessing a metric of error (the within-cluster sum of squares) concerning our choice of k.

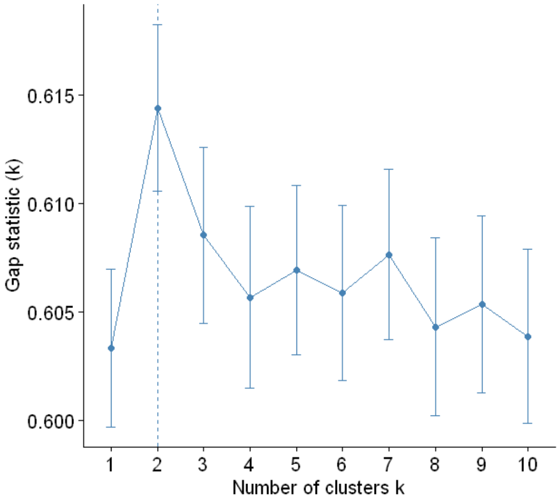

Figure 19: Gap statistic method to find the optimal number of clusters for k-means

From Figure 19, again, according to the Gap Statistic, the optimum number of clusters is the k=2, and since the three methods used in this subsection all suggested the number of clusters to be equal to 2, we pre-specified k=2 for the k-means clustering performed in subsection 5.1.

6 Hierarchical Clustering

6.1 HCPC on continuous variables

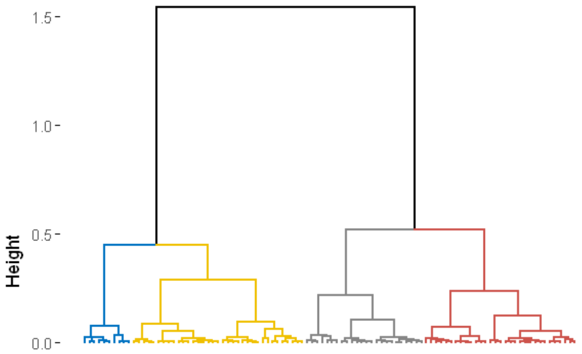

Figure 20: Dendrogram generated by hierarchical clustering performed on the first three principal components

6.2 How HC works?

Hierarchical clustering is an algorithm that groups similar observations into clusters. It starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps:

- 1 - identify the two clusters that are closest together.
- 2 - merge the two most similar clusters.

This iterative process continues until all the clusters are merged.

6.3 Pros and cons of HC compared to k-means

The major advantages of Hierarchical clustering are that it does not require to specify in advance the number of clusters to generate, ease of handling of any form of similarity or distance. However, the main disadvantage of HC is that it requires the computation and storage of an n×n distance matrix. For very large datasets, this consumes a lot of memory.

References

HCPC: *http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/117-hcpc-hierarchical-clustering-on-principal-components-essentials/*

PCA:*http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/118-principal-component-analysis-in-r-prcomp-vs-princomp/*

Oracle AI Data Science Blog ”Introduction to K-means Clustering”:*https://blogs.oracle.com/ai-and-datascience/post/introduction-to-k-means-clustering.*

Predictive ['hacks'] ”K-Means Elbow Method Code For Python” :*https://predictivehacks.com/k-means-elbow-method-code-for-python/*

Predictive ['hacks'] ”How To Determine The Number Of Clusters For K-Means In R” :*https://predictivehacks.com/how-to-determine-the-number-of-clusters-of-k-means-in-r/*