King Saud University

College of Computer and Information Sciences

Information Technology department

IT 326: Data Mining

Course Project

Case Study: Applicants for a Gold Digger position

Project Report: Data Summarization and Preprocessing

Group#:	6	
Section#:	64063	
Group Members	Name	ID
	Seba alahmadi	439201336
	Noura alsultan	441201306
	Nouf alfulaij	441201159
	Reem Almutairi	437201303

10/21/21

* كيف حلينا مشكلتنا بالكلاسفكيشن والكلسترنق ؟

مشكلتنا كانت ان وظيفة ال قولد ديقر وظيفه حساسه ومو اي احد يتوظف فيها ف الكلاسيفكيشن توقع لنا الموظفين الجداد اذا بيتوظفون او لا بيسد أون تراينق سيت ، والكلسترنق خلانا نلاقي السيميلاريتي بين كل كلستر او قروب عشان نعرف ونميز بين القروبات وايش كل قروب يتميز عن الثاني

مشكلتنا ؟

1 Problem

The gold-digger job is an important and sensitive job that needs people with high skills and specialists in this field. In this report, data mining techniques are used to predict whether a candidate will be hired or not based on set of information of people who applied for gold digger position. It is important because it will show summary of candidates that will help and serve people who are interested in hiring people with gold digger experience.

2 Data Mining Task

ايش راح يفيدنا الكلسترنق في الداتا سيت حقتنا ؟

The data mining tasks are clustering and classification.

In clustering we will analyze the Applicants for a Gold Digger position dataset, find the similarity to identify distinct groups and what are the factors that effects and distinguishes one applicant from another.

In classification, it is a binary classification problem where the class attribute is the **embauche**(**hiring**) which has two values: 1 which indicates that the applicant has been hired and 0 which indicates that the applicant has not been hired.

ايش يفيدنا الكلاسفكيشن في الداتا ست حقتنا ؟

The goal of the data mining task is to classify an applicant, based on years of relevant experience, salary expectation, highest qualification diploma, specialties, note(grade (out of 100) for gold digging exam) into either a candidate will be hired or not.

Page 26

بداية Phase 2

3 Data

We choose this data from Kaggle website https://www.kaggle.com/bryanb/applicants-for-a-gold-digger-position

The dataset as shown in *figure1* includes Applicants for a Gold Digger position of 20000 applicants(object) between 1/1/2010 and 30/12/2014 it consists of the following 12 attributes as shown in *figure2* with 396 missing values but we minimize the data to 999 rows as shown in *figure3* with 27 missing values as shown in *figure5*. The attributes names, description, data type, and possible values of features are given in *Table 1*.

Data set rows before

```
> #print NO of objects
> nrow(data)
[1] 20000
```

Figure1 number of rows

Data set rows after

```
> #print No of objects
> nrow(data)
[1] 999
> # print NUM of features
```

Figure3 number of rows minimized

```
> # print NUM of features
> ncol(data)
[1] 12
>
```

Figure2 number of columns

```
> #print names of the features
> names(data)
 [1] "X"          "date"       "cheveux"    "age"        "exp"
 [6] "salaire"    "sexe"       "diplome"    "specialite" "note"
[11] "dispo"      "embauche"
>
```

Figure4 names of features

Attributes name	description	Data type	possible values
x	Index of applicant	Numeric	Range between 1 - 999
date	date of the application	nominal	Range between 1/1/2010 – 30/12/2014
cheveux	hair color	nominal	(chatain,brun,blond,roux)
age	age of the candidate	numeric	Range between [-3…..74]
exp	Years of relevant experience	numeric	Range between [-2…..23]
salaire	Salary expectation	numeric	Range between [14.1k…54k]
sexe	Female or male	binary	F :Female OR M: Male
diplome	Highest qualification diploma	nominal	(bac, licence, master, doctorat)
specialite	minor of the diploma	nominal	(geologie, forage, detective, archeologie)
note	Grade out of 100 for gold digging exam	numeric	Range between [0…100]
dispo	directly available	binary	directly available: oui not directly available: non
embauche	Has the candidate been hired	binary	0: not hired 1: hired

Table 1 Description

Missing values:

```
> #missing values
> sum(is.na(data))
[1] 27
>
```

Figuare5 missing values

The five-number summary of the dataset:

Example: It can be noticed that the dataset contains candidates with ages in the range between -3 and 74 years old. The average age of the candidates in the dataset is 35 years old. As shown in figure6.

```
> # print five number summary
> #summary(data)
> summary(data$X)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0   248.5   504.0   500.9   750.5   998.0
>

> summary(data$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  10.00   29.00   35.00   34.85   41.00   59.00
>

> summary(data$exp)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.000   8.000  10.000   9.593  11.000  15.000
>

> summary(data$salaire)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  21735   31462   35132   35075   38528   46944
>

> summary(data$note)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  29.97   63.65   74.56   75.10   87.08  121.14
>

 > summary(data$embauche)
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.0000  0.0000  0.0000  0.1294  0.0000  1.0000
 >
```

Figure6 five number summary

Outlier Detection and box plots:

Outliers in the dataset affect the effectiveness of the prediction models and therefore it's important to detect and remove outliers from the dataset. *Figure 7,8,9 and 10.*

age boxplot before removing outliers

```
> boxplot(data$age)
> boxplot(data$age)$out
 [1] 61 65  8 62  8  8  5 63  5 62 62 62  8
> outliers1 <- boxplot(data$age, plot=FALSE)$out
> print(outliers1)
 [1] 61 65  8 62  8  8  5 63  5 62 62 62  8
> data[which(data$age %in% outliers1),]
```

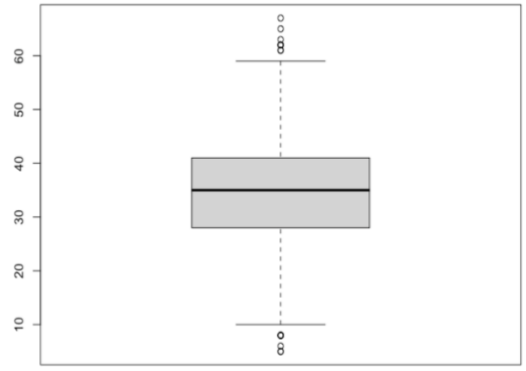


Figure7

The figure7 show us that there are outliers created in 60 and above, and between almost 5-10

note boxplot before removing outliers

```
> boxplot(data$note)
> boxplot(data$note)$out
[1] 127.58 125.47  12.63
> outliers3 <- boxplot(data$note, plot=FALSE)$out
> print(outliers3)
[1] 127.58 125.47  12.63
> data[which(data$note %in% outliers3),]
```

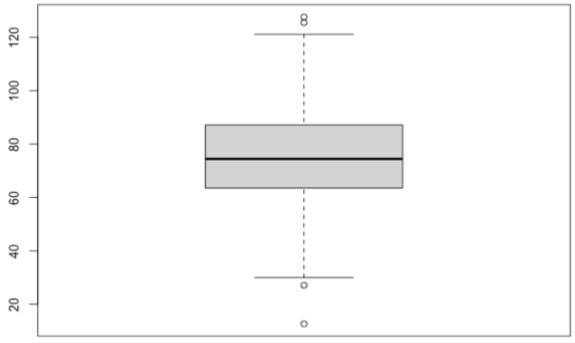
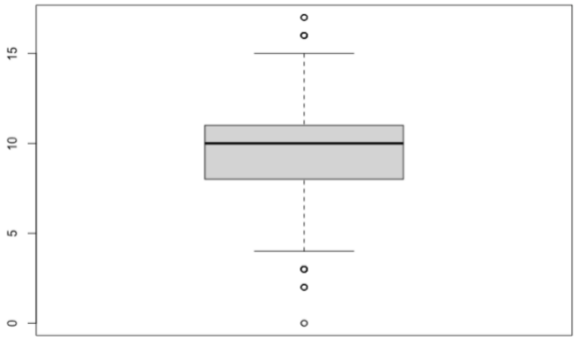


Figure8

The figure8 show us that there are outliers created in above 120 and in almost 15 and 30

Exp boxplot before removing outliers

```
> #data preprpccising
> #removing outliers
> library(outliers)
> #data$exp[which(data$exp>420)] <- c(data$exp[which(data&exp>420)]*2)
> boxplot(data$exp)
> boxplot(data$exp)$out
 [1]  3 16  3  3  3  3  3  2  3 17 17  3  3  0 16  3  3  3  3  3  0 16 16  2  2  3  3  2  2  3  3  3 16 17 16 16 16 16 16  3
[41] 16 16 17 16
> outliers <- boxplot(data$exp, plot=FALSE)$out
> print(outliers)
 [1]  3 16  3  3  3  3  3  2  3 17 17  3  3  0 16  3  3  3  3  3  0 16 16  2  2  3  3  2  2  3  3  3 16 17 16 16 16 16 16  3
[41] 16 16 17 16
> data[which(data$exp %in% outliers),]
```

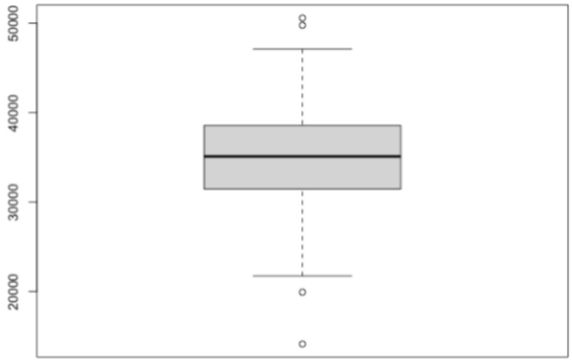


The figure9 show us that there are outliers created in 16 and above and between 0-4

Figure9

salarie boxplot before removing outliers

```
> boxplot(data$salaire)
> boxplot(data$salaire)$out
[1] 19925 50575 14128 49761
> outliers2 <- boxplot(data$salaire, plot=FALSE)$out
> print(outliers2)
[1] 19925 50575 14128 49761
> data[which(data$salaire %in% outliers2),]
```



The figure10 show us that there are outliers created in 5000 and above and between almost 1500 and 2000

Figure10

Description: After box plotting which provided us as box plots it shows that all attributes have outliers that need to be removed later on.

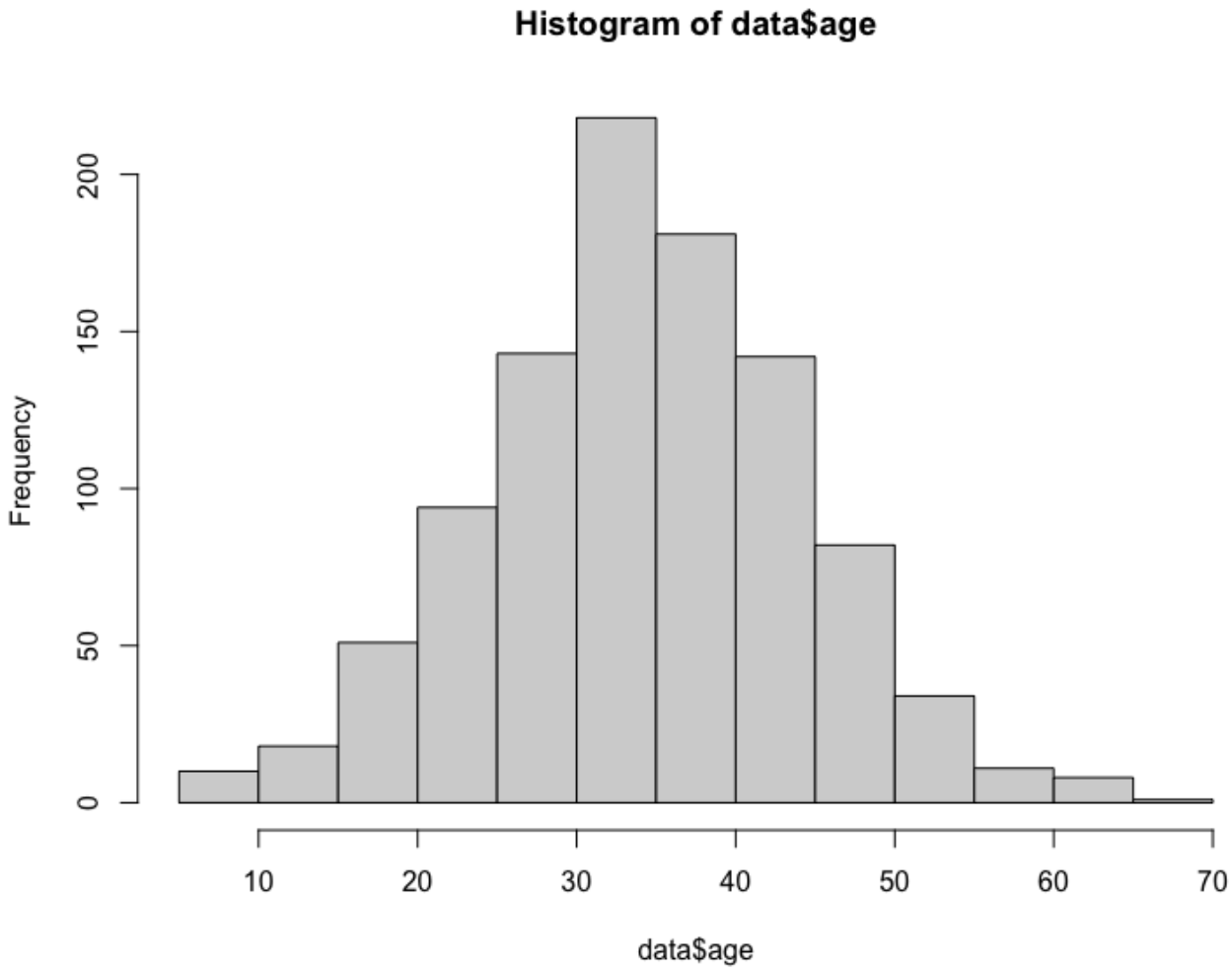


Figure11

Description: The age distribution *Figure* 11 over candidates in the dataset can be illustrated easily using a histogram as it is observed that the age of most of the candidates in the dataset ranges from 30 to 45 years old.

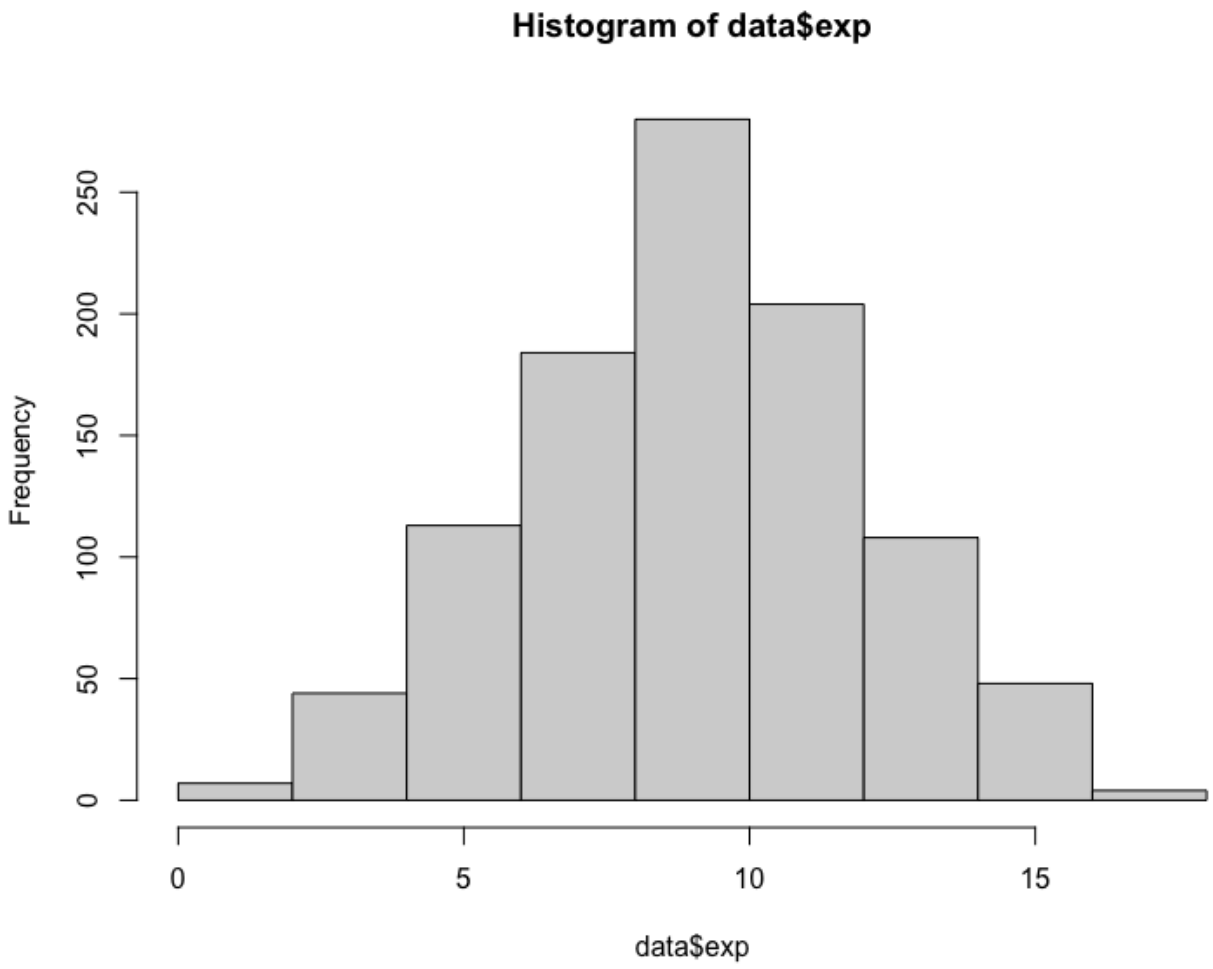


Figure12

Description: The exp distribution *Figure* 12 over candidates in the dataset shows that most of the candidates have years of experience between (5-15) years

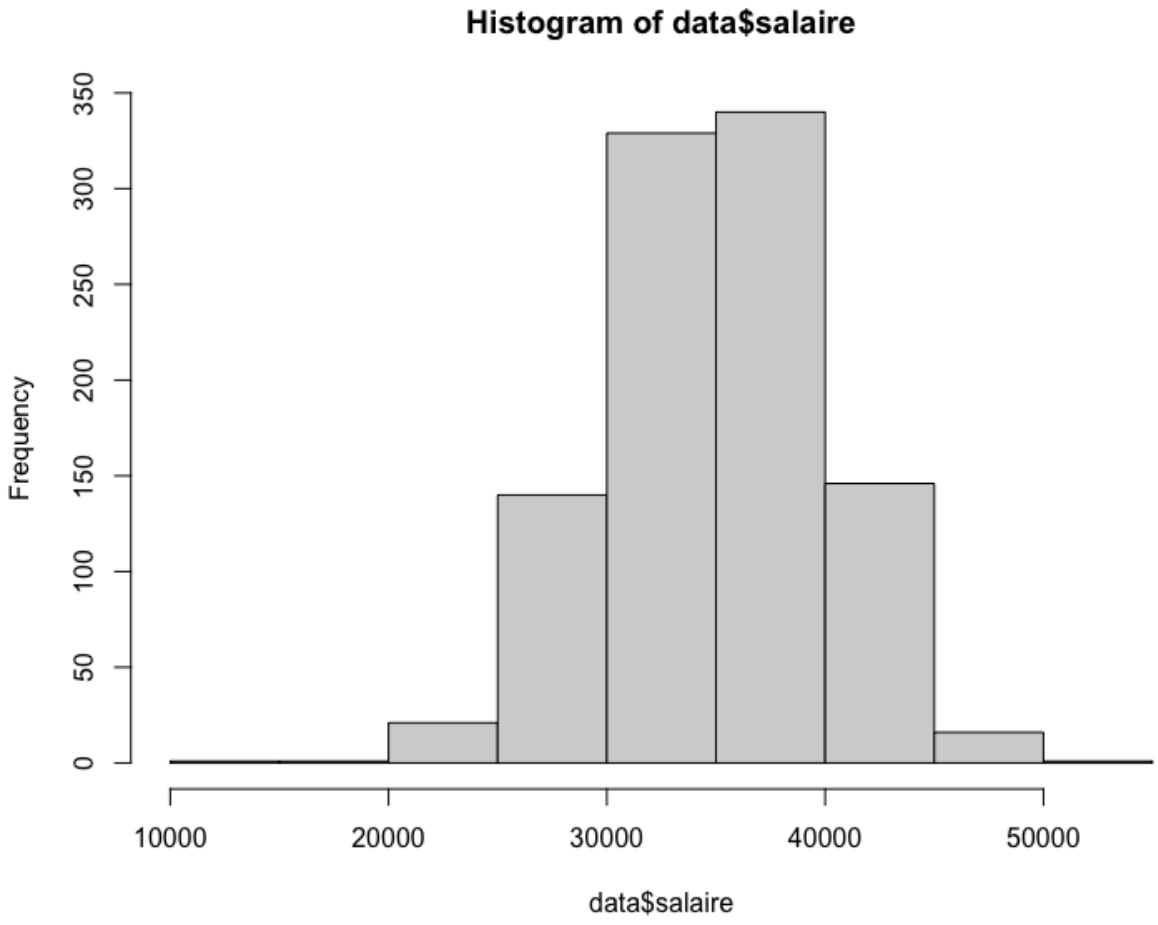


Figure13

Description: The salaire distribution *Figure* 13 over candidates in the dataset shows that most of the candidates receive salaries between (30000 and 40000)

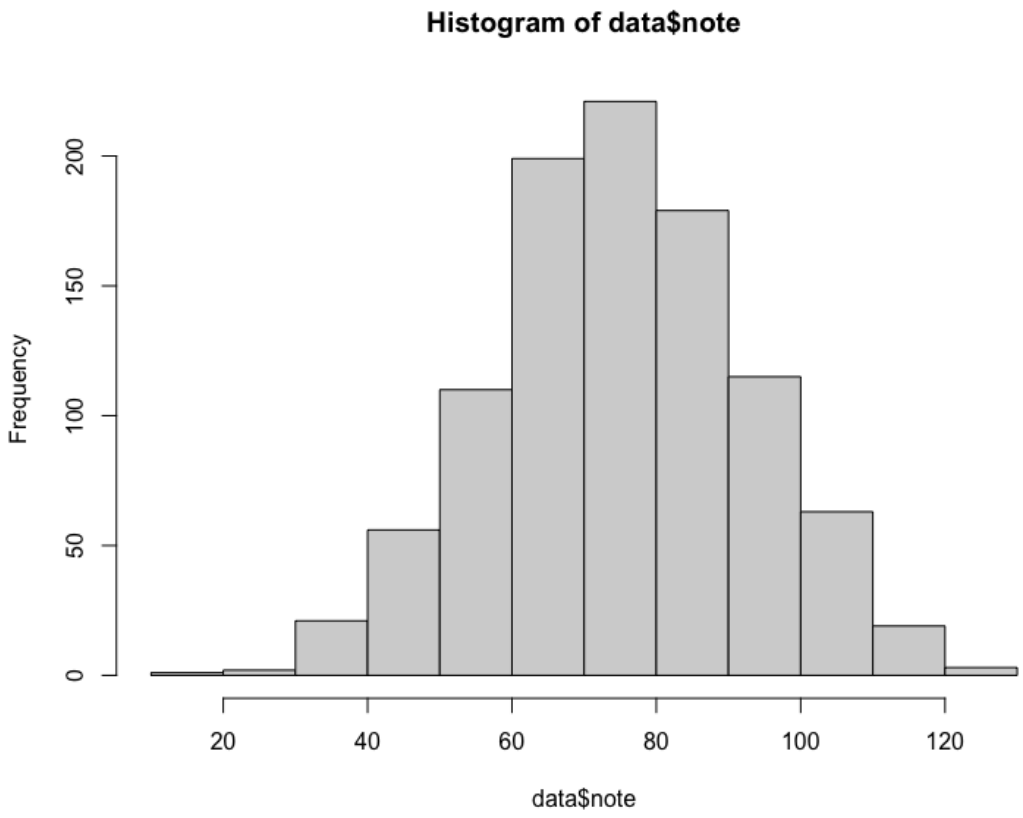


Figure14

Description: The note distribution *Figure* 14 over candidates in the dataset shows that most of the candidates have grade between(60-80)

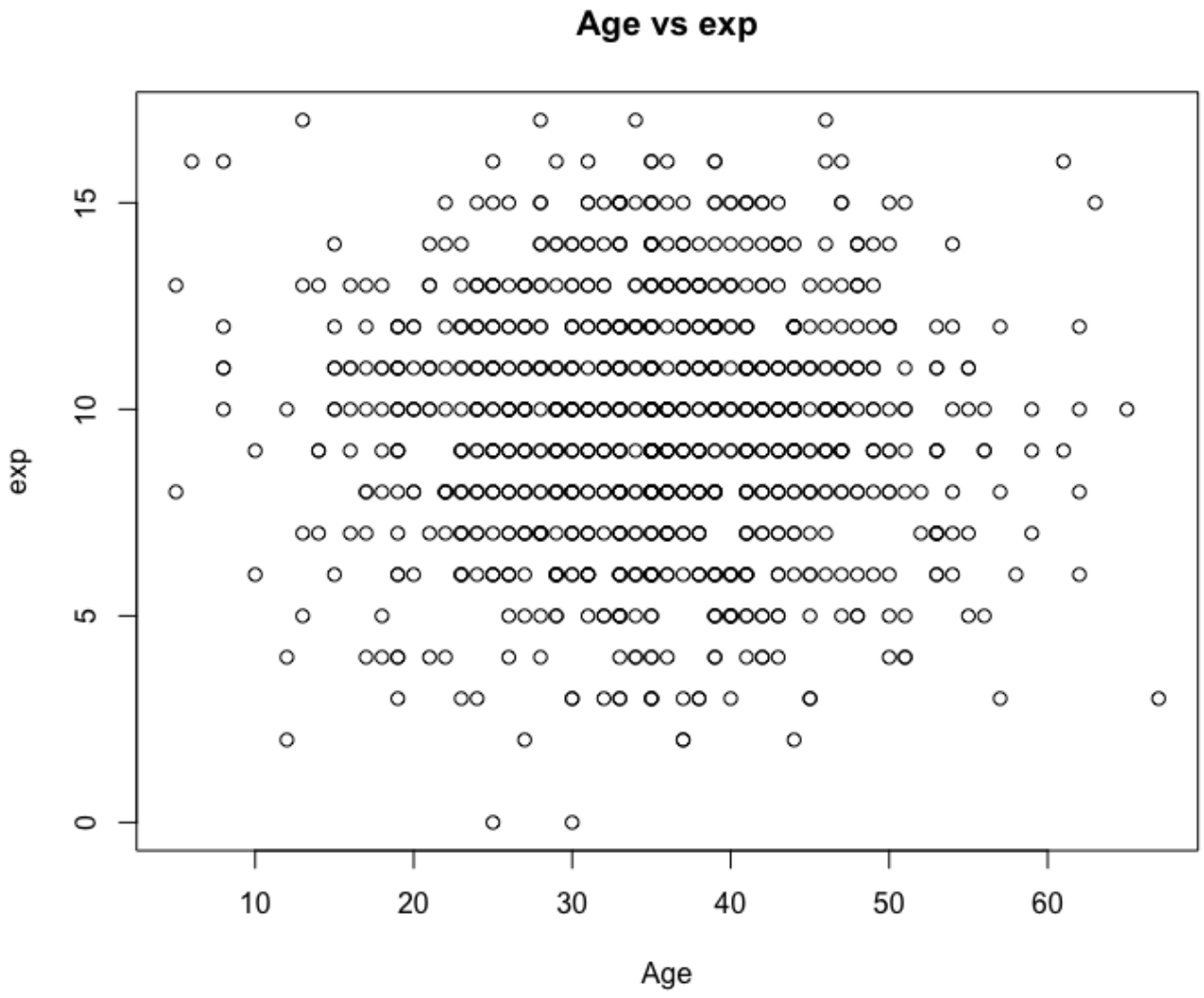


Figure15

Description: This scatter plot *Figure* 15 shows that there is no relationship between age and exp so there is no correlation between age and exp . Because the value range is spread through the scatter plot.

Hairbarplot

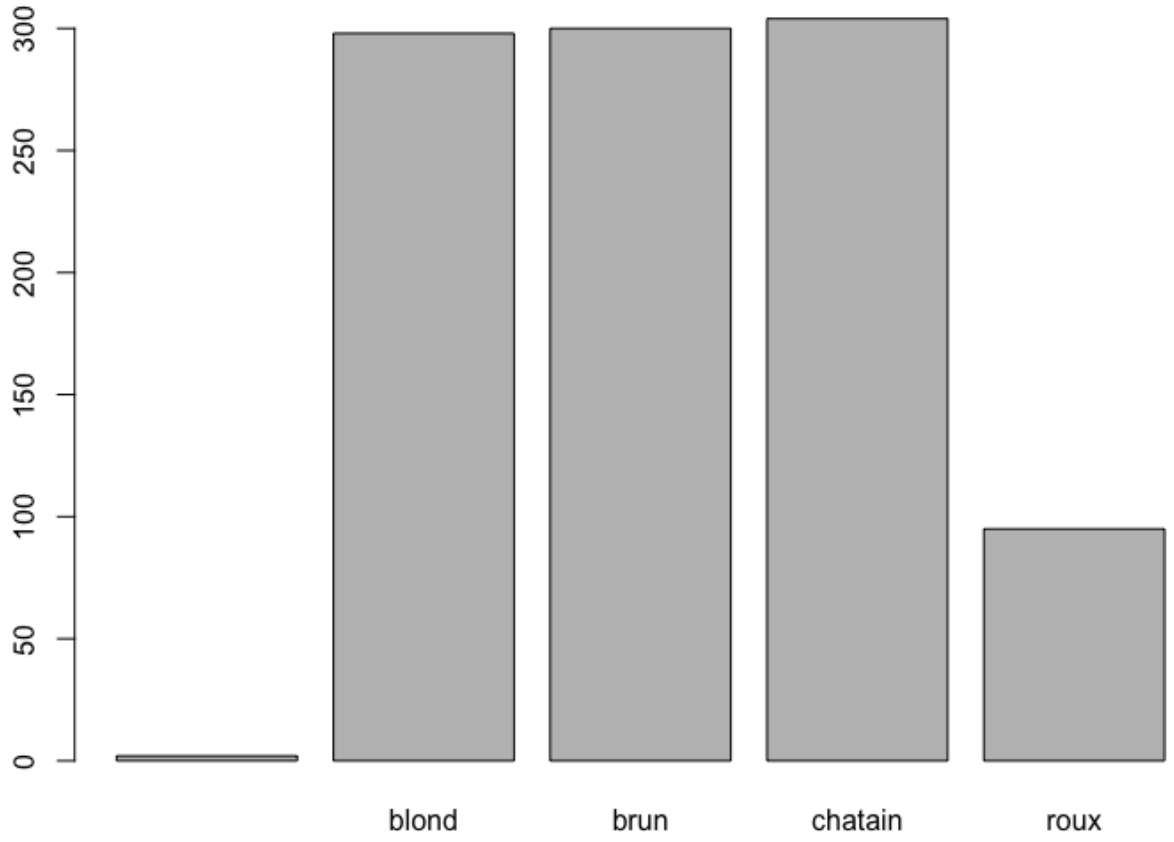


Figure16

Description: From the bar plot *Figure16* it shows that there are almost 300 blond and 300 brun and more than 300 chatain and 100 roux

Date barplot

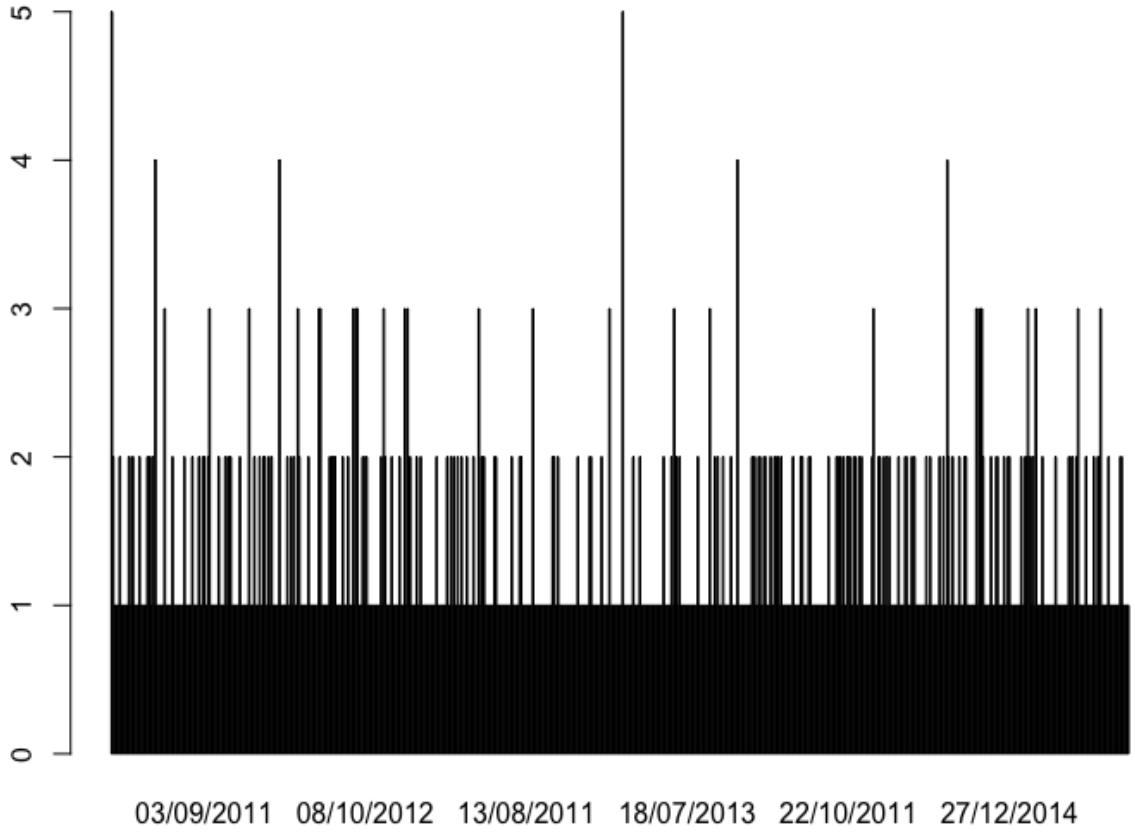


Figure17

Description: From the bar plot *Figure17* it shows that there are random distribution for dates from 2011-2014

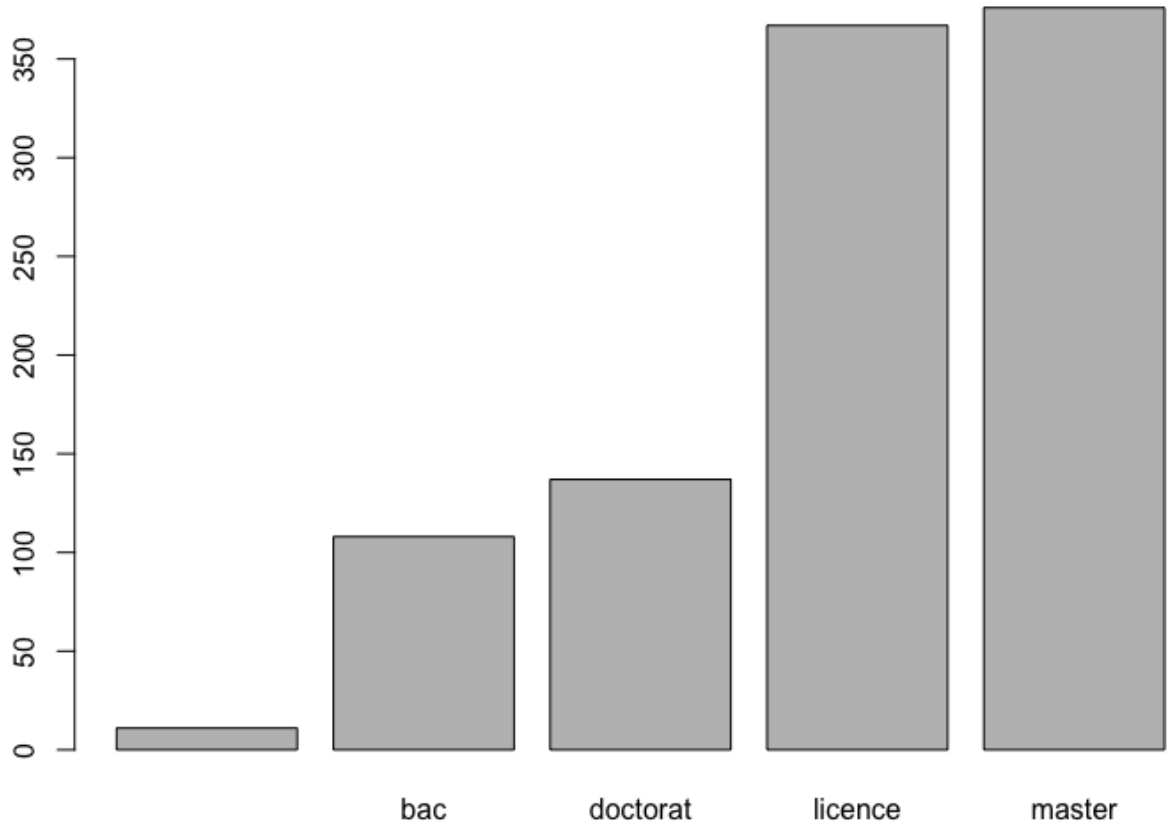


Figure18

Description: From the bar plot *Figure* 18 it shows that there are more than 100 bac and doctorat and 50 licence and more than 350 master.

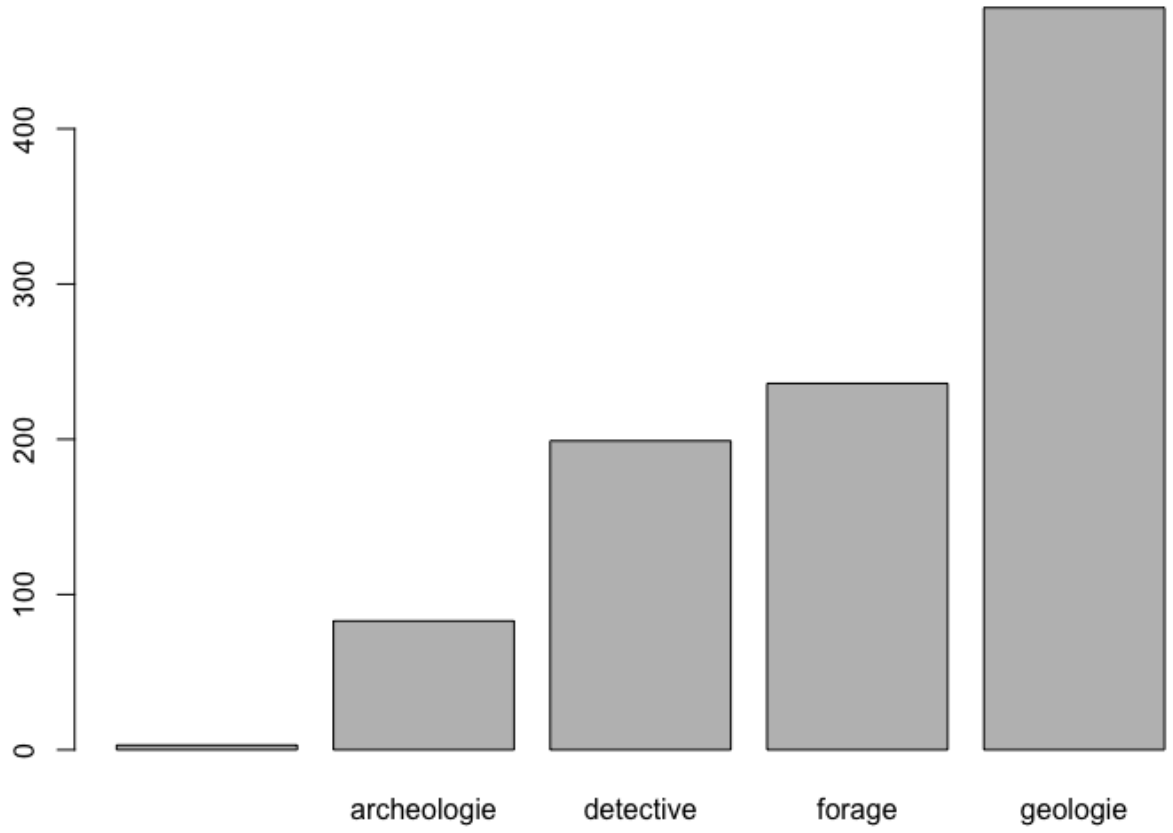


Figure19

Description: From the bar plot *Figure19* it shows that there are less than 100 archeologie and 200 detective and more than 200 forage and more than 400 geologie.

4 Data preprocessing

Our dataset before preprocessing *Figure20* **: is contain 999 object and 12 attribute.**

RStudio Source Editor

data

Filter

	X	date	cheveux	age	exp	salaire	sexe	diplome	specialite	note	dispo	embauche
1	0	2012-06-02	roux	25	9	26803	F	licence	geologie	97.08	non	0
2	1	2011-04-21	blond	35	13	38166	M	licence	forage	63.86	non	0
3	2	2012-09-07	blond	29	13	35207	M	licence	geologie	78.50	non	0
4	3	2011-07-01	brun	NA	12	32442	M	licence	geologie	45.09	non	0
5	4	2012-08-07	roux	35	6	28533	F	licence	detective	81.91	non	0
6	5	2014-02-12	chatain	37	8	38558	M	master	geologie	63.46	non	1
7	6	2013-11-11	brun	33	12	39476	M	master	geologie	50.20	oui	0
8	7	2012-03-10	roux	31	10	42392	M	licence	forage	62.20	oui	0
9	8	2014-10-17	chatain	43	10	28625	M	doctorat	geologie	65.17	non	1
10	9	2011-06-04	chatain	28	11	32454	M	master	forage	66.93	non	1
11	10	2014-08-06	brun	50	12	38516	F	licence	geologie	58.93	non	0
12	11	2010-04-22	blond	43	8	38719	M	master	archeologie	80.07	oui	0
13	12	2011-10-23	brun	44	9	28688	M	doctorat	forage	115.76	oui	0
14	13	2014-06-25	brun	39	4	30822	F	licence	geologie	66.85	non	0
15	14	2011-10-15	roux	23	14	33882	M	licence	geologie	61.46	non	0
16	15	2010-07-12	brun	38	3	40889	M	master	geologie	62.85	oui	0
17	16	2012-02-05	brun	31	10	28507	M	doctorat	forage	104.26	oui	0
18	17	2010-07-03	blond	37	13	41512	M	bac	geologie	55.91	non	0
19	18	2010-07-21	blond	35	14	30359	M	licence	geologie	94.58	non	0
20	19	2014-07-22	chatain	44	8	28956	M	master	geologie	74.82	non	0
21	20	2010-02-12	brun	31	12	34483	M	master	forage	66.27	oui	0
22	21	2014-05-22	chatain	30	10	39069	F	licence	forage	65.19	non	0
23	22	2010-01-04	brun	39	13	34632	M	master	geologie	81.35	non	0
24	23	2012-03-20	brun	31	8	41582	M	licence	geologie	58.94	non	0
25	24	2012-01-16	blond	35	8	30782	F	licence	geologie	90.31	non	0

Figure20

Data cleaning:

1-Filling in Missing values:

One of the most important preprocessing steps in data mining is finding instances that contain missing values and treating these missing values.

In this project, methods for checking the existence of missing values are first used as follows: *Figure21*

```
> #missing values
> sum(is.na(data))
[1] 27
>
```

Figure 21

Where the function “IS.NA ()” was used to check if there are missing values or not in the dataset and the “SUM” was used to sum all missing values in the dataset.

total number of missing values in the dataset is 27 which indicates that there are a lot of missing values that we need to fill in the dataset.

```
> #replace missing data with average value
> data$date=ifelse(is.na(data$date),ave(data$date,FUN = function(x)mean(x,na.rm=TRUE)),data$date)
> data$cheveux=ifelse(is.na(data$cheveux),ave(data$cheveux,FUN = function(x)mean(x,na.rm=TRUE)),data$chev
eux)
> data$age=ifelse(is.na(data$age),ave(data$age,FUN = function(x)mean(x,na.rm=TRUE)),data$age)
> data$exp=ifelse(is.na(data$exp),ave(data$exp,FUN = function(x)mean(x,na.rm=TRUE)),data$exp)
> data$salaire=ifelse(is.na(data$salaire),ave(data$salaire,FUN = function(x)mean(x,na.rm=TRUE)),data$sala
ire)
> data$sexe=ifelse(is.na(data$sexe),ave(data$sexe,FUN = function(x)mean(x,na.rm=TRUE)),data$sexe)
> data$diplome=ifelse(is.na(data$diplome),ave(data$diplome,FUN = function(x)mean(x,na.rm=TRUE)),data$dipl
ome)
> data$specialite=ifelse(is.na(data$specialite),ave(data$specialite,FUN = function(x)mean(x,na.rm=TRUE)),
data$specialite)
> data$note=ifelse(is.na(data$note),ave(data$note,FUN = function(x)mean(x,na.rm=TRUE)),data$note)
> sum(is.na(data))
[1] 0
>
```

Figure22

We replaced the missing values with the average value and the sum of the missing values at the end of the replacement process became zero. *Figure22*

2-Outliers Removal

Outliers should be removed from the dataset since they can affect on the accuracy of the data mining models. In this preprocessing step, outliers were detected and removed from the dataset.

Removing outlier from salaire:

```
> boxplot(data$salaire)
> boxplot(data$salaire)$out
[1] 19925 50575 14128 49761
> outliers2 <- boxplot(data$salaire, plot=FALSE)$out
> print(outliers2)
[1] 19925 50575 14128 49761
> data[which(data$salaire %in% outliers2),]
```

Figure23

Removing outlier from age:

```
> boxplot(data$age)
> boxplot(data$age)$out
 [1] 61 65  8 62  8  8  5 63  5 62 62 62  8
> outliers1 <- boxplot(data$age, plot=FALSE)$out
> print(outliers1)
 [1] 61 65  8 62  8  8  5 63  5 62 62 62  8
> data[which(data$age %in% outliers1),]
```

Figure24

salaire boxplot after removing outlier

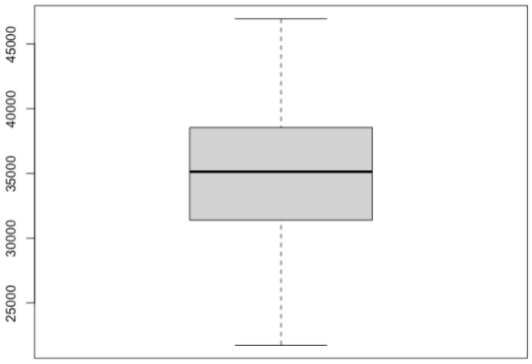


Figure25

age boxplot after removing outlier

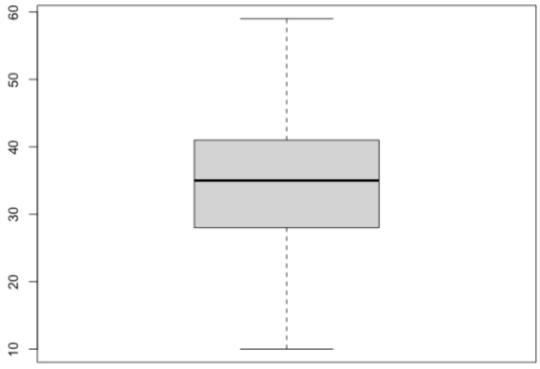


Figure26

note boxplot after removing outlier

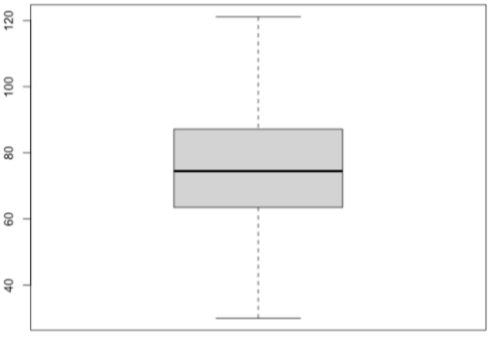


Figure27

Removing outlier from note

```
> boxplot(data$note)
> boxplot(data$note)$out
[1] 127.58 125.47  12.63
> outliers3 <- boxplot(data$note, plot=FALSE)$out
> print(outliers3)
[1] 127.58 125.47  12.63
> data[which(data$note %in% outliers3),]
```

Figure28

Removing outlier from exp:

```
> #data preprpccising
> #removing outliers
> library(outliers)
> #data$exp[which(data$exp>420)] <- c(data$exp[which(data&exp>420)]*2)
> boxplot(data$exp)
> boxplot(data$exp)$out
 [1]  3 16  3  3  3  3  3  2  3 17 17  3  3  0 16  3  3  3  3  3  0 16 16  2  2  3  3  2  2  3  3  3 16 17 16 16 16 16 16  3
[41] 16 16 17 16
> outliers <- boxplot(data$exp, plot=FALSE)$out
> print(outliers)
 [1]  3 16  3  3  3  3  3  2  3 17 17  3  3  0 16  3  3  3  3  3  0 16 16  2  2  3  3  2  2  3  3  3 16 17 16 16 16 16 16  3
[41] 16 16 17 16
> data[which(data$exp %in% outliers),]
```

Figure29

Exp boxplot after removing

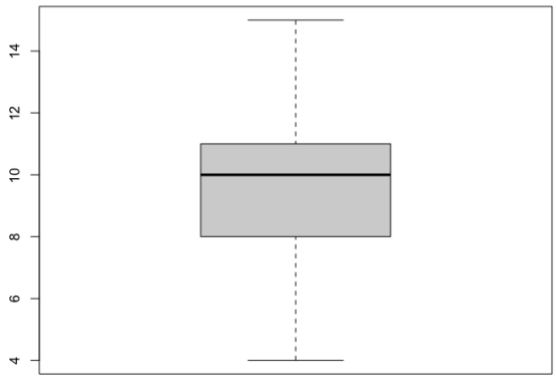


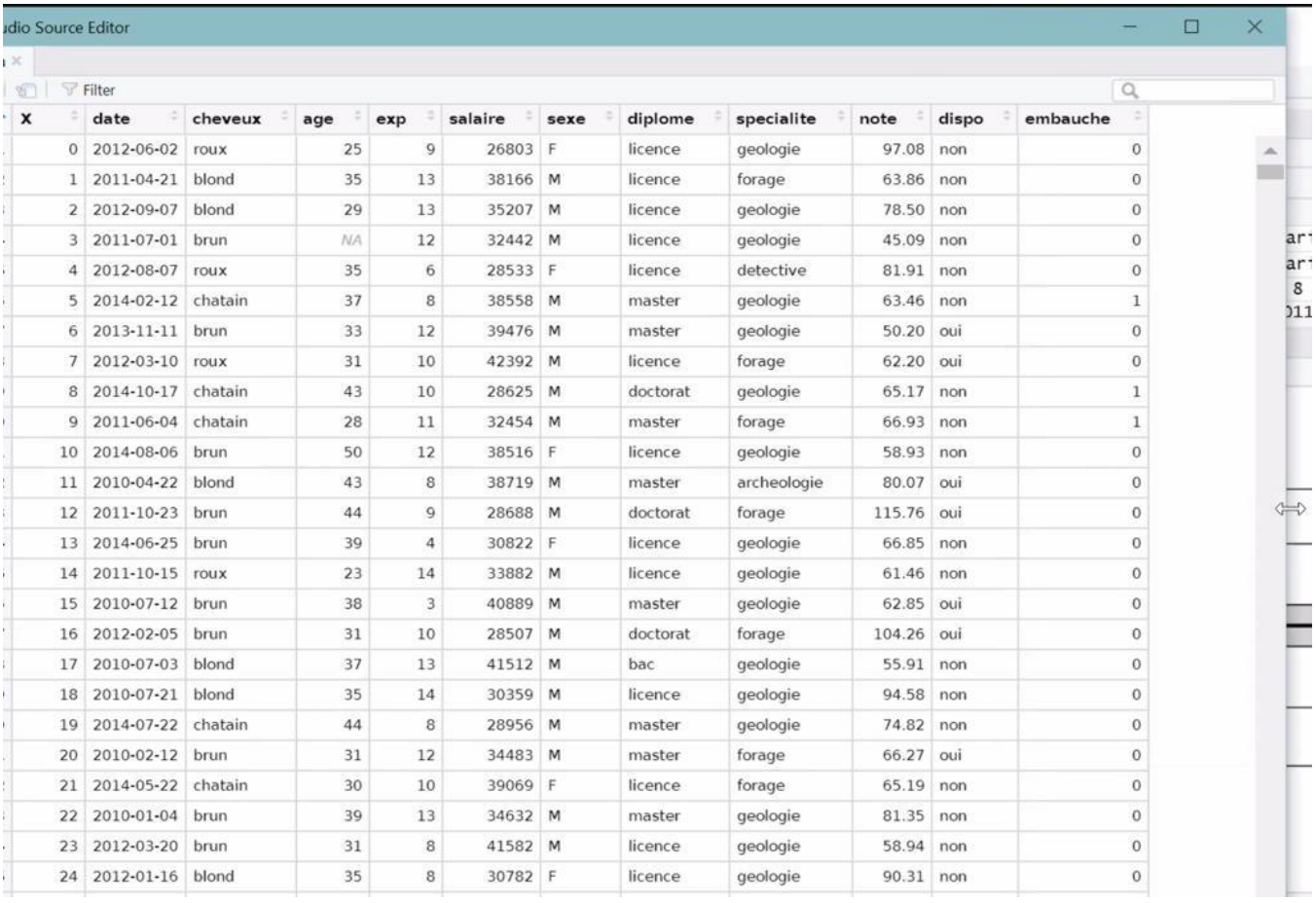
Figure30

Data transformation:

3-Normalization

Normalization of features can speed up training of some data mining models since we have attributes in different scale, we need to normalize to bring the dataset to a common scale (between 0 and 1) while keeping the distribution of the variables the same

the dataset before normalization *Figure31*:



udio Source Editor

Filter

X	date	cheveux	age	exp	salaire	sexe	diplome	specialite	note	dispo	embauche
0	2012-06-02	roux	25	9	26803	F	licence	geologie	97.08	non	0
1	2011-04-21	blond	35	13	38166	M	licence	forage	63.86	non	0
2	2012-09-07	blond	29	13	35207	M	licence	geologie	78.50	non	0
3	2011-07-01	brun	NA	12	32442	M	licence	geologie	45.09	non	0
4	2012-08-07	roux	35	6	28533	F	licence	detective	81.91	non	0
5	2014-02-12	chatain	37	8	38558	M	master	geologie	63.46	non	1
6	2013-11-11	brun	33	12	39476	M	master	geologie	50.20	oui	0
7	2012-03-10	roux	31	10	42392	M	licence	forage	62.20	oui	0
8	2014-10-17	chatain	43	10	28625	M	doctorat	geologie	65.17	non	1
9	2011-06-04	chatain	28	11	32454	M	master	forage	66.93	non	1
10	2014-08-06	brun	50	12	38516	F	licence	geologie	58.93	non	0
11	2010-04-22	blond	43	8	38719	M	master	archeologie	80.07	oui	0
12	2011-10-23	brun	44	9	28688	M	doctorat	forage	115.76	oui	0
13	2014-06-25	brun	39	4	30822	F	licence	geologie	66.85	non	0
14	2011-10-15	roux	23	14	33882	M	licence	geologie	61.46	non	0
15	2010-07-12	brun	38	3	40889	M	master	geologie	62.85	oui	0
16	2012-02-05	brun	31	10	28507	M	doctorat	forage	104.26	oui	0
17	2010-07-03	blond	37	13	41512	M	bac	geologie	55.91	non	0
18	2010-07-21	blond	35	14	30359	M	licence	geologie	94.58	non	0
19	2014-07-22	chatain	44	8	28956	M	master	geologie	74.82	non	0
20	2010-02-12	brun	31	12	34483	M	master	forage	66.27	oui	0
21	2014-05-22	chatain	30	10	39069	F	licence	forage	65.19	non	0
22	2010-01-04	brun	39	13	34632	M	master	geologie	81.35	non	0
23	2012-03-20	brun	31	8	41582	M	licence	geologie	58.94	non	0
24	2012-01-16	blond	35	8	30782	F	licence	geologie	90.31	non	0

Figure31

Normalization code : *Figure32*

```
#################
# Normalization of features

str(data)
normalize<- function(x){(x-min(x))/(max(x)-min(x))}

dataN = data
dataN$age <-normalize(data$age)
dataN$salaire <-normalize(data$salaire)
dataN$exp <- normalize(data$exp)
dataN$note <- normalize(data$note)

str(dataN)
```

Figure32

the dataset after normalization: *Figure33*

RStudio Source Editor

dataN

Filter

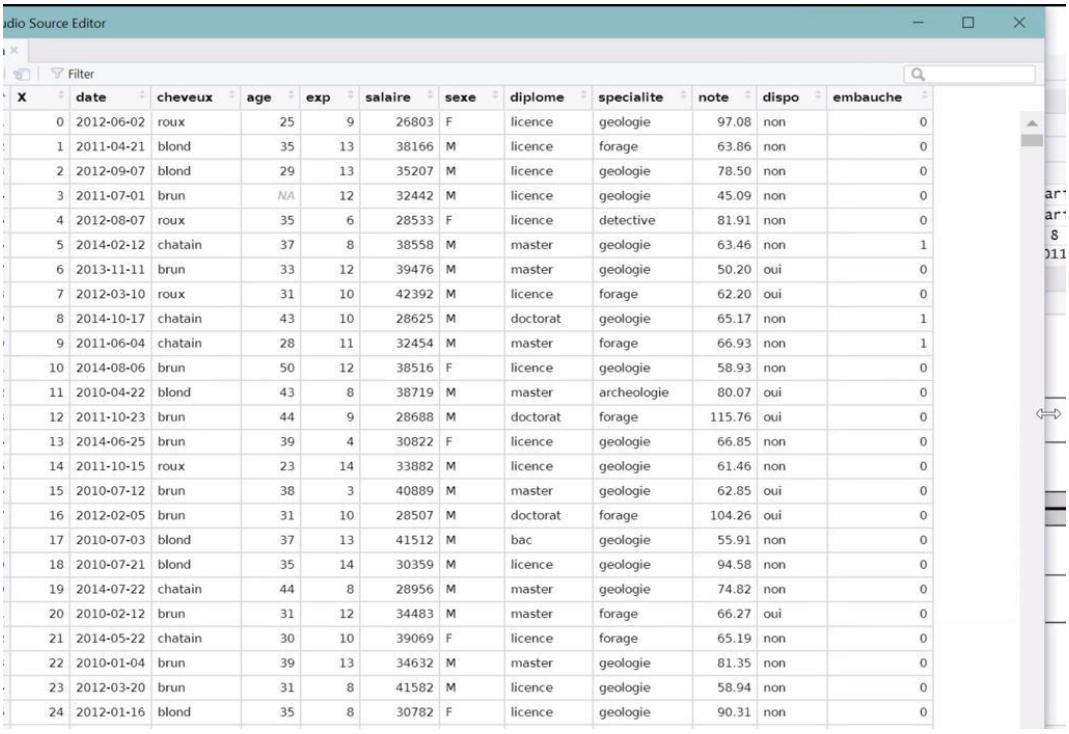
	X	date	cheveux	age	exp	salaire	sexe	diplome	specialite	note	dispo	embauche
1	0	2012-06-02	roux	0.3733333	0.4545455	0.3180757	F	licence	geologie	0.6570537	non	0
2	1	2011-04-21	blond	0.5066667	0.6363636	0.6032272	M	licence	forage	0.4101382	non	0
3	2	2012-09-07	blond	0.4266667	0.6363636	0.5289719	M	licence	geologie	0.5189535	non	0
4	3	2011-07-01	brun	0.5067269	0.5909091	0.4595849	M	licence	geologie	0.2706258	non	0
5	4	2012-08-07	roux	0.5066667	0.3181818	0.3614896	F	licence	detective	0.5442991	non	0
6	5	2014-02-12	chatain	0.5333333	0.4090909	0.6130643	M	master	geologie	0.4071652	non	1
7	6	2013-11-11	brun	0.4800000	0.5909091	0.6361013	M	master	geologie	0.3086071	oui	0
8	7	2012-03-10	roux	0.4533333	0.5000000	0.7092775	M	licence	forage	0.3977999	oui	0
9	8	2014-10-17	chatain	0.6133333	0.5000000	0.3637983	M	doctorat	geologie	0.4198751	non	1
10	9	2011-06-04	chatain	0.4133333	0.5454545	0.4598861	M	master	forage	0.4329567	non	1
11	10	2014-08-06	brun	0.7066667	0.5909091	0.6120103	F	licence	geologie	0.3734949	non	0
12	11	2010-04-22	blond	0.6133333	0.4090909	0.6171046	M	master	archeologie	0.5306229	oui	0
13	12	2011-10-23	brun	0.6266667	0.4545455	0.3653793	M	doctorat	forage	0.7958971	oui	0
14	13	2014-06-25	brun	0.5600000	0.2272727	0.4189315	F	licence	geologie	0.4323621	non	0
15	14	2011-10-15	roux	0.3466667	0.6818182	0.4957213	M	licence	geologie	0.3922997	non	0
16	15	2010-07-12	brun	0.5466667	0.1818182	0.6715601	M	master	geologie	0.4026312	oui	0
17	16	2012-02-05	brun	0.4533333	0.5000000	0.3608372	M	doctorat	forage	0.7104207	oui	0
18	17	2010-07-03	blond	0.5333333	0.6363636	0.6871942	M	bac	geologie	0.3510480	non	0
19	18	2010-07-21	blond	0.5066667	0.6818182	0.4073126	M	licence	geologie	0.6384718	non	0
20	19	2014-07-22	chatain	0.6266667	0.4090909	0.3721047	M	master	geologie	0.4916010	non	0
21	20	2010-02-12	brun	0.4533333	0.5909091	0.5108033	M	master	forage	0.4280511	oui	0
22	21	2014-05-22	chatain	0.4400000	0.5000000	0.6258877	F	licence	forage	0.4200238	non	0
23	22	2010-01-04	brun	0.5600000	0.6363636	0.5145424	M	master	geologie	0.5401368	non	0
24	23	2012-03-20	brun	0.4533333	0.4090909	0.6889508	M	licence	geologie	0.3735692	non	0
25	24	2012-01-16	blond	0.5066667	0.4090909	0.4179277	F	licence	geologie	0.6067341	non	0

Figure33

4-Encoding

We discovered that we have 4 nominal attributes(cheveux,diplome,dispo,specialite) that needs to be encoded to numeric values so we can do calculations on it.

the dataset before encoding: *Figure34*



	X	date	cheveux	age	exp	salaire	sexe	diplome	specialite	note	dispo	embauche
1	0	2012-06-02	roux	25	9	26803	F	licence	geologie	97.08	non	0
2	1	2011-04-21	blond	35	13	38166	M	licence	forage	63.86	non	0
3	2	2012-09-07	blond	29	13	35207	M	licence	geologie	78.50	non	0
4	3	2011-07-01	brun	NA	12	32442	M	licence	geologie	45.09	non	0
5	4	2012-08-07	roux	35	6	28533	F	licence	detective	81.91	non	0
6	5	2014-02-12	chatain	37	8	38558	M	master	geologie	63.46	non	1
7	6	2013-11-11	brun	33	12	39476	M	master	geologie	50.20	oui	0
8	7	2012-03-10	roux	31	10	42392	M	licence	forage	62.20	oui	0
9	8	2014-10-17	chatain	43	10	28625	M	doctorat	geologie	65.17	non	1
10	9	2011-06-04	chatain	28	11	32454	M	master	forage	66.93	non	1
11	10	2014-08-06	brun	50	12	38516	F	licence	geologie	58.93	non	0
12	11	2010-04-22	blond	43	8	38719	M	master	archeologie	80.07	oui	0
13	12	2011-10-23	brun	44	9	28688	M	doctorat	forage	115.76	oui	0
14	13	2014-06-25	brun	39	4	30822	F	licence	geologie	66.85	non	0
15	14	2011-10-15	roux	23	14	33882	M	licence	geologie	61.46	non	0
16	15	2010-07-12	brun	38	3	40889	M	master	geologie	62.85	oui	0
17	16	2012-02-05	brun	31	10	28507	M	doctorat	forage	104.26	oui	0
18	17	2010-07-03	blond	37	13	41512	M	bac	geologie	55.91	non	0
19	18	2010-07-21	blond	35	14	30359	M	licence	geologie	94.58	non	0
20	19	2014-07-22	chatain	44	8	28956	M	master	geologie	74.82	non	0
21	20	2010-02-12	brun	31	12	34483	M	master	forage	66.27	oui	0
22	21	2014-05-22	chatain	30	10	39069	F	licence	forage	65.19	non	0
23	22	2010-01-04	brun	39	13	34632	M	master	geologie	81.35	non	0
24	23	2012-03-20	brun	31	8	41582	M	licence	geologie	58.94	non	0
25	24	2012-01-16	blond	35	8	30782	F	licence	geologie	90.31	non	0

Figure34

The encoding code: *Figure35*

```
#########
#ENCODING

data$dispo = factor(data$dispo, levels = c("non", "oui"), labels = c(0, 1))

data$cheveux = factor(data$cheveux, levels = c("chatain", "brun","blond" ,"roux"), labels = c(1,2,3,4))

data$diplome = factor(data$diplome, levels = c("doctorat", "master","bac" ,"licence"), labels = c(1,2,3,4))
data$specialite = factor(data$specialite, levels = c("geologie", "forage","detective" ,"archeologie"), labels = c(1,2,3,4))
str(data)
```

Figure35

the dataset after encoding: *Figure36*

Filter

	X	date	cheveux	age	exp	salaire	sexe	diplome	specialite	note	dispo	embauche
1	0	02/06/2012	4	25.00000	9.000000	26803.00	F	4	1	97.08000	0	0
2	1	21/04/2011	3	35.00000	13.000000	38166.00	M	4	2	63.86000	0	0
3	2	07/09/2012	3	29.00000	13.000000	35207.00	M	4	1	78.50000	0	0
4	3	01/07/2011	2	34.84839	12.000000	32442.00	M	4	1	45.09000	0	0
5	4	07/08/2012	4	35.00000	6.000000	28533.00	F	4	3	81.91000	0	0
6	5	12/02/2014	1	37.00000	8.000000	38558.00	M	2	1	63.46000	0	1
7	6	11/11/2013	2	33.00000	12.000000	39476.00	M	2	1	50.20000	1	0
8	7	10/03/2012	4	31.00000	10.000000	42392.00	M	4	2	62.20000	1	0
9	8	17/10/2014	1	43.00000	10.000000	28625.00	M	1	1	65.17000	0	1
10	9	04/06/2011	1	28.00000	11.000000	32454.00	M	2	2	66.93000	0	1
11	10	06/08/2014	2	50.00000	12.000000	38516.00	F	4	1	58.93000	0	0
12	11	22/04/2010	3	43.00000	8.000000	38719.00	M	2	4	80.07000	1	0
13	12	23/10/2011	2	44.00000	9.000000	28688.00	M	1	2	115.76000	1	0
14	13	25/06/2014	2	39.00000	4.000000	30822.00	F	4	1	66.85000	0	0
15	14	15/10/2011	4	23.00000	14.000000	33882.00	M	4	1	61.46000	0	0
17	16	05/02/2012	2	31.00000	10.000000	28507.00	M	1	2	104.26000	1	0
18	17	03/07/2010	3	37.00000	13.000000	41512.00	M	3	1	55.91000	0	0
19	18	21/07/2010	3	35.00000	14.000000	30359.00	M	4	1	94.58000	0	0
20	19	22/07/2014	1	44.00000	8.000000	28956.00	M	2	1	74.82000	0	0
21	20	12/02/2010	2	31.00000	12.000000	34483.00	M	2	2	66.27000	1	0
22	21	22/05/2014	1	30.00000	10.000000	39069.00	F	4	2	65.19000	0	0
23	22	04/01/2010	2	39.00000	13.000000	34632.00	M	2	1	81.35000	0	0
24	23	20/03/2012	2	31.00000	8.000000	41582.00	M	4	1	58.94000	0	0
25	24	16/01/2012	3	35.00000	8.000000	30782.00	F	4	1	90.31000	0	0
26	25	13/05/2014	1	31.00000	12.000000	33238.00	F	2	3	70.39000	0	0
27	26	12/04/2013	2	33.00000	8.000000	35075.23	F	4	3	56.99000	0	0
28	27	18/06/2013	2	42.00000	8.000000	30641.00	F	4	1	78.89000	0	0
29	28	02/01/2010	1	27.00000	13.000000	31505.00	F	2	4	88.33000	0	0
30	29	25/07/2010	1	19.00000	9.000000	31357.00	M	1	3	88.69000	1	0
31	30	21/03/2010	2	19.00000	4.000000	39031.00	M	2	1	57.48000	1	0
32	31	29/05/2014	1	40.00000	13.000000	37495.00	M	1	1	80.86000	1	0
33	32	05/02/2012	1	43.00000	7.000000	37579.00	M	1	1	91.15000	1	0
34	33	03/09/2012	3	28.00000	13.000000	34485.00	F	4	2	76.94000	0	0
35	34	21/06/2010	3	38.00000	6.000000	41909.00	M	4	1	87.76000	1	1
36	35	04/11/2011	1	43.00000	8.000000	46245.00	M	4	1	49.39000	0	0
37	36	01/08/2010	2	49.00000	8.000000	38020.00	M	2	2	60.63000	1	0
38	37	10/01/2013	2	26.00000	11.000000	31759.00	F	2	2	102.66000	0	1
39	38	14/06/2013	3	59.00000	7.000000	40397.00	M	4	2	60.42000	1	0
40	39	08/11/2014	1	37.00000	6.000000	27002.00	M	2	2	55.83000	0	1

Figure36

5 Data Mining Technique

Classification:

التكنيك اللي استخدمناها عشان نحل مشكلتنا

We used the Decision tree classifier technique to classify candidates who applied for gold digger position into hired (embauche = 1) or not hired (embauche = 0) using the target value (embauche) since we're doing classification the dataset class label and number of classes are known. to do classification we need to split the data into training set and test set. Using training data set to determine the classification model. Then, Using the test data with the classification model to estimate accuracy rate of the classification model. We used Decision tree because it generates understandable rules. It performs classification without requiring much computation.

How we used technique ?

why we used our technique ?

Package:

الباكجز والميثود اللي استخدمناها في الكود

party "for recursive partitioning"

caret "loads packages as needed and assume they are installed"

rpart "builds the classification tree"

rpart.plot "print / plot the tree"

Methods:

nrow " method to return the number of rows of data"

predict "makes prediction for new data"

confusionMatrix "prediction table "

Clustering:

We used K-means clustering technique algorithm that first partition data into groups where data in the same group are similar, K-mean work by randomly select objects as cluster centers and assigns other objects to the nearest cluster center and then improves the clustering by iteratively updating the cluster centers and reassigning the objects to new centers. We used k-mean because it relatively simple to implement , it suitable to large data sets it also guarantees convergence,can warm-start the positions of centroids. K-mean generalizes to clusters of different shapes and sizes.

We used these packages:

cluster

factoextra(visualize the clusters)

NbClust(determine the optimal number)

GGally, plotly (plotting the cluster)

also this libraries: NbClust , cluster , factoextra, GGally, plotly

and because we have non numeric attributes, we convert our attributes to be numeric and we removed the class label form our data set.

```
> datacl$embauche<-NULL
> datacl$diplome<-as.numeric(as.character(datacl$diplome))
> datacl$sexe<-as.numeric(as.character(datacl$sexe))
> datacl$cheveux<-as.numeric(as.character(datacl$cheveux))
> datacl$specialite<-as.numeric(as.character(datacl$specialite))
> datacl$dispo<-as.numeric(as.character(datacl$dispo))
> datacl$specialite<-as.numeric(as.character(datacl$specialite))
> sapply(datacl,class)
   cheveux        age        exp    salaire       sexe    diplome specialite       note
 "numeric"  "numeric"  "numeric"  "numeric"  "numeric"  "numeric"  "numeric"  "numeric"
     dispo
 "numeric"
```

Methods:

- Set.seed(): Reproduce results.
- Scale(): Scaling and centering of matrix-like objects.

• Kmeans(): Classify observations into k groups, based on their similarity. Each group is represented by the mean value of points in the group, known as the cluster centroid.

• Fviz_cluster(): Visualize clustering results.

• silhouette(): find the average silhouette for each cluster and all cluster

Always must be between (-1,1), A good average silhouette is when average sillhoute is = 1

• Fviz_nbclust(): Determining and visualizing the optimal number of clusters.

• ggparcoord(): interpretation clusters relation by choosing specific attributes.

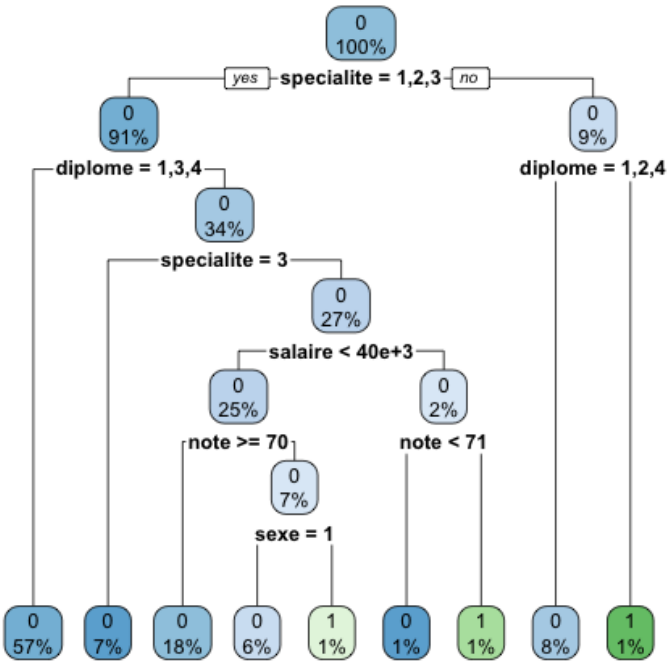
6 Evaluation and Comparison

Classification:

Decision tree (1):

In Decision tree (1) we divided the data set into training set with percentage (90%) and test set with percentage (10%).

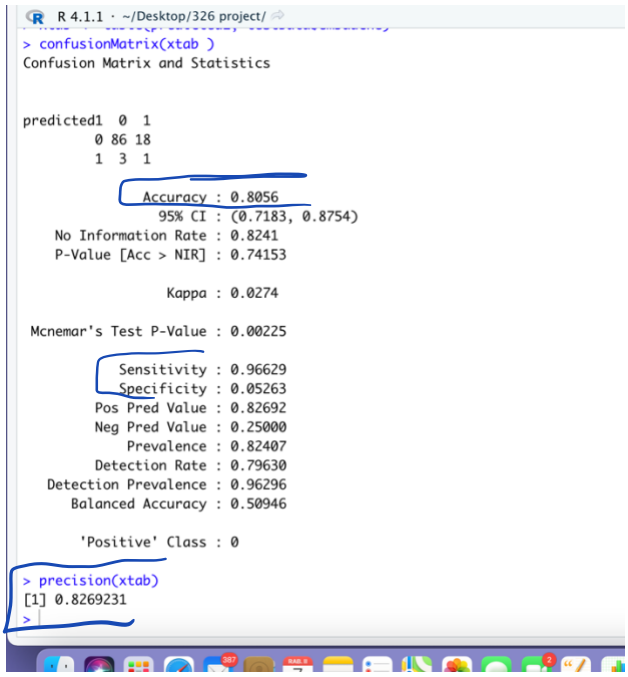
Decision tree (1)



Using rpart.plot we plot the Decision tree (1) it shows us that the root is specialite and that indicated that it has the highest information gain of all attributes.

The best/ most valuable attribute in the tree and is selected to split the data

Confusion Metrix for (1)

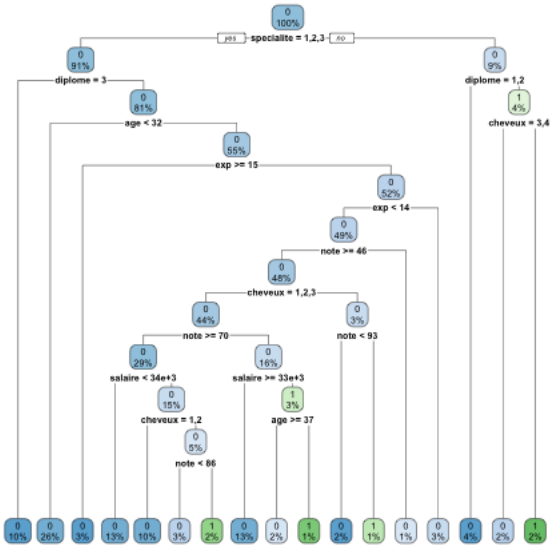


```
R 4.1.1 · ~/Desktop/326 project/
> confusionMatrix(xtab )
Confusion Matrix and Statistics


predicted1  0  1
         0 86 18
         1  3  1

               Accuracy : 0.8056
                 95% CI : (0.7183, 0.8754)
    No Information Rate : 0.8241
    P-Value [Acc > NIR] : 0.74153

                  Kappa : 0.0274

 Mcnemar's Test P-Value : 0.00225

            Sensitivity : 0.96629
            Specificity : 0.05263
         Pos Pred Value : 0.82692
         Neg Pred Value : 0.25000
             Prevalence : 0.82407
         Detection Rate : 0.79630
   Detection Prevalence : 0.96296
      Balanced Accuracy : 0.50946

       'Positive' Class : 0

> precision(xtab)
[1] 0.8269231
>
```

Decision tree (2): نفس أول وحدة

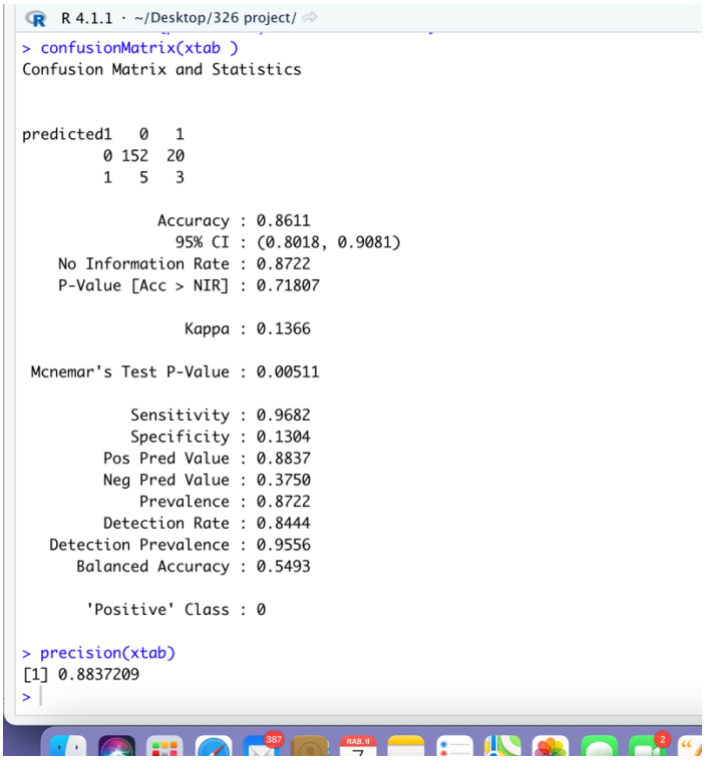
In Decision tree (2) we divided the data set into training set with percentage (80%) and test set with percentage (20%).

Decision tree (2)



Using rpart.plot we plot the Decision tree (2) it shows us that the root is also specialite and that indicated that it has the highest information gain of all attributes.

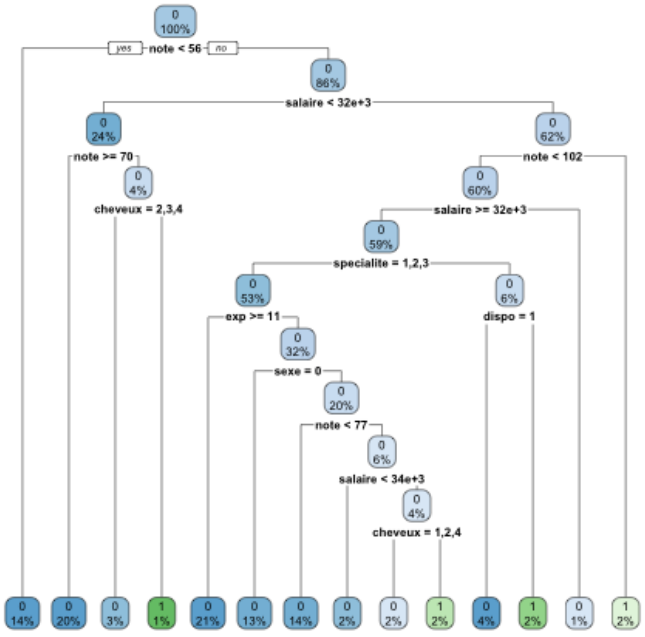
Confusion Metrix for (2)



Decision tree (3): نفس أول وحدة

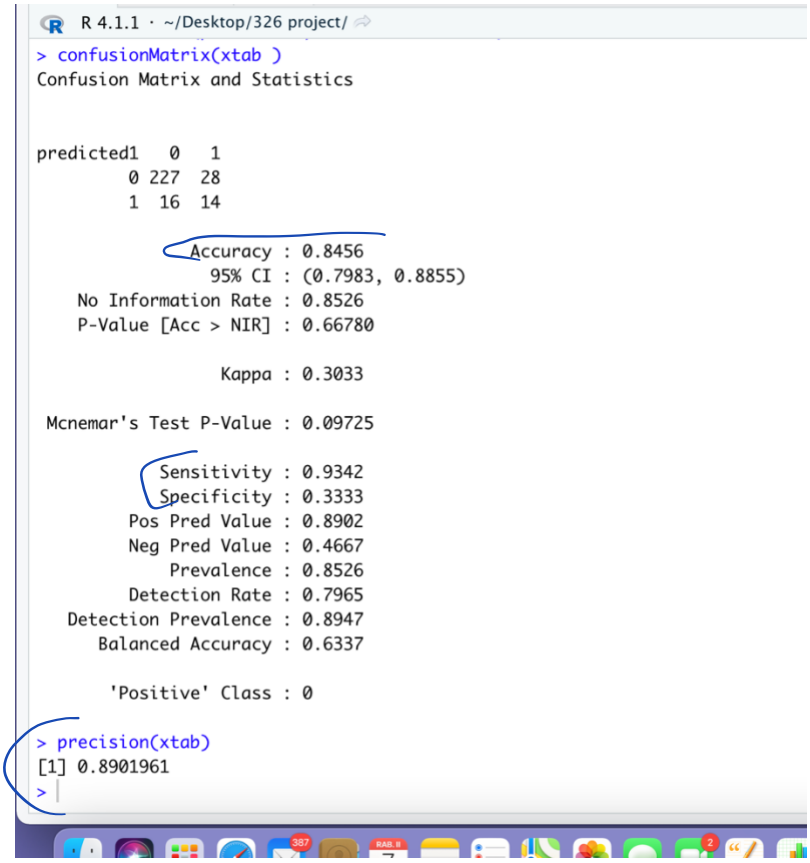
In Decision tree (3) we divided the data set into training set with percentage (70%) and test set with percentage (30%).

Decision tree (3)



Using rpart.plot we plot the Decision tree (3) it shows us that the root is note and that indicated that it has the highest information gain of all attributes.

Confusion Metrix for (3)



Evaluation matrix table

Decision trees	Decision tree (1): 90% - 10%	Decision tree (2) 80% - 20%	Decision tree (3) 70% - 30%
Accuracy	80%	86% highest accuracy	84%
precision	82%	88%	89%
sensitivity	96%	96%	93%
specificity	5%	13%	33%

Accuracy: number of classifications a model correctly predicts

Precision: Number of true positive divided by the total number of positive production

Sensitivity: predict true positives which is people that hired
Specificity: predict true negative which is people that's not hired

Clustering:

Size (1):

In size (1) we chose number of clusters to be k=2.

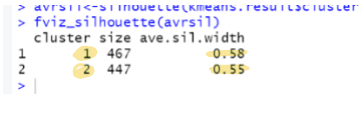
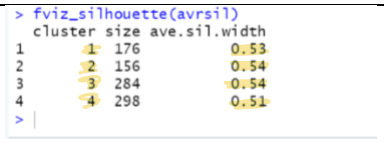
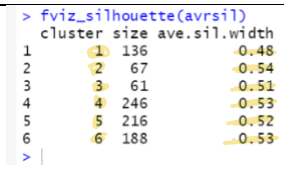
Size (2):

In size (2) we chose number of clusters to be k=4.

Size (3):

In size (3) we chose number of clusters to be k=6.

Clustering table

Clusters	K=2	K=4	K=6
Silhouette width for each cluster	 Figure2	 Figure4	 Figure6
Silhouette width for all clusters	average silhouette width=0.56	average silhouette width=0.53	average silhouette width=0.52
Visualization	Figure1	Figure3	Figure5

For size 1 (K=2)

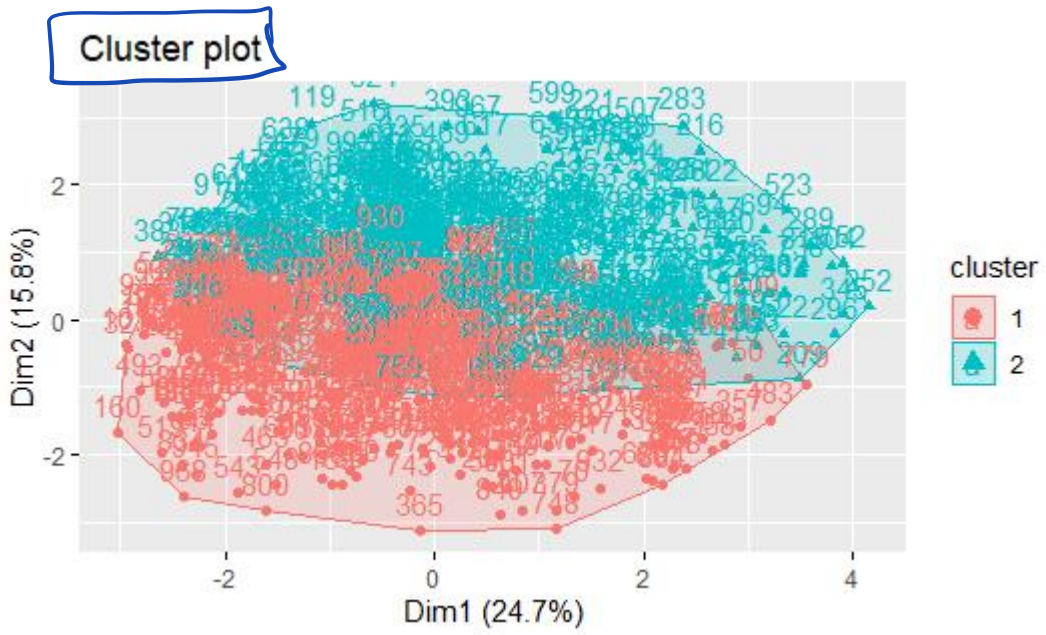


Figure1

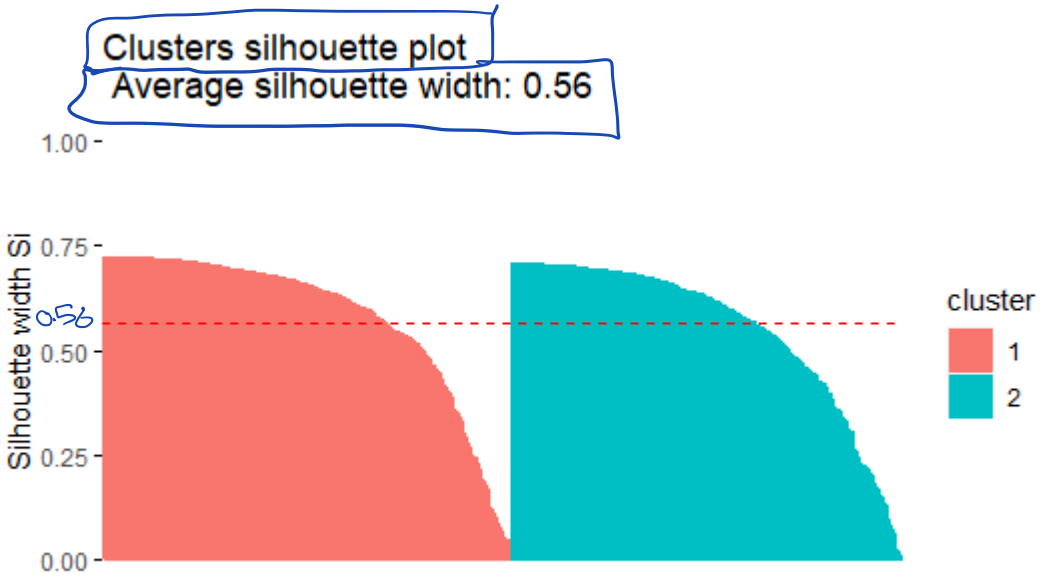


Figure2 K=2

For Size(2) (K=4)

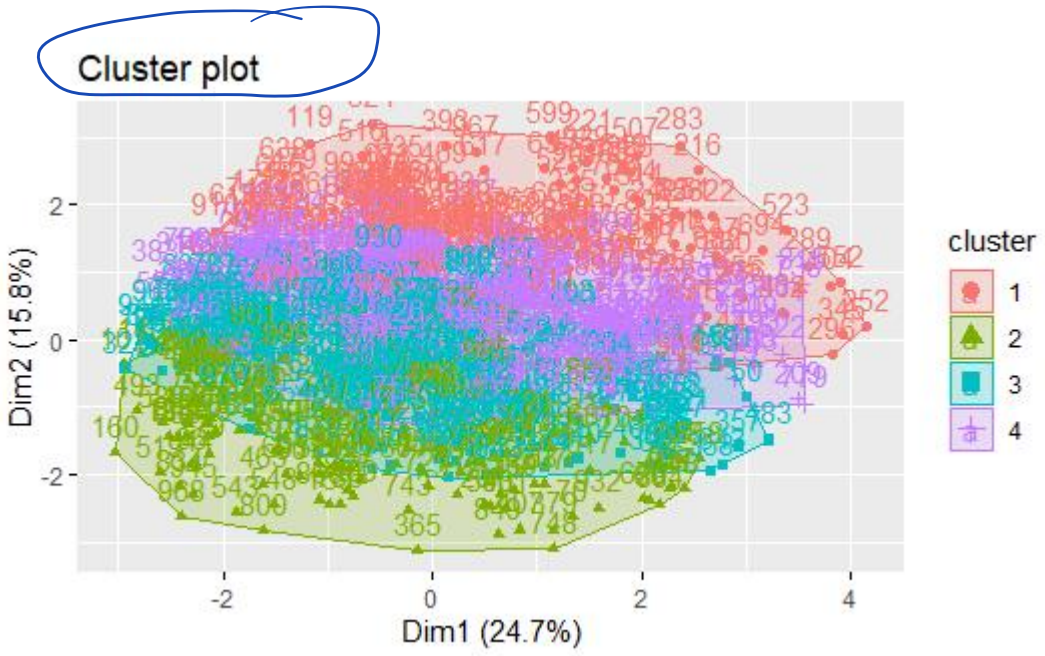


Figure3

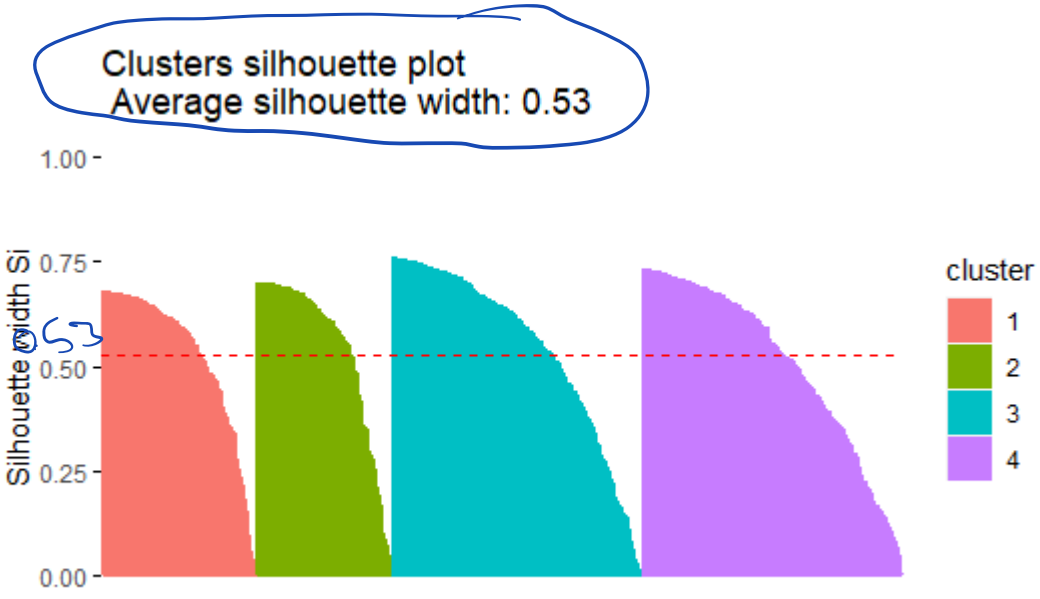


Figure4 K=4

For Size3 (K=6)

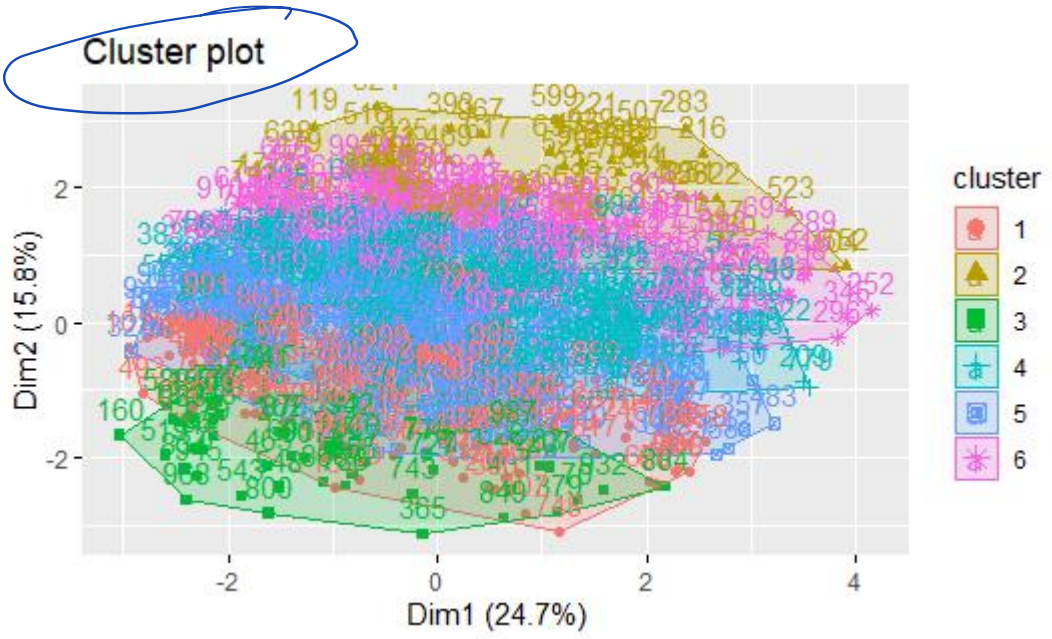


Figure5

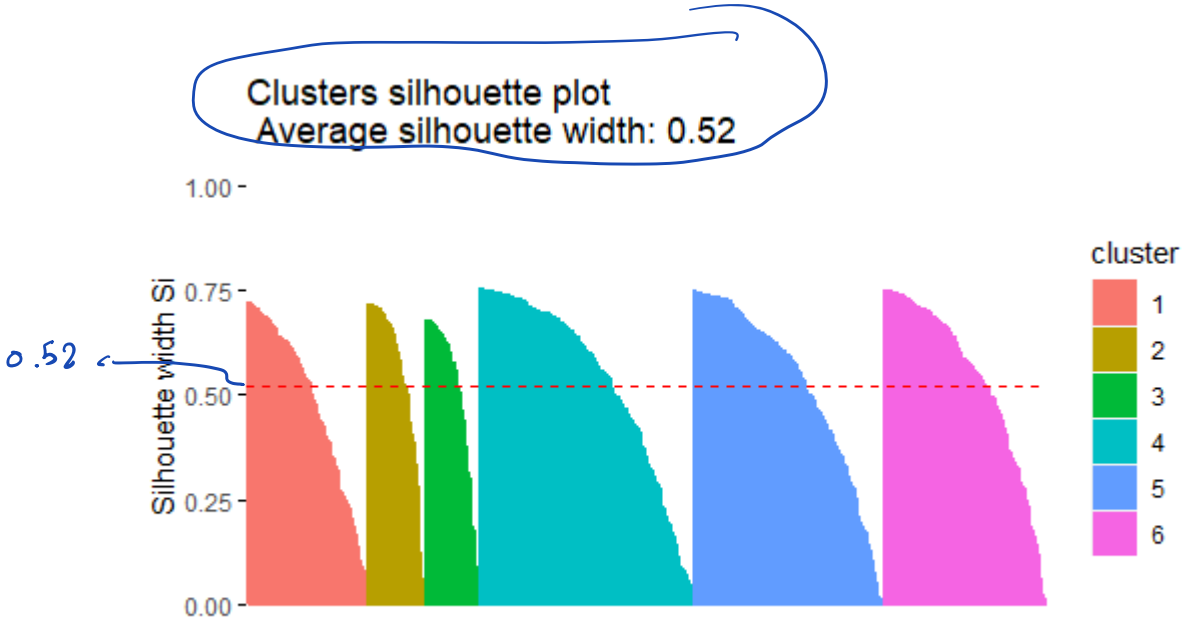


Figure 6 K=6

هنا زي الكلاسفكيشن، لازم نسوي evaluate عشان نختار مين افضل سايز عندنا واعطانا قيمة اقرب لل 1 ، استخدمنا طريقتين وكلهم اشاروا ان k=2 افضل يعني اننا نقسم الكلستر الى مجموعتين يعطينا نتايج افضل ويسمى k=2 في هذي الحاله optimal number

We used two ways to find the optimal number of clusters to make sure, the first way we used: fviz_nbclust using Silhouette method, and the second way is NbClust Validation, and in both ways they indicates k=2 to be the optimal number of cluster.

الطريقة الأولى

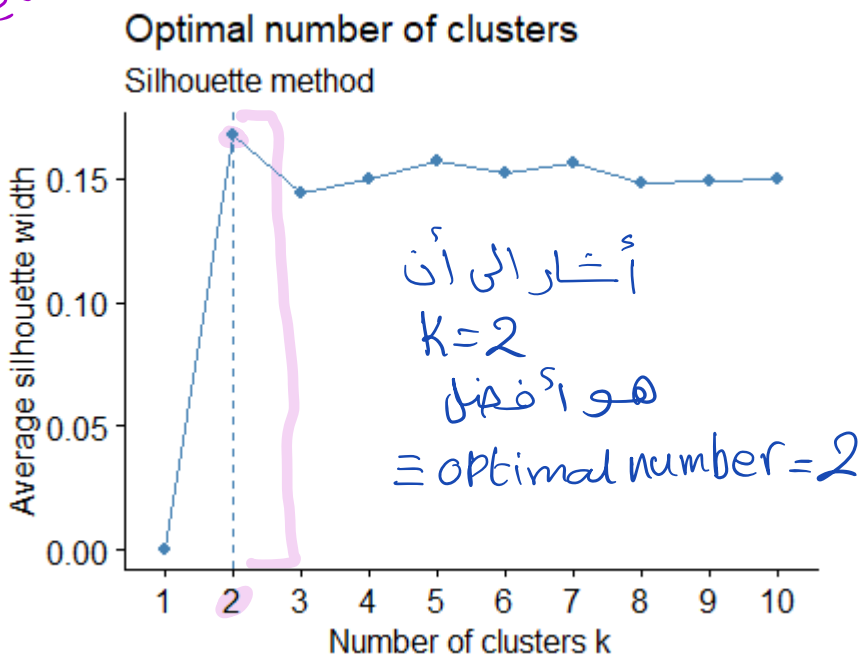
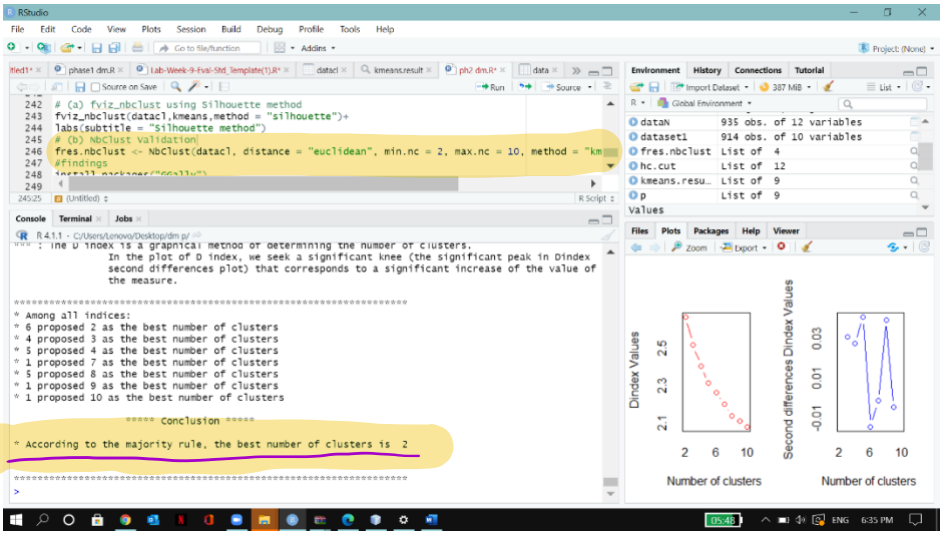


Figure7 The best number of k =2

الثانية



هنا كل واحدة فينا بتشرح الفايندنقز فكل نقطة فيها مهمة
لازم نفهم كل شيء

7 Findings

Classification:

First, we did Mining the data by using classification technique and used tree model, use rpart, predict, confusionMatrix, We applied tree algorithm on 3 different sizes: -training set 90% and testing set 10% -training set 80% and testing set 20% and 70% training 30% testing.

after calculating the accuracy for the 3 different sizes we conclude that the best result of accuracy was "86%" which belongs to (80% training -20% testing). That means most of the tuples was classified correctly based on our class label "embauche". From the decision tree it was split based on the "specialite" since it has the heights information gain, and it means that "specialite" has major effect on the class label "embauche" which indicate the candidates will be hired or not.

We used rpart method to plot our decision tree because it gave us more reasonable information than ctree, when we tried using ctree it only gave us one node so rpart was more understandable and suitable for our dataset.

Clustering:

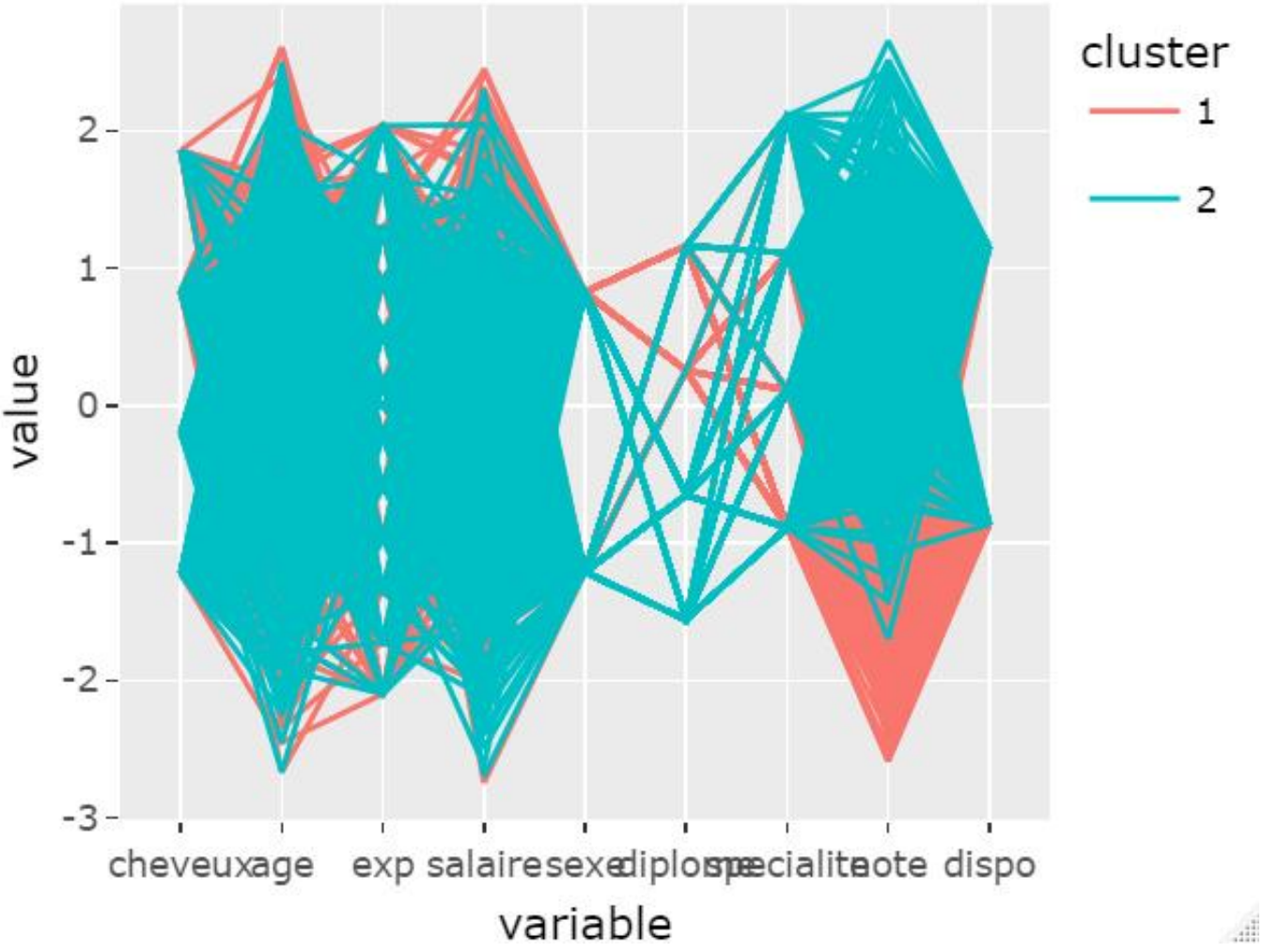
Secondly, we did Mining the data by using clustering technique by k means We want accurate results so we changed the number of clusters by choosing random k (2 ,4,6) to cluster the data based on their similarity. object in a cluster is similar to one another each object is assigned to the cluster to which it is most similar, based on the distance between the object and the cluster center for ex: in cluster one candidates with same or close years of experience and same age rang and same specialty...etc, are clustered together.

so when there is new candidate we can assign the candidate to one of our clusters. Also, we used fviz_cluster() to Visualize the clusters then we used the silhouette() method that gives the average of each cluster in the whole cluster .And we excepted the optimal number is 2 because it has the highest average among the number of k.

When the K=2 the average of cluster 0.56 which is the best average between our clusters, and in each cluster, silhouette are c1= 0.58 for first cluster and c2=0.55 for second cluster. When the K=4 the average of cluster 0.53, and in each cluster silhouette are c1= 0.53 for first cluster and c2= 0.54 for second cluster

c3=0.54for the third cluster c4=0.51 for the fourth cluster. When the K=6 the average of cluster 0.52, and in each cluster silhouette are c1= 0.48 for first cluster and c2=0.54 for second cluster, c3=0.51 for the third cluster c4=0.53 for the fourth cluster , c5=0.52for the fifth cluster c6=0.53 for the sixth .

we found that the optimal number of clusters (best number of clusters) in our data set is k=2 and the average width is 0.56



cluster 1: they have the highest in age, and have the lowest at note.

cluster 2: they have the highest in note, and have the lowest at salaires.

inconclusion:

we conclude that the classification is accurate in our dataset because the classification accuracy was 86% which is close to 100% which indicates that our classification is accurate and useful and in other hand when we used clustering, the optimal number of clusters was k=2 and silhouette average was 0.56 which is close to 1 which indicates that we **almost** have high quality cluster.

we reach 100% high quality when Silhouette average is =1

8 Code

Preprocessing:

```
data<-read.csv("data2.csv")
getwd()

#print No of objects
nrow(data)

# print NUM of features
ncol(data)

#print names of the features
names(data)

# print values of features of first five objects
head(data)
# print data type of features
str(data)


# print five number summary for numeric
#summary(data)
summary(data$X)
summary(data$age)
summary(data$exp)
summary(data$salaire)
summary(data$note)
summary(data$embauche)

#missing values
sum(is.na(data))
#replace missing data with average value

#missing values
sum(is.na(data))

#boxpolt outlier

```

```r
#boxpolt outlier

boxplot(data$age)
boxplot(data$exp)
boxplot(data$note)
boxplot(data$salaire)

# histogram
hist(data$age)
hist(data$exp)
hist(data$note)
hist(data$salaire)

barplot(table(data$cheveux))
barplot(table(data$date))
barplot(table(data$diplome))
barplot(table(data$specialite))

# scatter plot
plot(x = data$age, y = data$exp,
     xlab = "Age",
     ylab = "exp",
     main = "Age vs exp"
)

#data preprpccising
#removing outliers
library(outliers)

#data$exp[which(data$exp>420)] <- c(data$exp[which(data&exp>420)]*2)
boxplot(data$exp)
boxplot(data$exp)$out
outliers <- boxplot(data$exp, plot=FALSE)$out
print(outliers)
data[which(data$exp %in% outliers),]
data <- data[-which(data$exp %in% outliers),]
```

```r
data[which(data$exp %in% outliers),]
data <- data[-which(data$exp %in% outliers),]
boxplot(data$exp)

boxplot(data$age)
boxplot(data$age)$out
outliers1 <- boxplot(data$age, plot=FALSE)$out
print(outliers1)
data[which(data$age %in% outliers1),]
data <- data[-which(data$age %in% outliers1),]
boxplot(data$age)

boxplot(data$salaire)
boxplot(data$salaire)$out
outliers2 <- boxplot(data$salaire, plot=FALSE)$out
print(outliers2)
data[which(data$salaire %in% outliers2),]
data <- data[-which(data$salaire %in% outliers2),]
boxplot(data$salaire)

boxplot(data$note)
boxplot(data$note)$out
outliers3 <- boxplot(data$note, plot=FALSE)$out
print(outliers3)
data[which(data$note %in% outliers3),]
data <- data[-which(data$note %in% outliers3),]
boxplot(data$note)



#replace missing data with average value
data$date=ifelse(is.na(data$date),ave(data$date,FUN = function(x)mean(x,na.rm=TRUE)),data$date)
data$cheveux=ifelse(is.na(data$cheveux),ave(data$cheveux,FUN = function(x)mean(x,na.rm=TRUE)),data$cheveu
data$age=ifelse(is.na(data$age),ave(data$age,FUN = function(x)mean(x,na.rm=TRUE)),data$age)

data$exp=ifelse(is.na(data$exp),ave(data$exp,FUN = function(x)mean(x,na.rm=TRUE)),data$exp)
```

```r

data$exp=ifelse(is.na(data$exp),ave(data$exp,FUN = function(x)mean(x,na.rm=TRUE)),data$exp)
data$salaire=ifelse(is.na(data$salaire),ave(data$salaire,FUN = function(x)mean(x,na.rm=TRUE)),data$salaire)
data$sexe=ifelse(is.na(data$sexe),ave(data$sexe,FUN = function(x)mean(x,na.rm=TRUE)),data$sexe)
data$diplome=ifelse(is.na(data$diplome),ave(data$diplome,FUN = function(x)mean(x,na.rm=TRUE)),data$diplome)
data$specialite=ifelse(is.na(data$specialite),ave(data$specialite,FUN = function(x)mean(x,na.rm=TRUE)),data$specialite)
data$note=ifelse(is.na(data$note),ave(data$note,FUN = function(x)mean(x,na.rm=TRUE)),data$note)
sum(is.na(data))

# Normalization of features

str(data)
normalize<- function(x){(x-min(x))/(max(x)-min(x))}

dataN = data
dataN$age <-normalize(data$age)
dataN$salaire <-normalize(data$salaire)
dataN$exp <- normalize(data$exp)
dataN$note <- normalize(data$note)

str(dataN)

#########
#ENCODING

data$dispo = factor(data$dispo, levels = c("non", "oui"), labels = c(0, 1))

data$cheveux = factor(data$cheveux, levels = c("chatain", "brun","blond" ,"roux"), labels = c(1,2,3,4))

data$diplome = factor(data$diplome, levels = c("doctorat", "master","bac" ,"licence"), labels = c(1,2,3,4))
data$specialite = factor(data$specialite, levels = c("geologie", "forage","detective" ,"archeologie"), labels = c(1,2,3,4))
data$sexe = factor(data$sexe, levels = c("F", "M"), labels = c(0, 1))
str(data)


data [! is.na(data$embauche),]
```

a. **classification** → شرح كل سطر كود في الكومنت اللي فوقه

```
########classification########
library(caret)
library(party)
library(rpart)
library(rpart.plot)



#random number generator for objects that can be reproduced
set.seed(1234)

######## dicision tree 1 (90% -10% )###########

#partition data to 90% to be training set
#nrow is a method to return the number of rows of data
ind <- sample(2, nrow(data) , replace = TRUE , prob = c(0.9 , 0.1))
#store training set value 90% in variable trainData
trainData <- data[ind == 1,]
#store testing set value 10% in variable trainData
testData <-  data[ind == 2,]

#creating/building decision tree by saving all attributes (columns) data in variable "model1"
#rpart is a machine learning library that is used to build the decesion tree
model1<- rpart(embauche ~  cheveux + age + exp + salaire + sexe + diplome + specialite + note + dispo, data = trainData, meth

# plot decision tree for model1 , extra = 100 it display the number of percentage of observation in node of tree
rpart.plot(model1, extra = 100)

# test decision tree , predict the values based on the input we store the predected values in variable predectid1
predicted1 <- predict(model1, newdata = testData, type = "class")



#confusion metrix
xtab <- table(predicted1, testData$embauche)
confusionMatrix(xtab )
```

```r
#confusion metrix
xtab <- table(predicted1, testData$embauche)
confusionMatrix(xtab )
precision(xtab)

########dicision tree 2 (80% -20% )###########
#partition data to 50% to be training set
#nrow is a method to return the number of rows of data
ind <- sample(2, nrow(data) , replace = TRUE , prob = c(0.8 , 0.2))
#store training set value 90% in variable trainData
trainData <- data[ind == 1,]
#store testing set value 10% in variable trainData
testData <-  data[ind == 2,]


# Create the decision tree
model1<- rpart(embauche ~  cheveux + age + exp + salaire + sexe + diplome + specialite + note + dispo, data = trainData, method='cla

# plot the decision tree
rpart.plot(model1, extra = 100)

# test the decision tree
predicted1 <- predict(model1, newdata = testData, type = "class")

#confusion metrix
xtab <- table(predicted1, testData$embauche)
confusionMatrix(xtab )
precision(xtab)

######## dicision tree 3 (70% -30% )###########
#partition data to 70% to be training set
#nrow is a method to return the number of rows of data
ind <- sample(2, nrow(data) , replace = TRE , prob = c(0.7 , 0.3))
#store training set value 90% in variable trainData
trainData <- data[ind == 1,]
#store testing set value 10% in variable trainData
```

Evaluation

Evaluation

هنا نشرح كل كود نفس اللي فوق

فقط ضروري اللي تغيروا

```r
xtab <- table(predicted1, testData$embauche)
confusionMatrix(xtab )
precision(xtab)

######## dicision tree 3 (70% -30% )###########
#partition data to 70% to be training set
#nrow is a method to return the number of rows of data
ind <- sample(2, nrow(data) , replace = TRUE , prob = c(0.7 , 0.3))
#store training set value 90% in variable trainData
trainData <- data[ind == 1,]
#store testing set value 10% in variable trainData
testData <-  data[ind == 2,]


# Create decision tree, Using training data set to determine the classification model
model1<- rpart(embauche ~ cheveux + age + exp + salaire + sexe + diplome + specialite + note + dispo, data = trainData, method='class')

# plot decision tree
rpart.plot(model1, extra = 100)

# test decision tree
predicted1 <- predict(model1, newdata = testData, type = "class")

#confusion metrix
xtab <- table(predicted1, testData$embauche)
confusionMatrix(xtab )
precision(xtab)
```

Evaluation

b. clustering

```r
#clustring
datacl=data
datacl$i..<-NULL
datacl$date<-NULL
datacl$embauche<-NULL

datacl$diplome<-as.numeric(as.character(datacl$diplome))
datacl$sexe<-as.numeric(as.character(datacl$sexe))
datacl$cheveux<-as.numeric(as.character(datacl$cheveux))
datacl$specialite<-as.numeric(as.character(datacl$specialite))
datacl$dispo<-as.numeric(as.character(datacl$dispo))
datacl$specialite<-as.numeric(as.character(datacl$specialite))
#cheek if all attributes are numeric
sapply(datacl,class)
sum(is.na(datacl))
datacl<-(na.omit(datacl))
#k-means set seed to random number to make result reproducible
set.seed(8953)
datacl <- scale(datacl)

library(factoextra)
library(cluster)
library(NbClust)
# Split the dataset into 2clusters using k-mean method
kmeans.result <- kmeans(datacl,2)

# Print the result
kmeans.result

# Visualize clusters
fviz_cluster(kmeans.result, data = datacl)
```

size 1

```r

#average for each cluster
avrsil<-silhouette(kmeans.result$cluster,dist(datacl))
fviz_silhouette(avrsil)

# Split the dataset into 4 clusters using k-mean method
kmeans.result <- kmeans(datacl,4)

# Print the result
kmeans.result

# Visualize clusters
datacl<-(na.omit(datacl))
fviz_cluster(kmeans.result, data = datacl)

#average for each cluster
avrsil<-silhouette(kmeans.result$cluster,dist(datacl))
fviz_silhouette(avrsil)


# Split the dataset into 6 clusters using k-mean method
kmeans.result <- kmeans(datacl,6)

# Print the result
kmeans.result

# Visualize clusters
datacl<-(na.omit(datacl))
fviz_cluster(kmeans.result, data = datacl)
```

size 2

size 3

```r
#average for each cluster
avrsil<-silhouette(kmeans.result$cluster,dist(datacl))
fviz_silhouette(avrsil)

#Get the optimal number of clusters

# (a) fviz_nbclust using Silhouette method
fviz_nbclust(datacl,kmeans,method = "silhouette")+
labs(subtitle = "Silhouette method")
# (b) NbClust Validation
fres.nbclust <- NbClust(datacl, distance = "euclidean", min.nc = 2, max.nc = 10, method = "kmeans",index = "all")
#findings
#install.packages("GGally")
#install.packages("plotly")
library(GGally)
library(plotly)
dataset1<-as.data.frame(datacl)
dataset1$cluster<-as.factor(kmeans.result$cluster)
p<-ggparcoord(data=dataset1, columns=c(1,2,3,4,5,6,7,8,9),groupColumn = "cluster", scale = "std")
labs(x="d",y=ggplotly(p))
```

Evaluation

9 References

Kaggle website https://www.kaggle.com/bryanb/applicants-for-a-gold-digger-position

Use IEEE format https://libraryguides.vu.edu.au/ieeereferencing/gettingstarted

You can use any resource for reference generating such as https://www.citethisforme.com/citation-generator/ieee

https://towardsdatascience.com/clustering-analysis-in-r-using-k-means-73eca4fb7967

10 Tasks Distribution

ID	Name	Responsibilities
439201336	Seba alahmadi	Data mining technique Clustering Report
441201306	Noura Alsultan	Data mining technique Classification Report
441201159	Nouf alfulaij	Data mining technique Clustering Report
437201303	Reem Almutairi	Data mining technique Classification Report