Republique Algérienne démocratique et populaire.
Ministére de l'ensignement superieur et de la recherche scientifique

Ecole Nationale Superieur Se Statistique Et d'economie Appliqué

Under the Direction of : BREDGITAL

Classification of households according to their possessions.

Clustering problem

PRESENTED BY :

Bouras Mohamed Youcef

Lamri Mohamed Achraf

Gouaref Hamza

Tekouk Mohamed El-Hilal

Bensadoun Anas AbdelMoudjib

6 mai 2023

Table des figures

Liste des tableaux

Chapitre 1

Introduction

1.1 Introduction

In recent years, machine learning and data analytics have become increasingly important tools for solving complex problems in various domains. One such domain is data science, where techniques such as classification, regression, and clustering are used to extract insights from large datasets. In this report, we focus on the Clustering problem in the context of a datathon that we participated in.

The datathon presented a Clustering problem that involved predicting a binary outcome variable based on a set of features. Specifically, we were given a dataset of historical observations and tasked with predicting whether future observations would result in the outcome of interest or not. The problem was challenging due to the large number of features and the imbalanced nature of the dataset.

In this report, we describe the approach we took to tackle the Clustering problem and the results we obtained. We also discuss the limitations of our approach and suggest future work that could be done to improve the Clustering performance. Overall, our goal is to contribute to the growing body of research on Clustering problems and to demonstrate the value of machine learning techniques in solving real-world problems.

1.2 Definition of the problem

The problem of "Classification of households according to their possessions" involves using a set of criteria to categorize households based on the possessions and assets they own. This can include items such as TVs, computers, cars, and other consumer goods. The goal is to create a system of classification that can be used to better understand the characteristics of different types of households and their socioeconomic status, as well as to inform policies and interventions aimed at improving the standard of living for households with fewer possessions. The classification system can also be used for market research, consumer profiling, and other related purposes.

Chapitre 2

Presentation of the Data

2.1 Data presentation

The dataset was collected from a survey of over 28,000 Algerian households, which is representative of the Algerian population as per the RGPG 2008. The survey collected information on two main aspects - the income levels of the households and the possessions they have. For the possessions, the surveyors considered a set of criteria, which includes various electronic devices such as TV, oven, washing machine, computer, freezer, air conditioner, game console, dishwasher, dryer, set-top box, laptop, ADSL, tablet, smartphone, and the number of cars and connections. Overall, the dataset provides a comprehensive overview of the possessions and income levels of Algerian households and can be used for various analytical and modeling purposes. WILAYA : the region where the household is located
SEXE : gender of the head of the household
AGE1 : age of the head of the household
$Q7_A4C1$: possession of a television
$Q7_A4C2$: possession of two televisions
$Q7_A4C3$: possession of three or more televisions
$Q7TV$: possession of at least one television
$Q7_A4C4$: possession of an electric oven and microwave
$Q7_A4C5$: possession of a washing machine
$Q7_A4C6$: possession of a desktop computer
$Q7_A4C7$: possession of a freezer
$Q7_A4C8$: possession of an air conditioner
$Q7_A4C9$: possession of a video game console (SD, PlayStation, Wii)
$Q7_A4C10$: possession of a dishwasher
$Q7_A4C11$: possession of a clothes dryer machine
$Q7_A4C12$: possession of a simple set-top box
$Q7_A4C13$: possession of a digital set-top box with decoder
$Q7_A4C14$: possession of a set-top box with internet update
$Q7DEMO$: possession of at least one set-top box
$Q7_A4C15$: possession of a laptop or portable computer
$Q7_A4C16$: possession of an ADSL internet connection
$Q7_A4C17$: possession of a tablet
$Q7_A4C18$: possession of a smartphone
$Q7_A4C19$: possession of a 3G internet connection
$Q7_A4C20$: possession of a 4G internet connection

$Q7CONX$: possession of at least one internet connection
$Q7_A5$: possession of at least one car by the household
$Q7_A6$: global income of the household.

2.2 Importing the dataset

```
# Import the neccessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

We can import the dataset using this command

```
data = pd.read_excel("Desktop/DataThon/PROJECT/File For Challenge Open Data.xlsx")
data.head()
```

ımed: 0	id	wilaya	sexe	age1	televiseur	x2_televiseurs	x3_televiseurs_et_plus	au_moins_un_televiseur	four_electrique_micro_onde	...	lap_top_pc_portable
1	1	Adrar	Femme	45-54	O	N	N	O	N	...	O
2	11	Adrar	Femme	35-44	N	O	N	O	N	...	N
3	16	Adrar	Femme	25-34	N	N	O	O	O	...	N
4	20	Adrar	Homme	15-24	N	N	O	O	O	...	O
5	34	Adrar	Homme	55+	N	N	O	O	O	...	O

31 columns

FIGURE 2.1 – Data set

This photo represent the head of the data set which contain 31 columns "features" and 28000 Observations.

Chapitre 3

Data prepossessing

3.1 data cleaning

Data cleaning is the process of identifying, correcting, and removing errors, inconsistencies, and inaccuracies in the collected data.

3.1.1 Duplicates

```
# Detect duplicates
duplicate_mask = data.duplicated()
duplicate_count = duplicate_mask.sum()
duplicate_count
```

There is no duplicates in the dataset.

3.1.2 Missing Values

Missing Values are values that are not present in a data set where values are expected to be.These can be caused by a variety of factors such as data entry errors, sensor malfunctions, or non-response in surveys.it is important to handle missing values appropriately in data analysis by applying the optimum methods. in our case after examining the counts of the modalities of each variables we concluded that the all of them are coded as character "NSP" which adds no value to our study and they appear in only one variable which is the family range of income "revenue des menages gloubales".these missing values represents 20% of the totale counts, meaning that dropping the observations with missing values will cause a loss of 20 % of the data so it's better to find a way to impute them

```
# Detection:
data["Revenu_global_du_ménage"].value_counts()/ len(data)
```

3.1.3 Imputation Using Decision Tree

There are several methods that can be used for imputing missing values in categorical data like : mode imputation that suggests filling the missing data with the modalitie with the highest number of counts . but this methode is not suited for this data because some modalities are too close in counts like : , and this would mess the distribution of the variable targeted. A better method would be to

impute with Decision tree, This involves using a regression model to predict the missing value based on the observed values in the other categories. This method can capture the relationships between variables and produce accurate imputations.

```
without_nsp<-data %>%
  filter(!revenu_global_du_menage=="NSP") %>%
  mutate(new=as.factor(new))
with_nsp<-data %>%
  filter(revenu_global_du_menage=="NSP")%>%
  mutate(new=as.character(new))
#########################################################
train<-without_nsp
test<-with_nsp
#########################################################
linear_reg<-decision_tree() %>% set_engine("rpart") %>% set_mode("classification")
#########################################
recipe<-train %>%
  recipe(new~.) %>%
  step_rm(id,wilaya,sexe,age1,revenu_global_du_menage) %>%
  step_dummy(all_nominal_predictors())
#########################################################
wkfl<-workflow() %>%
  add_recipe(recipe) %>%
  add_model(linear_reg)
#########################################################
fit<-wkfl %>% fit(train)
pred<-fit %>% predict(test)
#########################################################
data<-with_nsp %>% bind_cols(pred) %>%
  mutate(new=NULL) %>%
  rename(new=".pred_class") %>%
  bind_rows(without_nsp)
```

The accuracy of 36 percent for a decision tree indicates that only 36 percent of the predictions made by the model are correct. This means that the model is not performing well in classifying the data correctly, and it preserved 36 percent of the data without shifting the counts of the variable of income, and 67% roc_auc indicate that area under curve means that the model has a moderate ability to distinguish between the positive and negative classes. The value ranges from 0 to 1, with a higher score indicating a better performance of the model, It may not be the perfect model for predicting but it is one of the most convenient methods for imputing missing data.

3.2 Feature Engineering

Feature engineering is the process of selecting and transforming the most relevant features (input variables) in a dataset to improve the performance of a machine learning model.

```
# Define the coefficients for each variable
coefficients = {
    "Téléviseur": 1,
    "2_Téléviseurs": 2,
    "3_Téléviseurs_et_plus": 3,
    "Console_de_jeux_vidéo_Play_Station_Wii": 5,
    "Lap_top / PC_Portable": 4,
    "Ordinateur_de_bureau": 3,
    "Tablettes": 3,
```

```
    "Smartphone": 2
}
# Create a new column called "Entertainment_Score" and assign a score to each row
data["Entertainment_Score"] = 0
for var, coef in coefficients.items():
    data.loc[(data[var] == "O"), "Entertainment_Score"] += coef
```

This code is creating a new column in the dataset called "$Entertainment_Score$" and assigning a score to each row based on the possession of certain entertainment-related items.
The coefficients dictionary contains the coefficients (or weights) assigned to each possession item. For example, owning a single "Téléviseur" (TV) has a coefficient of 1, owning "$2_Tlviseurs$" (2 TVs) has a coefficient of 2, and so on.
The for loop iterates through each item in the coefficients dictionary, and for each item it adds the coefficient to the "$Entertainment_Score$" column for any row where the possession of that item is indicated with an "O".
So for example, if a household has a TV and a laptop, the "$Entertainment_Score$" for that household would be calculated as follows :
The possession of "Téléviseur" is indicated with an "O", so 1 is added to the "$Entertainment_Score$" The possession of "$Ordinateur_de_bureau$" (Desktop computer) is not indicated with an "O", so 0 is added to the "$Entertainment_Score$" The possession of "Lap_top / $PC_Portable$" (Laptop) is indicated with an "O", so 4 is added to the "$Entertainment_Score$" The total "$Entertainment_Score$" for this household would be 1 + 0 + 4 = 5. We use this method to summarise the variables that are similar enough to reduce the dimensions of the data frame to be more manageable to explain and calculate considering the conditions of the K-mode algorithm.

```
# Define the coefficients for each variable
coefficients = {
    "Four électrique_&_micro-onde": 4,
    "Lave-linge": 3,
    "Congélateur": 3,
    "Climatiseur": 5,
    "Lave-vaisselle": 4,
    "Machine_Sèche-linge": 3,
    "Démodulateurs_simples": 1,
    "Démodulateurs_numériques_avec_décodeurs": 2,
    "Démodulateurs / Mise_à_jour_sur_Internet": 3
}
# Create a new column called "Electromenager_Score" and assign a score to each row
data["Electromenager_Score"] = 0
for var, coef in coefficients.items():
    data.loc[(data[var] == "O"), "Electromenager_Score"] += coef
```

This code is creating a new column called "$Electromenager_Score$" in the dataframe "data" and assigning a score to each row based on whether certain variables have a value of "O". The variables and corresponding coefficients are defined in the dictionary "coefficients", with the key being the name of the variable and the value being the corresponding coefficient. The for loop iterates through each key-value pair in "coefficients". For each pair, it checks if the value of the variable (specified by the key) in a given row of the dataframe is "O". If it is, it adds the corresponding coefficient (specified by the value) to the "$Electromenager_Score$" for that row. Overall, this code is calculating a score for each row in the dataframe based on the presence or absence of certain household appliances, with each appliance being assigned a different weight (specified by the corresponding coefficient) in the calculation of the score.

```
# Define the coefficients for each variable
coefficients = {
    "Connexion_Internet_ADSL": 2,
    "Connexion_Internet_3G": 1,
    "Connexion_Internet_4G": 3,
}
# Create a new column called "Telecom_Score" and assign a score to each row
data["Telecom_Score"] = 0
for var, coef in coefficients.items():
    data.loc[(data[var] == "O"), "Telecom_Score"] += coef
```

Unnamed: 0	id	wilaya	sexe	age1	televiseur	x2_televiseurs	x3_televiseurs_et_plus	au_moins_un_televiseur	four_electrique_micro_onde	...	lap_top_pc_portable
1	1	Adrar	Femme	45-54	O	N	N	O	N	...	O
2	11	Adrar	Femme	35-44	N	O	N	O	N	...	N
3	16	Adrar	Femme	25-34	N	N	O	O	O	...	N
4	20	Adrar	Homme	15-24	N	N	O	O	O	...	O
5	34	Adrar	Homme	55+	N	N	O	O	O	...	O

31 columns

FIGURE 3.1 – The new dataset

```
data<-data %>%
  mutate(new= case_when(revenu_global_du_menage=="Entre 20.000 DA et 40.000 DA" ~ "30000",
                        revenu_global_du_menage=="Entre 40.000 DA et 60.000 DA" ~ "50000",
                        revenu_global_du_menage=="NSP" ~"0",
                        revenu_global_du_menage=="Moins de 20.000 DA" ~ "20000",
                        revenu_global_du_menage=="Entre 60.000 DA et 80.000 DA" ~ "70000",
                        revenu_global_du_menage=="Entre 80000 DA et 100.000 DA" ~ "90000",
                        revenu_global_du_menage=="Entre 100000 DA et 150.000 DA" ~ "125000",
                        revenu_global_du_menage=="Entre 150.000 DA et 300.000 DA"~ "225000",
                        revenu_global_du_menage=="Plus 300.001DA" ~ "300000",
                        TRUE ~ "na"))
```

```
revenu_map = {
     20000: 1,
     30000: 2,
     50000: 3,
     70000: 3,
     90000: 4,
     125000: 5,
     225000: 6,
     300000: 7,

}
# Apply the mapping to the categorical variable
data['revenue_numeric'] = data['Revenu_global_du_ménage'].map(revenue_map)
```

This code creates a new column called "revenue_numeric" in the dataframe "data", which maps the categorical variable "Revenu_global_du_ménage" to a numerical scale based on the "revenue_map"

dictionary.
The "revenue_map" dictionary assigns a numerical value to each category of the "Revenu_global_du_ménage" variable. For example, a household with a revenue of 20,000 will be assigned a value of 1, and a household with a revenue of 30,000 will be assigned a value of 2.

OUR DATASET IS READY FOR EXPLORING

Chapitre 4

Exploratory data analysis

4.1 Univariete Analysis

Univariate analysis is a statistical analysis technique that focuses on analyzing one variable at a time.

```
data %>%
 ggplot(aes(x = fct_rev(fct_infreq(as.factor(Telecom_Score)))))+
 stat_count(geom = "bar",fill="#3C565B",col="black") +
 coord_flip()+
 labs(title = "Telecom Score Frequency",x="Telecome_score",y="frequency")
```

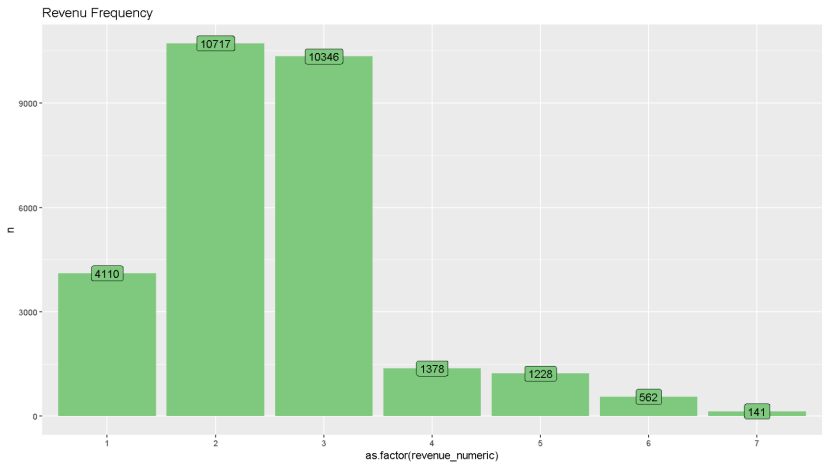

FIGURE 4.1 – Outcome

This Histogramme represente the number of families in each outcome classes, We can see that most of families observed has a globale income that ranges between 20000 DA to 40000 DA.

```
data %>%
group_by(revenue_numeric) %>%
  count() %>%
  ggplot(aes(x=as.factor(revenue_numeric),y=n,fill="#3C565B"))+
  geom_col(position = "dodge") +
  scale_fill_brewer(palette = "Accent") +
  geom_label(aes(label = n))+
  labs(title = "Revenu Frequency")
```

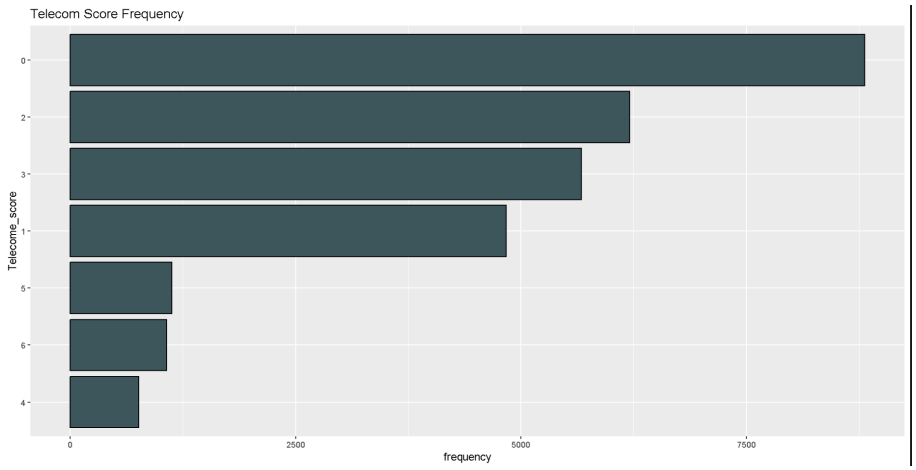

FIGURE 4.2 – *Telecome$_s$core*

The 4g is the most telecommunication methods used in 2018.

```
data %>%
group_by(possession_dune_voiture_menage) %>%
summarize(n =n()) %>%
arrange(-n) %>%
ggplot(aes(y = n, x= "", fill = possession_dune_voiture_menage)) +
geom_bar(width = 1,stat = "identity") +
coord_polar("y", start = 0) +
geom_text(aes(label = n),
position = position_stack(vjust = 0.5))
```

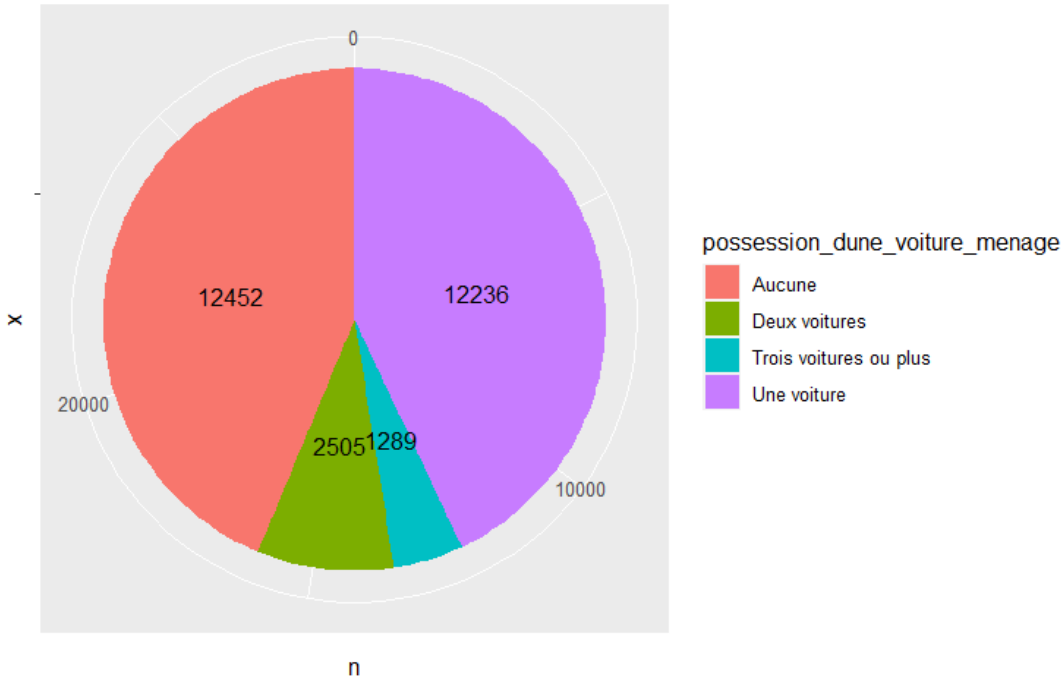

FIGURE 4.3 – Pie Chart

12452 family of our data set don't have any car, and 12236 have just one car, just 1289 of the families has 3 or more cars, which mean that the most of them don't have or just have one.

4.2 Bivariete Analysis

Bivariate analysis is a statistical analysis technique used to examine the relationship between two variables. It helps to identify patterns and relationships between variables, and to determine if there is a significant association or correlation between them. The two variables can be either continuous, categorical or a combination of both.

```
data %>%
  group_by(wilaya) %>%
  summarise(mean=mean(as.integer(new))) %>%
  arrange(-mean) %>%
  head() %>%
  ggplot(aes(x = wilaya,y=mean))+
  geom_col(fill="#95B9C7")+
  labs(title = "Highest Average Salary per Wilaya",x="wilaya",y="average salary")
```

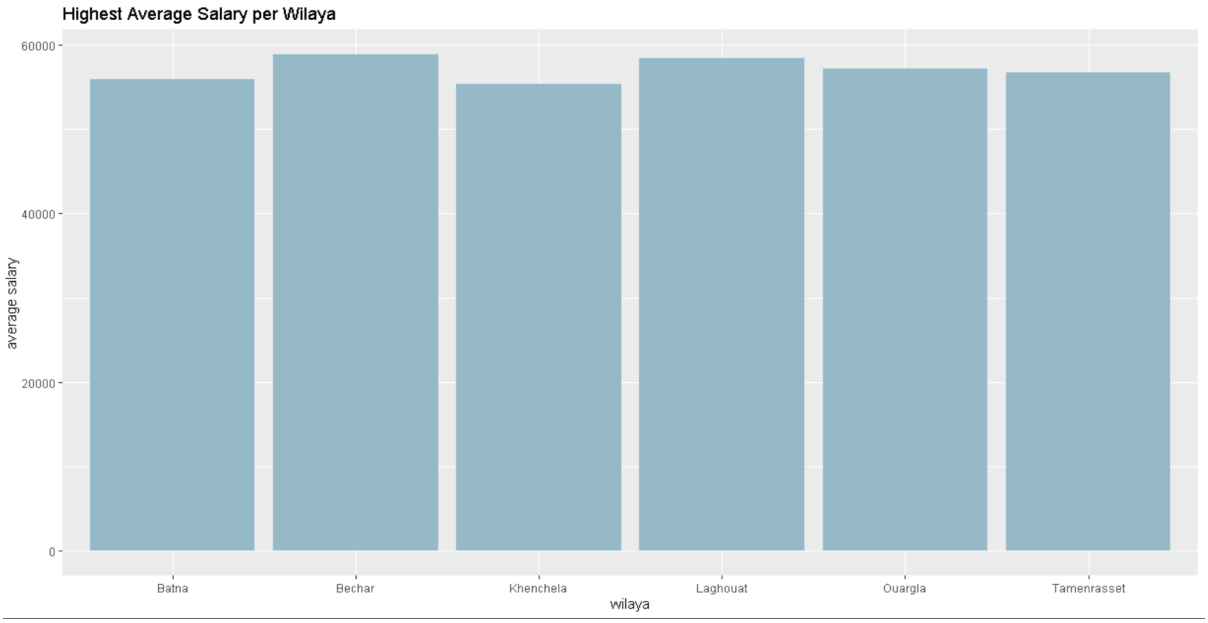

FIGURE 4.4 – Caption

This figure represent the highest average salary per Wilaya : We can see that the Wilayas : Batna Bechar Ouragla Leghouat Tamenrasset Khanchela has the most highest salary in the country.This result seems illogical for some wilayas like Leghouat it impossible t have an average salary high than the capital or the largest wilayas like constantine, Oran ...

```
data %>%
   group_by(new.possession_dune_voiture_menage)%>%
   count()
```

new <chr>	possession_dune_voiture_menage <int>	n <int>
125000	0	152
125000	1	640
125000	2	290
125000	3	146
20000	0	2875
20000	1	1097
20000	2	99
20000	3	39
225000	0	43
225000	1	192
225000	2	185
225000	3	142
30000	0	6349
30000	1	4096
30000	2	515
30000	3	218
300000	0	12
300000	1	29
300000	2	24
300000	3	76
50000	0	2107
50000	1	4045
50000	2	847
50000	3	420
70000	0	677
70000	1	1346
70000	2	304
70000	3	139

FIGURE 4.5 – Data represente the relationship between the salary and number of cars

In this dataFrame we grouped the classes of salary with number of cars and we have found some illogical results : like some one who has the lowest salary but he has 3 cars or more, also people that have the the highest salary and don't have any cars, So we suggest that it can be a mistake in the survey or wrong informations gived buy peoples.

```
c <- cor(data)
COR <- as.data.Frame(matrix(c,ncol = 5, byrow = FALSE)
col <- ColorRampPalette(c("white"), "darked"))(200)
heatmap(x= c, col = col, symm = TRUE)
```

This heat map presents the correlation between the different variables on a colour pallet from white to red, in our case we can see that there is a decent correlation that is distributed equally between all the variables.

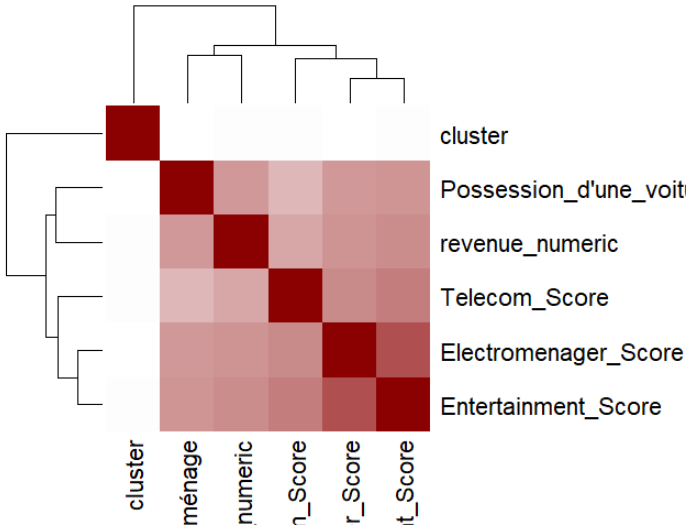

FIGURE 4.6 – Heatmap

Chapitre 5

Clustering

Clustering is a technique in unsupervised machine learning that involves grouping a set of objects in a way that objects in the same group (called a cluster) are more similar to each other than to those in other groups (clusters). The goal of clustering is to identify patterns and structures in data that are not immediately obvious, such as natural groupings or associations between variables.
There are various methods of clustering, including :
K-means clustering : A method that partitions a dataset into k distinct clusters based on the similarity of the data points to the mean of each cluster.
Hierarchical clustering : A method that creates a tree-like structure of clusters by recursively dividing or merging them based on the similarity of their objects.
DBSCAN (Density-Based Spatial Clustering of Applications with Noise) : A method that groups together data points that are closely packed together (dense regions), separated by regions of lower point density.
Mean-shift clustering : A method that identifies the modes (peaks) of a density function and assigns each data point to the closest mode.
Spectral clustering : A method that uses the spectral properties of a similarity matrix to group data points together.
Fuzzy clustering : A method that allows objects to belong to multiple clusters with varying degrees of membership, based on the degree of similarity to the centers of the clusters.

5.1 KModes

KModes is a clustering algorithm that extends the traditional k-means algorithm for categorical data. It is a partitional clustering method that aims to partition the dataset into k clusters, where k is a user-defined parameter. It works by iteratively assigning each observation to the cluster whose centroid is closest to it in terms of a dissimilarity measure.
The KModes algorithm, unlike KMeans, operates with modes instead of means. In KModes, a mode is the most frequently occurring value in a cluster for each categorical feature. The mode of each cluster is used as its centroid, and the objective is to minimize the dissimilarity between the observations in a cluster and its mode. Dissimilarity in KModes is measured by counting the number of features for which two observations differ.
The KModes algorithm is particularly useful for datasets that contain only categorical features. It is computationally efficient and can handle large datasets. KModes can also handle missing data by assigning missing values to the mode of the cluster. However, like KMeans, it requires the user to specify the number of clusters k beforehand.

5.2 Steps of KModes

Initialize the number of clusters k.
Randomly select k initial centroids from the dataset.
Assign each data point to the nearest centroid using a distance metric, which is based on the categorical attributes of the data.
Update the centroids for each cluster by choosing the mode (most common value) for each categorical attribute for the data points in that cluster.
Repeat steps 3 and 4 until the centroids no longer change or a maximum number of iterations is reached.
The final result is k clusters, where each cluster contains the data points with similar categorical attributes.

```
import pandas as pd
import numpy as np
from kmodes.kmodes import KModes
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
X = data_for_clustring.values
range_n_clusters = range(2, 6)
sse = []
for n_clusters in range_n_clusters:
    kmodes = KModes(n_clusters=n_clusters, init='Cao', n_init=10, verbose=0)
    kmodes.fit(X)
    sse.append(kmodes.cost_)
plt.plot(range_n_clusters, sse)
plt.xlabel("Number of clusters")
plt.ylabel("Sum of squared distances")
plt.show()
```

we choosed to use init='Cao' the method proposed by Cao et al. (2009) to select initial centroids based on both the frequency and density of categorical values because of the high dimensionality of the data. According to the plot we choose the value of k of k-modes which equal to 3. Fitting the model

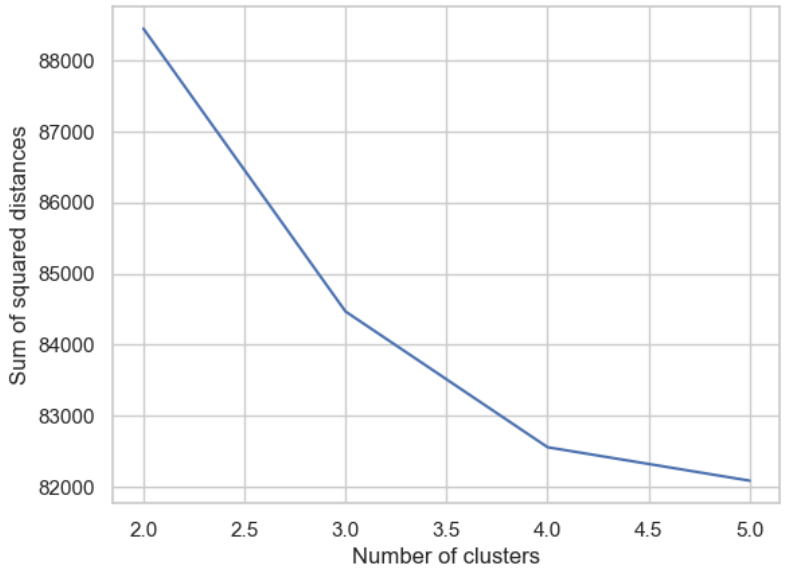

FIGURE 5.1 – Elbow plot

using k=3 to the data and adding the cluster variable to the dataframe.

```
k = 3
kmodes = KModes(n_clusters=k, init='Cao', n_init=10, verbose=0)
kmodes.fit(X)
labels = kmodes.labels_
data_for_clustring['cluster'] = labels
```

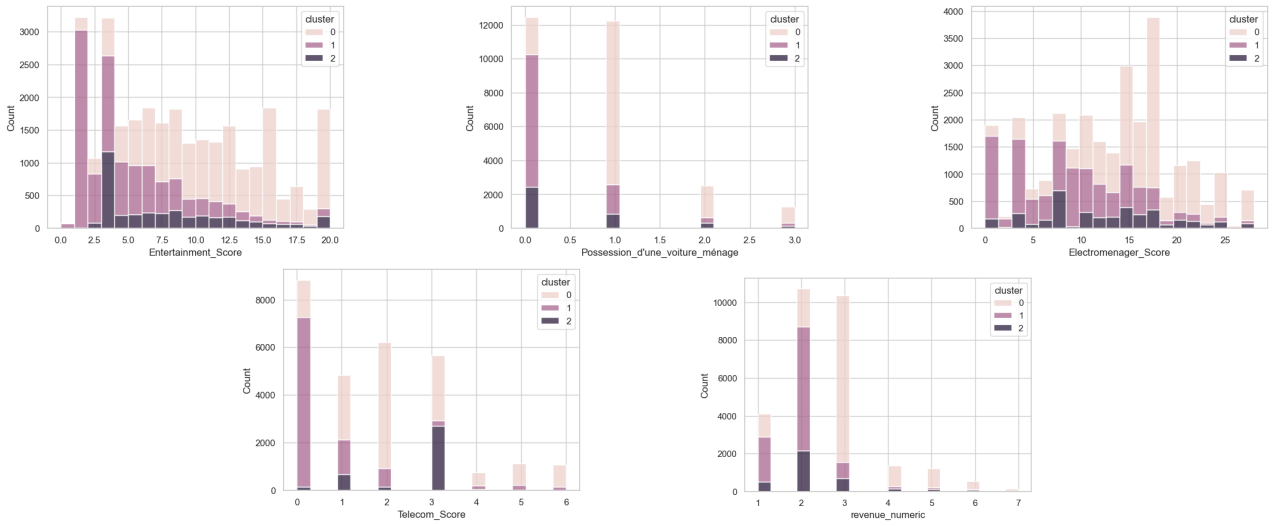

FIGURE 5.2 – Clustering graphs

graph1 "Entertainment" : in the entertainment score we can see that the cluster 0 is centred relatively in the middle but it represents the majority of families with full entertainment score, while the cluster 1 is mostly shifted in lower entertainment score peaking in the lowest score the cluster 2 however has the lowest counts compared to the other two groups while being distributed uniformly on scale grid. graph1"car" :
most of families in the cluster 0 has only one car while the most of the cluster 1 and 2 counts has no car at all.
graph3"home appliances"(Electromenagé) :
the cluster 0 is centered in average score of home appliance while cluster 1 being centered in lower values of it,the cluster 2 is distributed all over the grid with a peak in relatively low score.

graph4"Telecom score" :
the cluster 0 still shows a accumulation in average value of the grid while the cluster 1 is peaking in the lowest value of the telecom score , but the cluster 2 having most of its families in slightly high score.

graph5 "income"(revenue) :
the cluster 0 is peaking on middle value of income but it's the only one with the highest income, the other two clusters is peaking in low incomes.

in general the cluster 0 and 1 represent the highest concentration of individuals . We can clearly see that cluster 0 is almost always giving an average score of quality of life and its mostly focused on entertainment and quality of internet so we can conclude that it represents the middle class of the Algerian society .on the other hand the cluster 1 is always displaying lower scores compared to the

the other clusters focusing its consumption on home appliances makes it the lower class of society . the cluster 2 however is hard to define because it lacks enough counts of families but it's mostly brunched from the low class cluster because it has a lot of it's attributes.

The Cluster 0 mostly refers to the people that have just one car or they don't have any, and for electromenager and revenu the score is in the middle the m

Chapitre 6

Conclusion

Our analysis began with a categorical dataset with missing values. We imputed these missing values using decision tree predictions, allowing us to move forward with our analysis. We then created a score grid for each life quality indicator, including entertainment, salary, telecommunications, and home appliances, to better understand the relationships between these variables.

We explored the data using both univariate and bivariate analysis techniques, and found that there were strong correlations between certain variables. This led us to perform clustering analysis using the kmodes method, which helped us identify three distinct clusters within the data.

However, while kmodes clustering can be useful for categorical data, it has some limitations. One such limitation is that it is sensitive to the initialization of the algorithm, meaning that different starting points can lead to different cluster assignments. Additionally, it may struggle with datasets that have high cardinality, meaning that the number of distinct values for each feature is large.

To address these limitations, we propose a new solution : treating the cluster assignment as a target variable and using supervised learning techniques to predict the cluster of new individuals based on information about them. We suggest using a Random Forest classifier, which has been shown to be effective in handling categorical data and can handle high-dimensional feature spaces.

In summary, our analysis has provided valuable insights into the relationships between various life quality indicators and has identified distinct clusters within our dataset. Moving forward, we can use these clusters to better understand and target different segments of our population. Additionally, by treating the cluster assignment as a target variable, we can continue to make use of this clustering in a supervised learning context, allowing us to make predictions about new individuals based on the features that we have available.

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

# Define the input variables without the "cluster" column
X = data.drop(columns=["cluster"])

# Define the target variable as the "cluster" column
y = data["cluster"]

# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a random forest model
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Use the model to predict the target variable for the test set
```

```
y_pred = rf_model.predict(X_test)

# Calculate the accuracy of the model
accuracy = rf_model.score(X_test, y_test)
print("Accuracy:", accuracy)
```

A 99% accuracy score for random forest model is an impressive result. It suggests that the variables of our dataset used to train the model are strongly related to the cluster column.