Unsupervised Machine Learning

Final Assignment

What are the most important features in the market for home loans and how many distinct groups are there on the demand side?

February 17, 2023

630510

1 Introduction

The market for home loans is a complex and dynamic industry, with a wide range of factors influencing both supply and demand. One important aspect of this market is understanding the most important features in the home loan market, and how many segments there are in the demand side. By identifying these features and grouping customers based on the available data, lenders can better tailor their products and services to meet the needs of specific customer segments. In this analysis, we aim to explore the market for home loans and answer two key questions:

What are the most important features in the market for home loans and how many distinct groups are there on the demand side?

By answering these questions, we hope to provide insights that can inform strategic decision-making in the home loan industry.

2 Data

The data-set is also publicly available on Kaggle (Analytics Vidhya, 2016). This dataset is from a housing finance company and is used to automate the loan eligibility process based on customer data from an online application form. It contains 614 data points at the individual level with 12 features. Only four of these variables are numeric variables: Income of applicant, income of co-applicant loan amount, loan amount terms. A short summary for them can be seen in the table below. The remaining variables are gender, employment status, number of dependents, marriage status, education and credit history of the applicant as well as property area and loan status. All of these variables are categorical variables. Our initial checks suggest that we have complete data for only 480 of individuals. We conduct our analyses on this reduced data.

variable name	mean	variance	description
ApplicantIncome	5403	610	monthly income for applicants
LoanAmountTerm	342	65.1	loan amount terms in months
LoanAmount	146	85.5	loan amount in thousand dollars
CoapplicantIncome	1620	2920	monthly income for co-applicants

3 Methods

To answer our research question, we will utilize factor analysis of mixed data (FAMD) (Saporta, 1990; Pagès, 2004) which is a generalized version of principal component analysis (PCA) as a first step. In the next step, we will implement hierarchical clustering (Ward Jr, 1963) where we implement agglomerative and divisive clustering.

Since the aim of the first part of the proposed research question is to find the most central features in the given dataset, PCA will be a good starting point as it's goal is to reduce the dimensionality of given data from p numerical variables to $r < p$ while preserving as much information as possible. We should note that PCA analysis is mostly designed for data-sets where we only have

numeric variables. On the contrary, the majority of the features in our data-set consists of categorical variables. Since the challenge with categorical variables is to find a suitable way to represent distances between variable categories and individuals in the factorial space, we believe that FAMD is doing a better job than PCA as PCA cannot deal with the categorical variables in the same way as it deals with numeric variables. Instead, the FAMD algorithm wants to give the exact same weight to all the variables, numerical or categorical, when searching for the principal components.

To start the analysis of FAMD, one should first calculate the square correlation coefficient $r^2(\mathbf{X},\mathbf{Y})$ for correlation among numeric variables $\mathbf{Y}$ and $\mathbf{X}$. For the relation between categorical variable $\mathbf{A}$ and a numerical variable, the square correlation ratio $\eta^2(\mathbf{A},\mathbf{Y})$ should be calculated. Both of these correlations takes value between 1 and 0. More specifically, they can be calculated as follows:

$$r(\mathbf{X},\mathbf{Y}) = \frac{\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i-\bar{y})^2}} = \frac{cov(\mathbf{X},\mathbf{Y})}{\sigma_X \sigma_Y}, \tag{1}$$

where N is the number of rows. We also have:

$$\eta^2(\mathbf{A},\mathbf{Y}) = \frac{\sum_{m\in M_A}\sum_{k\in K_m}(\overline{y_{*,m}}-\bar{y})^2}{\sum_{m\in M_A}\sum_{k\in K_m}(\overline{y_{k,m}}-\bar{y})^2}, \tag{2}$$

where M_A modalities of A which is equal to the number of different answers for categorical variable $\mathbf{A}$, K_m is data points taking modality m, $\overline{y_{*,m}}$ the mean of $\mathbf{Y}$ from K_m.

After getting correlation measures for both type of variables, FAMD aims to find the principal components as follows:

$$C_k = \underset{\|w\|=1,\bar{w}=0}{\arg\min}\ (\sum_{j=1}^{P} r^2(w,v_j) + \sum_{j=P+1}^{P+Q}\eta^2(w,v_j) | w \perp Vect(C_1,...,C_{k-1})), \tag{3}$$

where P is the total number of numerical variables, Q is the total number of categorical variables, C_k is the unit vector centered in R^n and orthogonal to the previous principal components. K is the maximum number of components that we allow in the reduced data. The difference with regular PCA is that, FAMD can treat categorical variables in the same way as numeric variables after the calculation of $\eta^2(\mathbf{A},\mathbf{Y})$ for each $\mathbf{A}$. Whereas in PCA, even though we one-hot encode all the categorical data, it would be impossible to give a similar weight to all the initial variables over the calculated components.

The second step is intended to answer the second part of the proposed question: "How many distinct groups are there on the demand side?". For this part, we will implement divisive and agglomerative clustering. Hierarchical clustering methods doesn't require us to specify the number of clusters in advance. That's why, we think it is more advantageous to use them when we do not have any starting point or previous literature knowledge on potential number of clusters, like in our case.

Agglomerative clustering first calculates the distances between all pairs of data points and stores them in a distance matrix. It then finds the two data points that are closest to each other and merges them into a cluster. This new cluster is then used to calculate the distances between other clusters based on the chosen linkage criterion. In this work, we decided to use complete linkage which can be calculated as follows:

$$d(G,H) = \max_{i \in G, i' \in H} d_{ii'}, \tag{4}$$

where G and H are two clusters. The algorithm repeats this process, merging the closest clusters at each step, until all data points are in a single cluster. In this way, the algorithm builds a hierarchical tree-like structure of clusters, called a dendrogram, that can be used to visualize the relationships between the data points. Divisive clustering, on the other hand, starts with all the data points in a single cluster and recursively splits the clusters until each data point is in its own cluster.

There are some important things to consider in this analysis. Since FAMD results will reflect the relative scales of the variables, we must standardize our variables. To decide on the optimal number of features, we will utilize contribution and scree plots (Cattell, 1966) for both type of variables.

To pin down the optimal number of clusterings, we will utilize scree and silhouette plots for both clustering techniques. We know that agglomerative clustering is doing a better job in discovering small clusters whereas divisive methods are preferable if we naturally have larger clusters. Since we do not have any previous knowledge, we wanted to deploy both and do a comparative analysis. Lastly, we will analyse the dendrogram of our analysis.

4 Results

Before explaining the major results from our analyses, there are several points worth mentioning. Firstly, due to the nature of unsupervised machine learning techniques, there is often no single and clear goal for the analysis. In this work, we aimed to find the ”building blocks” of the data we had by employing FAMD and hierarchical clustering. Since there is no standard benchmark in these methods, heuristic arguments are used to evaluate the results and they can be subjective. In this work, we present our results by considering different scenarios that a researcher can face.

First, we will present our results on FAMD. In figure 1, the scree plots for FAMD and regular PCA can be seen. Since PCA cannot deal with the categorical variables as smoothly as FAMD, the results are different. From the plots, it is clearly seen that FAMD suggests 4 features are enough to explain significant part of the data whereas in PCA plot we see that 5 or 7 features are better choices. Due to the reasons that we have already explained in the methodology part, we continue with FAMD results.

In the figure 2, we can see the impact of each variable on explaining the variance of the whole data. On the left side, numeric variables are represented. We can see length of each vector corresponds to this impact. On the other hand, the graph on the right side shows the categorical variables. Since the distance between the answers of each category is not as interpretable as in numeric vari-

ables, the vector representation is not being used here. We can see the relative importance for each answer from categorical variables. For instance, being female has much more contribution in explaining the variance compared to being a graduate. We can see a clearer picture in figure 3. Loan amount, marriage status, dependent number, income and gender are the features which contribute more to first dimension compared to the average contribution which is indicated by the red dotted line. For dimension 2, we only see the loan status and credit history.

Overall results from FAMD tells us the following: This data can be explained by 4 or 5 features and good candidates for these features are the following: Loan amount, applicant income, marriage status, gender and loan status. On top of these findings, we have also run a correlation check among several variables. More detailed results can be seen in the appendix. One thing to note is that applicant income and loan amount are very correlated, therefore we can decide to keep only one of them if we need to shrink the data further. In the appendix, readers can also see figures on how gender and marriage categorizes individuals in this dataset.

Next, we move to our analyses on clustering. One could also do clustering on the reduced data after FAMD but this work aimed to check the clustering on the overall market. In figure 4, several measurements for optimal cluster decision can be seen. The aim is to find cluster number where the average distance within clusters is low while the average distance between clusters is high. The trade-off can be seen clearly in the graphs. In the left-side, we can see the measurements for agglomerative clustering and divisive clustering can be seen on the right side. For readability, we fixed maximum number of clusters to 7. The observation numbers in each cluster can also be seen. From the tables, it can roughly be seen that having 4 clusters seems like a good candidate for both of the methods. Next, we move to scree plots. In figure 5, we see that having 4, 7 or 12 clusters are good options for agglomerative clustering while having 4, 6 or 11 are better options for divisive clustering. The next thing to look at before deciding on the optimal number of clusters is the silhouette method. The idea in this method is that we should choose the number that maximizes the average silhouette width because we want clusters that are distinctive enough to be considered separate. The optimal cluster numbers are 4, 7 or 12 for agglomerative clustering while it is 4 or 11 for divisive method. The details can be seen in the appendix. As a result, we conclude that 4 would be a robust choice. However, if researcher do not need to shrink the data to that extent, the next good number would be 12.

Lastly, we can see the dendrogram for agglomerative clustering in figure 6. It can be seen that the height of first division (4 clusters) is doing a good start. If we want more detailed information from each cluster, we can implement 12 clusters. In that case, we can see in the dendrogram that it reaches approximately 45 percentage of the total height. For more details, you can also check dendrogram for divisive clustering in the appendix.

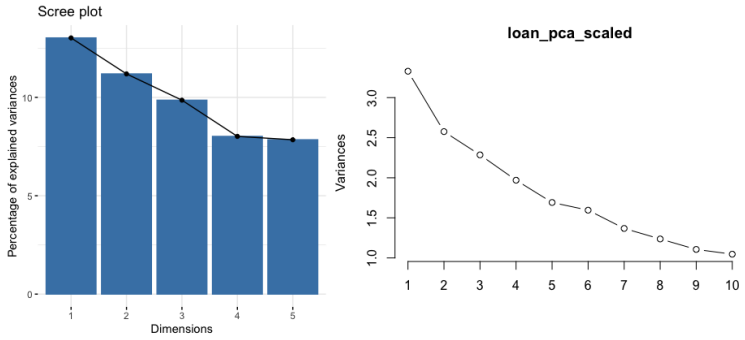

Figure 1: Left-side: Scree plot for FAMD
Right-side: Scree plot for PCA on the scaled data

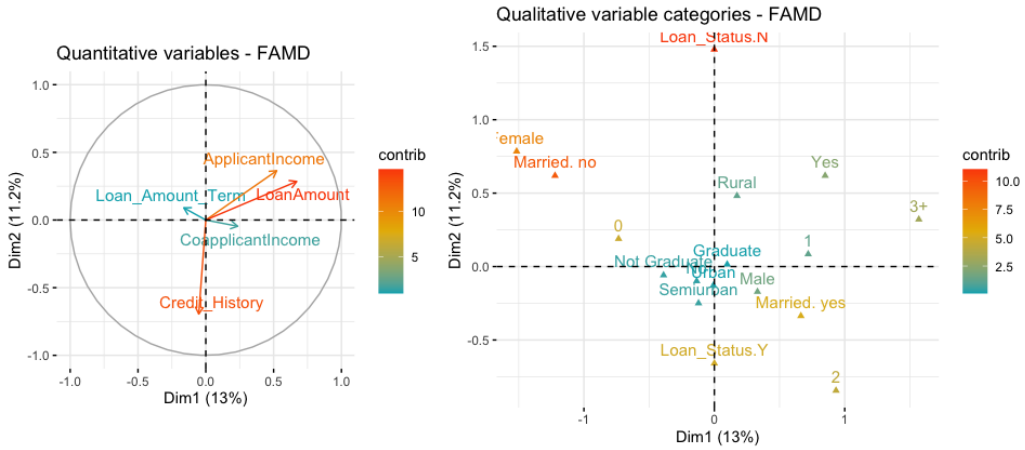

Figure 2: Left-side: Numeric variables' quality of representation on the factor map for FAMD
Right-side: Categorical variables' quality of representation on the factor map for FAMD

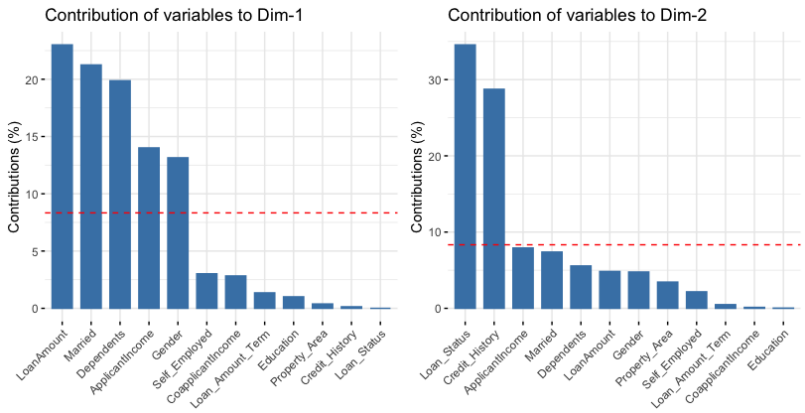

Figure 3: Left-side:Variables' contribution to the first dimension
Right-side: Variables' contribution to the second dimension

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
cluster.number	2.00	3.00	4.00	5.00	6.00	7.00
n	480.00	480.00	480.00	480.00	480.00	480.00
within.cluster.ss	50.33	47.46	34.83	33.26	31.22	25.46
average.within	0.42	0.41	0.34	0.34	0.32	0.29
average.between	0.53	0.52	0.50	0.50	0.49	0.49
wb.ratio	0.80	0.78	0.68	0.67	0.66	0.60
dunn2	1.24	1.26	1.21	1.26	1.03	1.06
avg.silwidth	0.18	0.11	0.20	0.16	0.14	0.20
Cluster- 1 size	466.00	437.00	289.00	289.00	289.00	225.00
Cluster- 2 size	14.00	29.00	148.00	148.00	89.00	64.00
Cluster- 3 size	0.00	14.00	29.00	11.00	59.00	89.00
Cluster- 4 size	0.00	0.00	14.00	14.00	11.00	59.00
Cluster- 5 size	0.00	0.00	0.00	18.00	14.00	11.00
Cluster- 6 size	0.00	0.00	0.00	0.00	18.00	14.00
Cluster- 7 size	0.00	0.00	0.00	0.00	0.00	18.00

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
cluster.number	2.00	3.00	4.00	5.00	6.00	7.00
n	480.00	480.00	480.00	480.00	480.00	480.00
within.cluster.ss	41.09	39.90	32.19	30.73	26.58	25.77
average.within	0.38	0.37	0.33	0.32	0.30	0.30
average.between	0.54	0.54	0.52	0.51	0.50	0.50
wb.ratio	0.70	0.69	0.65	0.63	0.60	0.59
dunn2	1.41	1.31	1.28	1.04	1.04	1.04
avg.silwidth	0.30	0.25	0.27	0.24	0.26	0.26
Cluster- 1 size	393.00	393.00	310.00	310.00	265.00	265.00
Cluster- 2 size	87.00	78.00	83.00	83.00	45.00	45.00
Cluster- 3 size	0.00	9.00	78.00	25.00	83.00	75.00
Cluster- 4 size	0.00	0.00	9.00	53.00	25.00	25.00
Cluster- 5 size	0.00	0.00	0.00	9.00	53.00	53.00
Cluster- 6 size	0.00	0.00	0.00	0.00	9.00	9.00
Cluster- 7 size	0.00	0.00	0.00	0.00	0.00	8.00

Figure 4: Left-side: Decision table for agglomerative clustering
Right-side: Decision table for divisive clustering

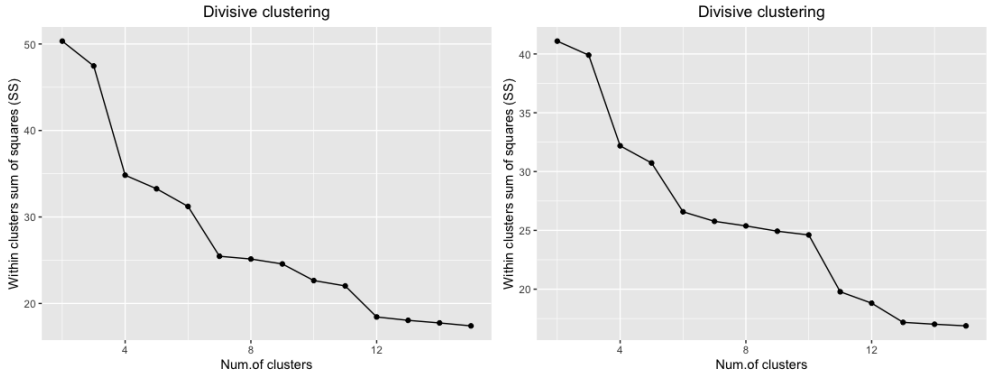

Figure 5: Left-side: Scree plot for agglomerative clustering
Right-side: Scree plot for divisive clustering

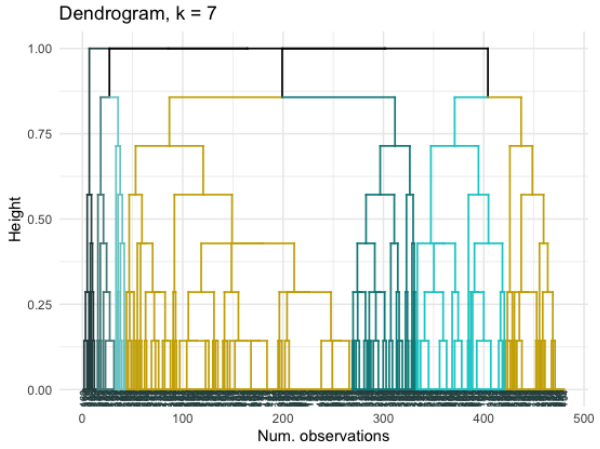

Figure 6: Dendrogram for agglomerative clustering

All in all, we can argue that we can shrink the data to 4 clusters with only 4 features while we still maintain the most of the variance of the original data. These features are gender, marriage

status, income and dependent number. However, one should note that this is a rather subjective result and it is subject to differ depending on the specific research question.

5 Conclusion

In conclusion, our analysis has demonstrated that the market for home loans can be effectively segmented into four distinct clusters and can be explained with only the four features of gender, marriage status, income, and dependent number. By reducing the dimensionality of the data in this way, we are able to capture a large portion of the variance in the original data while still providing meaningful insights into the preferences and needs of different customer segments. However, it is important to note that these results are subject to change depending on the specific research question and the objectives of the analysis. As such, further research and analysis may be necessary to fully understand the nuances of the market for home loans and to develop effective strategies for meeting the needs of different customer segments.

6 Appendix

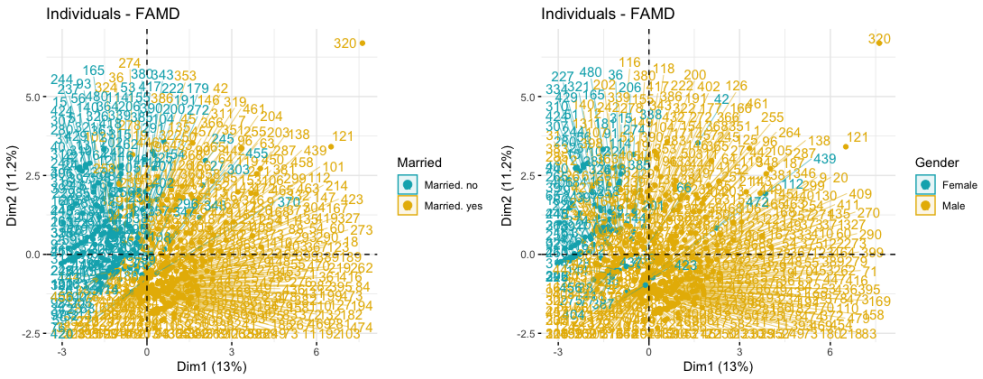

Figure 7: Left-side: graph for individuals with similar profiles in marriage status
Right-side: graph for individuals with similar profiles in gender

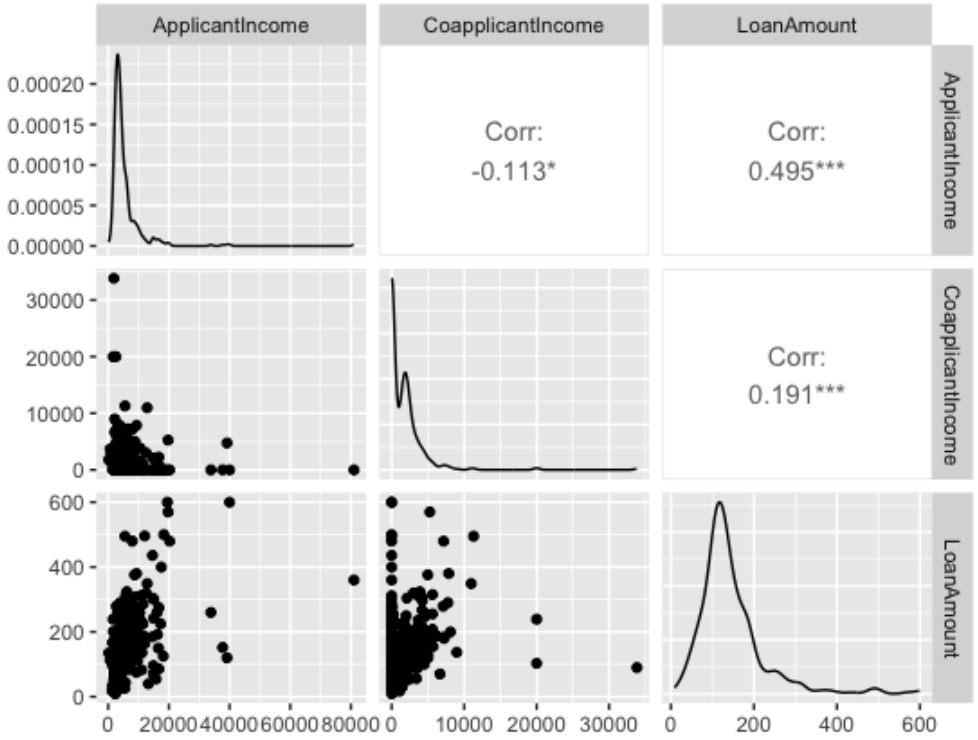

Figure 8: Scatterplot matrix for numeric variables which capture most of the variance in the data as suggested by FAMD analysis

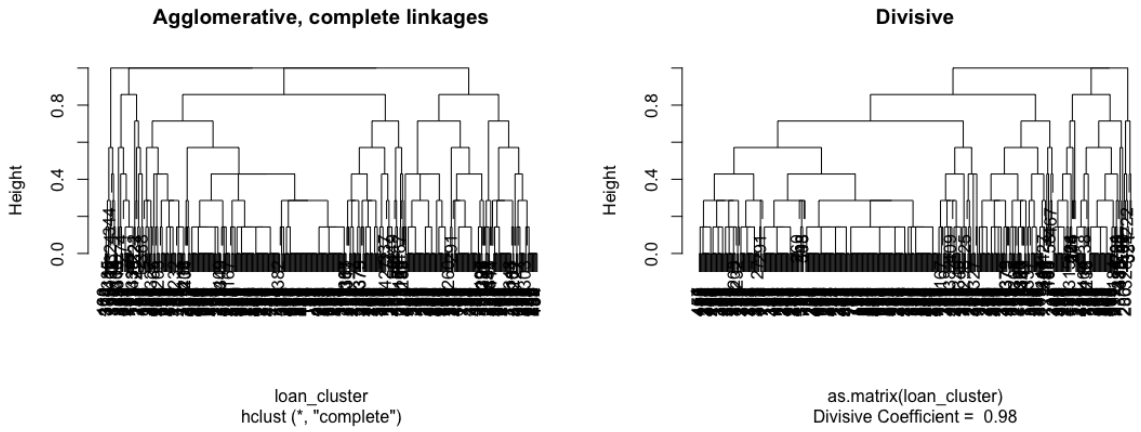

Figure 9: Left-side: Dendrogram for agglomerative clustering with complete linkage
Right-side: Dendrogram for divisive clustering

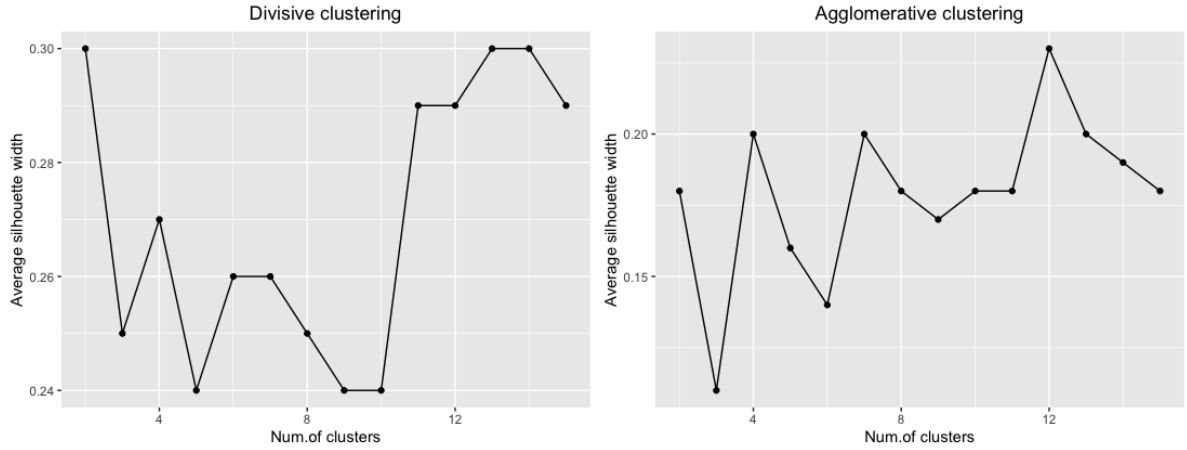

Figure 10: Left-side: Average silhouette width for divisive clustering
Right-side: Average silhouette width for agglomerative clustering

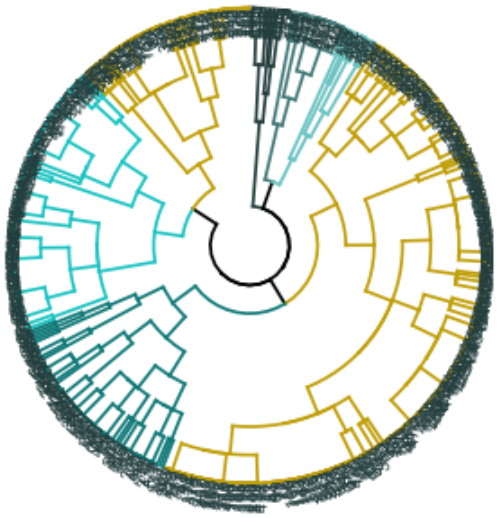

Figure 11: Radial Plot of Dendrogram of Figure 6

References

Analytics Vidhya, A. (2016). Loan prediction dataset.

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate behavioral research 1*(2), 245–276.

Pagès, J. (2004). Analyse factorielle de donnees mixtes: principe et exemple d'application. *Revue de statistique appliquée 52*(4), 93–111.

Saporta, G. (1990). Simultaneous analysis of qualitative and quantitative data. In *Societa Italiana di Statistica. XXXV riunione scientifica*, Volume 1, pp. 62–72. CEDAM.

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association 58*(301), 236–244.

7 Code

7.1 Code

Listing 1: SML code for Final Assignment

```r
### file path etc
#rm(list=ls())
options(scipen=6, digits=4)
##packages anf libraries
if (!require("pacman")) install.packages("pacman")
pacman::p_load(ggplot2, tidyverse, tinytex, rmarkdown, glmnet, matlib, MASS, pdist, PMA,
    softImpute, dplyr, plotrix, kernlab, ranger, randomForest, )
install.packages("FactoMineR")
install.packages("vcd")
install.packages("fpc")
install.packages("factoextra")
install.packages("fastDummies")
install.packages("GGally")
install.packages("cluster")
install.packages("NbClust")
library(GGally)
library(fastDummies)
library(matlib)
library(glmnet, quietly = TRUE)
library(caTools)
library("PMA")
library("softImpute")
library(FactoMineR)
library(vcd)
library(factoextra)
library(cluster)
library(NbClust)
library(readr)

##Data & seed
set.seed(8913)
data <- read_csv("~/Desktop/USML/Loan data for finak/loan_sanction_train.csv")
loan_data<-data[,c(2,3,4,5,6,7,8,9,10,11,12,13)]
View(loan_data)

#Check whether we have omitted variables
summary(loan_data)
sum(complete.cases(loan_data))
loan_data_final<-loan_data[complete.cases(loan_data), ]
str(loan_data_final)
sum(complete.cases(loan_data_final))

#Organising and scaling
loan_data_final$Gender <- as.factor(loan_data_final$Gender)  # Convert character column to
    factor
loan_data_final$Married <- as.factor(loan_data_final$Married)  # Convert character column
    to factor
loan_data_final$Dependents <- as.factor(loan_data_final$Dependents)  # Convert character
    column to factor
loan_data_final$Education <- as.factor(loan_data_final$Education)  # Convert character
    column to factor
loan_data_final$Self_Employed <- as.factor(loan_data_final$Self_Employed)  # Convert
    character column to factor
loan_data_final$Property_Area <- as.factor(loan_data_final$Property_Area)  # Convert
    character column to factor
loan_data_final$Loan_Status <- as.factor(loan_data_final$Loan_Status)  # Convert character
    column to factor

loan_data_final[, 6:10] <- scale(loan_data_final[, 6:10]) #Scale numeric variables
```

```r

loan_data_final$Married<-tolower(loan_data_final$Married) #this is done to prevent an error in the FAMD graphs
loan_data_final$Married<- paste("Married_", loan_data_final$Married)

#PCA data only works with numeric. So we transform categorical variables into dummies
loan_pca_data <- dummy_cols(loan_data_final)
loan_pca_data<-loan_pca_data[,c(6,7,8,9,10,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29)]


#Factor Analysis for Mixed Data
loan_famd <- FAMD(loan_data_final, graph=TRUE)
summary(loan_famd)
fviz_screeplot(loan_famd,choice = "eigenvalue")
fviz_famd_var(loan_famd, repel = TRUE)
fviz_contrib(loan_famd, "var", axes = 1)
fviz_contrib(loan_famd, "var", axes = 2)

#graph of qualitative variables
fviz_famd_var(loan_famd, "quanti.var", repel = TRUE,
              col.var = "black")
fviz_famd_var(loan_famd, "quanti.var", col.var = "contrib",
              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
              repel = TRUE)

#graph of qualitative variables
quali.var <- get_famd_var(loan_famd, "quali.var")
quali.var

fviz_famd_var(loan_famd, "quali.var", col.var = "contrib",
              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07")
)

ind <- get_famd_ind(loan_famd)
ind

fviz_famd_ind(loan_famd, col.ind = "cos2",
              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
              repel = TRUE)

fviz_mfa_ind(loan_famd,
             habillage = "Gender", # color by groups
             palette = c("#00AFBB", "#E7B800", "#FC4E07"),
             addEllipses = TRUE, ellipse.type = "confidence",
             repel = TRUE # Avoid text overlapping
)

fviz_mfa_ind(loan_famd,
             habillage = "Married", # color by groups
             palette = c("#00AFBB", "#E7B800", "#FC4E07"),
             addEllipses = TRUE, ellipse.type = "confidence",
             repel = TRUE # Avoid text overlapping
)


fviz_mfa_ind(loan_famd,
             habillage = "Dependents", # color by groups
             palette = c("#00AFBB", "#E7B800", "#FC4E07","green"),
             addEllipses = TRUE, ellipse.type = "confidence",
             repel = TRUE # Avoid text overlapping
)

#PCA (here we use data we have organized for PCA)
loan_pca <- prcomp(loan_pca_data)
## Scree plot of component variances
```

```r
plot(loan_pca, type = "l")

## Biplot showing the first two components and the variable loadings for these
biplot(loan_pca)

## Redo but know with columns normalized
loan_pca_scaled <- prcomp(loan_pca_data, scale = TRUE)
loan_pca_scaled

## Inspect the results as before
summary(loan_pca_scaled)
plot(loan_pca_scaled, type = "l")
biplot(loan_pca_scaled)
## Result: keep 4 or 5 variables: loan amount, married, dependents, income and maybe gender


####Clustering
ggpairs(loan_data_final,          # Data frame
        columns = c(6,7,8)) # Columns
##loan amount and applicant income are very correlated: should I keep only one of them?

ggparcoord(loan_pca_data, scale = "uniminmax") #Not very readable

# we cannot simply run NbClust on data where we have categorical variables.
#First construct dissimilarity matrix
loan_data_final$Married <- as.factor(loan_data_final$Married)  # this was added to fix an error at the next stage

loan_cluster <- daisy(loan_data_final[ ,c(1,2,3,4,5,11,12)], metric = c("gower"))
class(loan_cluster)

# The main input for the code below is dissimilarity (distance matrix)
# After dissimilarity matrix was calculated, the further steps will be the same for all data types
#------------ DIVISIVE CLUSTERING ------------#
divisive.clust <- diana(as.matrix(loan_cluster),
                        diss = TRUE, keep.diss = TRUE)
plot(divisive.clust, main = "Divisive")

#------------ AGGLOMERATIVE CLUSTERING ------------#
# I am looking for the most balanced approach
# Complete linkages is the approach that best fits this demand
aggl.clust.c <- hclust(loan_cluster, method = "complete")
plot(aggl.clust.c,
     main = "Agglomerative, complete linkages")

# Cluster stats comes out as list while it is more convenient to look at it as a table
# This code below will produce a dataframe with observations in columns and variables in row
# Not quite tidy data, which will require a tweak for plotting, but I prefer this view as an output here as I find it more comprehensive
library(fpc)
cstats.table <- function(dist, tree, k) {
  clust.assess <- c("cluster.number","n","within.cluster.ss","average.within","average.between",
                    "wb.ratio","dunn2","avg.silwidth")
  clust.size <- c("cluster.size")
  stats.names <- c()
  row.clust <- c()
  output.stats <- matrix(ncol = k, nrow = length(clust.assess))
  cluster.sizes <- matrix(ncol = k, nrow = k)
  for(i in c(1:k)){
    row.clust[i] <- paste("Cluster-", i, " size")
  }
  for(i in c(2:k)){
```

```
    stats.names[i] <- paste("Test", i-1)

    for(j in seq_along(clust.assess)){
      output.stats[j, i] <- unlist(cluster.stats(d = dist, clustering = cutree(tree, k = i
    ))[clust.assess])[j]

    }

    for(d in 1:k) {
      cluster.sizes[d, i] <- unlist(cluster.stats(d = dist, clustering = cutree(tree, k =
    i))[clust.size])[d]
      dim(cluster.sizes[d, i]) <- c(length(cluster.sizes[i]), 1)
      cluster.sizes[d, i]

    }
  }
  output.stats.df <- data.frame(output.stats)
  cluster.sizes <- data.frame(cluster.sizes)
  cluster.sizes[is.na(cluster.sizes)] <- 0
  rows.all <- c(clust.assess, row.clust)
  # rownames(output.stats.df) <- clust.assess
  output <- rbind(output.stats.df, cluster.sizes)[ ,-1]
  colnames(output) <- stats.names[2:k]
  rownames(output) <- rows.all
  is.num <- sapply(output, is.numeric)
  output[is.num] <- lapply(output[is.num], round, 2)
  output
}
# I am capping the maximum amout of clusters by 7
# I want to choose a reasonable number, based on which I will be able to see basic
    differences between customer groups as a result
stats.df.divisive <- cstats.table(loan_cluster, divisive.clust, 7)
stats.df.divisive
#for aggloerative clustering
stats.df.aggl <-cstats.table(loan_cluster, aggl.clust.c, 7)
stats.df.aggl

# --------- Choosing the number of clusters ---------#
# Using "Elbow" and "Silhouette" methods to identify the best number of clusters
# to better picture the trend, I will go for more than 7 clusters.
library(ggplot2)
# Elbow
# Divisive clustering
ggplot(data = data.frame(t(cstats.table(loan_cluster, divisive.clust, 15))),
       aes(x=cluster.number, y=within.cluster.ss)) +
  geom_point()+
  geom_line()+
  ggtitle("Divisive clustering") +
  labs(x = "Num.of clusters", y = "Within clusters sum of squares (SS)") +
  theme(plot.title = element_text(hjust = 0.5))
# Agglo clustering
ggplot(data = data.frame(t(cstats.table(loan_cluster, aggl.clust.c, 15))),
       aes(x=cluster.number, y=within.cluster.ss)) +
  geom_point()+
  geom_line()+
  ggtitle("Divisive clustering") +
  labs(x = "Num.of clusters", y = "Within clusters sum of squares (SS)") +
  theme(plot.title = element_text(hjust = 0.5))
# Silhouette
ggplot(data = data.frame(t(cstats.table(loan_cluster, divisive.clust, 15))),
       aes(x=cluster.number, y=avg.silwidth)) +
  geom_point()+
  geom_line()+
  ggtitle("Divisive clustering") +
  labs(x = "Num.of clusters", y = "Average silhouette width") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
#plot
ggplot(data = data.frame(t(cstats.table(loan_cluster, aggl.clust.c, 15))),
       aes(x=cluster.number, y=avg.silwidth)) +
  geom_point()+
  geom_line()+
   ggtitle("Agglomerative clustering") +
  labs(x = "Num.of clusters", y = "Average silhouette width") +
  theme(plot.title = element_text(hjust = 0.5))
#dendrogram
library("ggplot2")
library("reshape2")
library("purrr")
library("dplyr")
# let's start with a dendrogram
library("dendextend")
dendro <- as.dendrogram(aggl.clust.c)
dendro.col <- dendro %>%
  set("branches_k_color", k = 7, value =   c("darkslategray", "darkslategray4", "
     darkslategray3", "gold3", "darkcyan", "cyan3", "gold3")) %>%
  set("branches_lwd", 0.6) %>%
  set("labels_colors",
      value = c("darkslategray")) %>%
  set("labels_cex", 0.5)
ggd1 <- as.ggdend(dendro.col)
ggplot(ggd1, theme = theme_minimal()) +
  labs(x = "Num. observations", y = "Height", title = "Dendrogram, k = 7")
# Radial plot looks less cluttered (and cooler)
ggplot(ggd1, labels = T) +
  scale_y_reverse(expand = c(0.2, 0)) +
  coord_polar(theta="x")

###Do the same clustering on the reduced data
reduced_data<-loan_data_final[,c(1,2,3,6,8)]

reduced_cluster <- daisy(reduced_data[ ,c(1,2,3,4,5)], metric = c("gower"))
class(reduced_cluster)
#same divisive and agglo. clusterings
divisive.clust2 <- diana(as.matrix(reduced_cluster),
                         diss = TRUE, keep.diss = TRUE)
plot(divisive.clust2, main = "Divisive")

aggl.clust.c2 <- hclust(reduced_cluster, method = "complete")
plot(aggl.clust.c,
     main = "Agglomerative, complete linkages")
# I am capping the maximum amout of clusters by 7
# I want to choose a reasonable number, based on which I will be able to see basic
    differences between customer groups as a result
stats.df.divisive2 <- cstats.table(reduced_cluster, divisive.clust2, 7)
stats.df.divisive
#for aggloerative clustering
stats.df.aggl2 <-cstats.table(reduced_cluster, aggl.clust.c2, 7)
stats.df.aggl

# Elbow
# Divisive clustering
ggplot(data = data.frame(t(cstats.table(reduced_cluster, divisive.clust2, 15))),
       aes(x=cluster.number, y=within.cluster.ss)) +
  geom_point()+
  geom_line()+
   ggtitle("Divisive clustering") +
  labs(x = "Num.of clusters", y = "Within clusters sum of squares (SS)") +
  theme(plot.title = element_text(hjust = 0.5))
# Agglo clustering
ggplot(data = data.frame(t(cstats.table(reduced_cluster, aggl.clust.c2, 15))),
       aes(x=cluster.number, y=within.cluster.ss)) +
  geom_point()+
```

```
  geom_line()+
  ggtitle("Divisive clustering") +
  labs(x = "Num.of clusters", y = "Within clusters sum of squares (SS)") +
  theme(plot.title = element_text(hjust = 0.5))
# Silhouette
ggplot(data = data.frame(t(cstats.table(reduced_cluster, divisive.clust2, 15))),
       aes(x=cluster.number, y=avg.silwidth)) +
  geom_point()+
  geom_line()+
  ggtitle("Divisive clustering") +
  labs(x = "Num.of clusters", y = "Average silhouette width") +
  theme(plot.title = element_text(hjust = 0.5))
#plot
ggplot(data = data.frame(t(cstats.table(reduced_cluster, aggl.clust.c2, 15))),
       aes(x=cluster.number, y=avg.silwidth)) +
  geom_point()+
  geom_line()+
  ggtitle("Agglomerative clustering") +
  labs(x = "Num.of clusters", y = "Average silhouette width") +
  theme(plot.title = element_text(hjust = 0.5))
```