YUFEI LI (2204273)
AMAN JEAN-JACQUES (2101768)

PROJET STATISTIQUES DESCRIPTIVES 2
Fraudes ou absence de fraude sur des données bancaires

Chargé d'enseignement : Mickael Lallouche

1) Introduction

L'une des premières étapes a consisté à consulter les données du fichier train.csv.
Nous avons très vite été freiné par la possibilité de lire toutes les données sur le logiciel Microsoft Excel à cause de leur trop grande volumétrie.
Cependant, grâce aux descriptions sur le jeu de données fournis par le document accompagnant le dossier de Statistiques descriptives 2, les différentes explications de nos professeurs et nos recherches effectuées en parallèles, nous avons peu à peu cerné les objectifs, les différentes méthodes utiles et avons commencé à nous organiser afin de travailler dans de bonnes conditions.

Etant donnée les capacités limitées de nos machines informatiques, nous avons opté pour une solution répondant à la problématique concernant la volumétrie des données : l'utilisation du package « disk.frame ».

Le fichier train est composé d'un tableau à deux dimensions comportant 4.262.956 observations et 12 variables : id, step, type, amount, nameOrig, oldbalanceOrg, newbalanceOrig, nameDest, oldbalanceDest, newbalanceDest, isFraud, isFlaggedFraud.

```
> str(train)
'data.frame':   4262956 obs. of  12 variables:
 $ id             : int  1 2 5 7 8 9 12 13 14 15 ...
 $ step           : int  1 1 1 1 1 1 1 1 1 1 ...
 $ type           : chr  "PAYMENT" "PAYMENT" "PAYMENT" "PAYMENT" ...
 $ amount         : num  9840 1864 11668 7108 7862 ...
 $ nameOrig       : chr  "C1231006815" "C1666544295" "C2048537720" "C154988899" ...
 $ oldbalanceOrg  : num  170136 21249 41554 183195 176087 ...
 $ newbalanceOrig: num  160296 19385 29886 176087 168226 ...
 $ nameDest       : chr  "M1979787155" "M2044282225" "M1230701703" "M408069119" ...
 $ oldbalanceDest: num  0 0 0 0 0 0 0 0 0 0 ...
 $ newbalanceDest: num  0 0 0 0 0 0 0 0 0 0 ...
 $ isFraud        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ isFlaggedFraud: int  0 0 0 0 0 0 0 0 0 0 ...
```

Data Preview:

id (double)	step (double)	type (character)	amount (double)	nameOrig (character)	oldbalanceOrg (double)	newbalanceOrig (double)	nameDest (character)	oldbalanceDest (double)	newbalanceDest (double)
1	1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0	0.00
2	1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0	0.00
5	1	PAYMENT	11668.14	C2048537720	41554.00	29885.86	M1230701703	0	0.00
7	1	PAYMENT	7107.77	C154988899	183195.00	176087.23	M408069119	0	0.00
8	1	PAYMENT	7861.64	C1912850431	176087.23	168225.59	M633326333	0	0.00
9	1	PAYMENT	4024.36	C1265012928	2671.00	0.00	M1176932104	0	0.00
12	1	PAYMENT	3099.97	C249177573	20771.00	17671.03	M2096539129	0	0.00
13	1	PAYMENT	2560.74	C1648232591	5070.00	2509.26	M972865270	0	0.00
14	1	PAYMENT	11633.76	C1716932897	10127.00	0.00	M801569151	0	0.00
15	1	PAYMENT	4098.78	C1026483832	503264.00	499165.22	M1635378213	0	0.00
18	1	PAYMENT	1157.86	C1237762639	21156.00	19998.14	M1877062907	0	0.00
19	1	PAYMENT	671.64	C2033524545	15123.00	14451.36	M473053293	0	0.00

Previewing first 50 entries.

2) Recodage des données

Le fichier train.csv contient plus de 4.000.000 de lignes. Afin d'alléger les valeurs afin d'explorer les données, de les analyser et de travailler sur les prédictions, nous avons décider de recourir à plusieurs étapes :

- La première a consisté à réduire au maximum et de manière compréhensible la taille des caractères contenus dans la variable « Type ». Ainsi, nous avons fait le choix de modifier le

nom des chaînes de caractère en valeurs plus petite. Exemple : « PAYMENT » = « PAY » ou encore « DEBIT » = « DEB ». Il en a été de même concernant le renommage des intitulés de variables. Exemple : « oldbalanceOrg » = « oldbalOrig »

- La deuxième étape a consisté à arrondir les valeurs contenues dans les variables « amount », « oldbalanceOrg », « newbalanceOrig », « oldbalanceDest », « newbalanceDest ».

Ainsi, nous avons créé un dataframe prenant en compte ces modifications : df_projetdatas

```
id <- projetstats$id
step<-projetstats$step
type <- factor(projetstats$type, levels=c(unique(projetstats$type)), labels=c("pay", "tra","deb","casi",'caso'))
amount<-round(projetstats$amount,0) #On arrondi la variable amount
naOrig<-projetstats$nameOrig
oldbalOrig <- round(projetstats$oldbalanceOrg,0) #On arrondi la variable oldbalanceOrg
nbOrig<- round(projetstats$newbalanceOrig,0) #On arrondi la variable newbalanceOrg
nameDest<-projetstats$nameDest
oldbalDest<-round(projetstats$oldbalanceDest,0) #On arrondi la variable oldbalanceDest
nwbalDest<-round(projetstats$newbalanceDest,0) #On arrondi la variable newbalanceDest
isFraud<-factor(projetstats$isFraud,levels=c(unique(projetstats$isFraud)), labels=c("nofraud", "fraud"))
isFlag<-factor(projetstats$isFlaggedFraud,levels=c(unique(projetstats$isFlaggedFraud)), labels=c("non_flag", "flag"))

df_projetdatas<- data.frame(id,step,type,amount,naOrig,oldbalOrig,nbOrig,nameDest,oldbalDest,nwbalDest,isFraud,isFlag)
```

A partir de cette dataframe df_projetdatas, nous avons constitué les trois échantillons suivants :

- df_train (60% des données de df_projetdatas)
- df_validation (20% des données de df_projetdatas)
- df_test (20% des données de df_projetdatas)

Ces trois échantillons sont réalisés de manière aléatoire (sample_frac) et sans réplications (replace = FALSE).

```
#FALSE permet de ne pas reutiliser les donnees dans les autres echantillons
#Creation de la dataframe train (60% des donnees de df_projetstats en echantillon aleatoire)
df_train <- collect(sample_frac(df_projetdatas, 0.6), replace = FALSE)
```

Afin d'indiquer sur les id des éléments retenus, nous avons décidé de filtrer les lignes correspondantes à nos données (contenues dans df_train, df_validation et df_test), de les stocker des variables et de les ordonner : df_trainIDcorrespondant, df_validationIDcorrespondant, df_testIDcorrespondant).

```
#identifiants des lignes utilisees pour df_train
df_trainIDcorrespondant <- df_train[,1]
df_trainIDcorrespondant <-sort(df_trainIDcorrespondant)
df_trainIDcorrespondant
```

Les colonnes id, ne nous servant pas dans le cadre du traitement de données, nous avons choisi de les enlever.

```
#On enleve la colonne ID car elle ne doit pas entrer en ligne de compte dans nos traitements,
df_train2 <- select(df_train,step,type,amount,naOrig,oldbalOrig,nbOrig,nameDest,oldbalDest,nwbalDest,isFraud,isFlag)
```

Les colonnes isFraud et isFlaggedFraud ne sont pas réellement des entiers. Ce sont des variables qualitatives nominales. Ainsi, nous les transformant en factor afin de le prendre en compte dans nos traitements pour df_train, df_validation et df_test.

```
df_train2$isFraud <- factor(df_train2$isFraud)
df_train2$isFlag <- factor(df_train2$isFlag)
```

3) Statistiques descriptives et analyse des données

Nous avons commencé par consulter les informations de base sur les données.

```
> summary(df_train2)
      step          type           amount           oldbalOrig          nbOrig          oldbalDest
 Min.   :  1.0   pay :864692   Min.   :       0   Min.   :       0   Min.   :       0   Min.   :       0
 1st Qu.:155.0   tra :214359   1st Qu.:   13380   1st Qu.:       0   1st Qu.:       0   1st Qu.:       0
 Median :239.0   deb : 16663   Median :   74916   Median :   14215   Median :       0   Median :   132502
 Mean   :243.2   casi:900101   Mean   :  179737   Mean   :  833624   Mean   :  854883   Mean   :  1098935
 3rd Qu.:334.0   caso:561959   3rd Qu.:  208821   3rd Qu.:  107349   3rd Qu.:  144260   3rd Qu.:   942284
 Max.   :743.0                 Max.   :92445517   Max.   :59585040   Max.   :49585040   Max.   :356015889
   nwbalDest           isFraud            isFlag
 Min.   :        0   nofraud:2554505   non_flag:2557768
 1st Qu.:        0   fraud  :   3269   flag    :      6
 Median :   214527
 Mean   :  1223340
 3rd Qu.:  1111067
 Max.   :356179279
```

Nous avons calculé les écart types (ici, calcul de l'écart-types des variables contenus dans df_train2)

amount : 610966
oldbalorig : 2888151
newbalorig : 2923890
oldbalDest : 3356985
newbalDest : 3634943

Dans le livrable du projet, il est indiqué que les transactions frauduleuses sont signalées lorsque le montant de transfert est supérieur a 200.000.
Nous avons donc effectué une vérification des lignes correspondantes.

verif_isFlag2 <-filter(df_projetdatas, isFlag=="flag")

verif_isFlag2_sup200 <- filter(df_projetdatas, amount > 200000 & isFlag=="flag")

verif_isFlag2_projetdata <- filter(df_projetdatas, amount > 200000)

'on a plus de 1.000.000 de lignes dans les donnees principales dont le montant de transfert est superieur a 200.000$ et qui n ont pas ete signale (isflag = noflag)'

```
> verif_isFlag2_projetdata <-  filter(df_projetdatas, amount > 200000)
> verif_isFlag2_projetdata
    id step type  amount      naOrig oldbalOrig  nbOrig    nameDest oldbalDest nwbalDest isFraud   isFlag
1   20    1  tra  215310 C1670993182        705       0 C1100439041      22425         0 nofraud non_flag
2   25    1  tra  311686 C1984094095      10835       0  C932583850       6267   2719173 nofraud non_flag
3   83    1  tra  224607  C873175411          0       0  C766572210     354679         0 nofraud non_flag
4   85    1  tra  379856 C1449772539          0       0 C1590550415     900180  19169205 nofraud non_flag
5   87    1  tra  554027 C1603696865          0       0  C766572210     579286         0 nofraud non_flag
6   90    1  tra 1429051 C1520267010          0       0 C1590550415    2041544  19169205 nofraud non_flag
7   91    1  tra  358832  C908084672          0       0  C392292416     474385   3420103 nofraud non_flag
8   92    1  tra  367768  C288306765          0       0 C1359044626     370763     16518 nofraud non_flag
9   93    1  tra  209711 C1556867940          0       0 C1509514333     399215      2415 nofraud non_flag
10  95    1  tra 1724887 C1495608502          0       0 C1590550415    3470595  19169205 nofraud non_flag
11  97    1  tra  581294  C843299092          0       0 C1590550415    5195482  19169205 nofraud non_flag
12 113    1 casi  212228 C1896074070          0       0  C401424608     429747   1178808 nofraud non_flag
13 124    1  tra  330757 C1494346128     103657       0  C564160838      79676   1254956 nofraud non_flag
14 162    1 casi  289646 C1446001495          0       0 C1023714065     871443   1412484 nofraud non_flag
15 163    1 casi  267149 C1261044180          0       0  C401424608     641975   1178808 nofraud non_flag
16 166    1 casi  344464  C793293778          0       0  C766572210    1133313         0 nofraud non_flag
17 178    1 casi  220691 C1123559518          0       0 C1590550415    6093091  19169205 nofraud non_flag
18 188    1 casi  296753  C589363823          0       0 C1531333864     404815     55975 nofraud non_flag
19 190    1 casi  365510 C1299327689          0       0  C564160838     564573   1254956 nofraud non_flag
20 196    1 casi  210370 C2121995675          0       0 C1170794006    1442298     22191 nofraud non_flag
21 199    1 casi  338767  C691691381          0       0  C453211571     544481   3461666 nofraud non_flag
22 237    1  tra  480223  C201677908      11110       0  C451111351    1293741   3940085 nofraud non_flag
23 271    1 casi  280878 C1544614339       2189       0 C1297685781     462914     16997 nofraud non_flag
24 280    1 casi  369989 C1936550492       9516       0 C1789550256     518243   4619799 nofraud non_flag
25 281    1 casi  215338  C594651850          0       0   C75457651     285756     31470 nofraud non_flag
26 284    1 casi  262392 C2069500590          0       0 C2083562754     457286   1186557 nofraud non_flag
27 286    1 casi  202625  C452364286          0       0  C985934102    1056495    971419 nofraud non_flag
28 289    1 casi  498961 C1957078537          0       0 C1360767589     608612   2107965 nofraud non_flag
29 297    1 casi  302693  C140404585      18455       0 C1688019098     306239     97264 nofraud non_flag
30 309    1 casi  410202 C1279740095      21448       0 C1750905143      61613    424250 nofraud non_flag
31 356    1  tra  211076 C1540894701          0       0  C564160838     930083   1254956 nofraud non_flag
32 358    1  tra  473500 C1198197478          0       0  C453211571     883248   3461666 nofraud non_flag
33 359    1  tra 1538200  C476579021          0       0 C1590550415    6977445  19169205 nofraud non_flag
34 360    1  tra 2421578  C106297322          0       0 C1590550415    8515646  19169205 nofraud non_flag
35 363    1  tra 1457214  C396918327          0       0 C1590550415   10937224  19169205 nofraud non_flag
36 365    1  tra  445039  C547441493          0       0 C1531333864     802245     55975 nofraud non_flag
37 366    1  tra 1123207  C967677821          0       0  C451111351    1773963   3940085 nofraud non_flag
38 373    1  tra  438437  C977160959          0       0  C248609774     740675   6453431 nofraud non_flag
39 374    1  tra  928723 C1563053805          0       0  C985934102    1259120    971419 nofraud non_flag
40 376    1  tra 2545478 C1057507014          0       0 C1590550415   12394437  19169205 nofraud non_flag
41 377    1  tra 2061083 C2007599722          0       0 C1590550415   14939915  19169205 nofraud non_flag
42 379    1  tra  635508   C65080774          0       0  C747464370     834457   1567435 nofraud non_flag
43 380    1  tra  848232 C2116179210          0       0 C1170794006    4114920     22191 nofraud non_flag
44 381    1  tra  739113 C1172535934          0       0  C564160838    1141159   1254956 nofraud non_flag
45 382    1  tra  324398 C1648700617          0       0  C932583850     373059   2719173 nofraud non_flag
46 384    1  tra  955855   C94830685          0       0  C248609774    1179113   6453431 nofraud non_flag
47 389    1 casi  373068 C1047934137      20034       0 C1286084959    1427961   2107778 nofraud non_flag
48 391    1 caso  228452 C1614133563     143236  371688 C2083562754     719678   1186557 nofraud non_flag
49 398    1 caso  349641 C1493042329    1023112 1372752  C909295153     360951   5602235 nofraud non_flag
50 402    1 caso  311024 C1078262677    2306780 2617803  C766572210    1477777         0 nofraud non_flag
51 405    1 caso  220431 C1543518287    2998376 3218807  C451111351    2897170   3940085 nofraud non_flag
52 407    1 caso  231882  C464872674    3401668 3633550 C1509514333     894142      2415 nofraud non_flag
53 412    1 caso  764773  C482307698    3946209 4710982  C985934102    3056434    971419 nofraud non_flag
54 420    1 caso  257348 C1278839936    5226912 5484260  C985934102    2291661    971419 nofraud non_flag
55 421    1 caso  201074 C2143739483    5484260 5685334  C401424608     537767   1178808 nofraud non_flag
56 424    1 caso  227335 C1621254922    6073129 6300464  C564160838    1817946   1254956 nofraud non_flag
57 431    1 caso  355294 C1860886124    6962605 7317899  C747464370    1469965   1567435 nofraud non_flag
58 433    1 caso  349506  C173791568    7330236 7679741 C1590550415   17000998  19169205 nofraud non_flag
59 434    1 caso  285185 C1293462056    7679741 7964927 C1335050193     451065    353533 nofraud non_flag
60 436    1 caso  214851 C2002174925    8097880 8312732   C33524623     939719   1517262 nofraud non_flag
61 443    1 caso  313374 C1552870927      20140  333514  C985934102    2034313    971419 nofraud non_flag
62 461    1 caso  223555  C373097727    1908051 2131607  C564160838    1462657   1254956 nofraud non_flag
63 464    1 caso  222711 C2123533871    2419069 2641780 C1590550415   16651492  19169205 nofraud non_flag
64 466    1 caso  628719 C2022689531    2726761 3355480 C1359044626    1484769     16518 nofraud non_flag
65 475    1 caso  345348  C538667887    4505975 4851323  C240650537     355418         0 nofraud non_flag
66 476    1 caso  355500 C1967496309    4851323 5206823  C766572210    1166753         0 nofraud non_flag
67 479    1 caso  259753 C1045731788    5418633 5678386 C1209271652     260112         0 nofraud non_flag
68 480    1 caso  234094 C1739267143    5678386 5912481  C985934102    1278333    971419 nofraud non_flag
69 486    1 caso  277807  C212963786    6810068 7087875 C1531333864     907460     55975 nofraud non_flag
70 493    1 caso  236748 C1747053097    7622040 7858788  C757108857     390963         0 nofraud non_flag
71 497    1 caso  289273  C312168418    8125618 8414891   C75457651     413977     31470 nofraud non_flag
72 510    1  tra  420532  C582300198       5386       0 C1971489295     614566         0 nofraud non_flag
73 524    1  tra  276461 C1871680329        595       0 C1360767589    1105242   2107965 nofraud non_flag
74 553    1 casi  562904   C24039137       9045       0   C33524623     564520   1517262 nofraud non_flag
75 558    1 casi  227478 C1394010463      25744       0 C1590550415   16428781  19169205 nofraud non_flag
76 601    1 caso  414239 C1523649562     529390  943628 C1359044626     808520     16518 nofraud non_flag
77 607    1 caso  336828 C1244880808    1479063 1815890  C476402209     401494     51513 nofraud non_flag
78 611    1 caso  314134  C464649704    2245808 2559942 C1509514333     570802      2415 nofraud non_flag
79 612    1 caso  206406  C367967231    2559942 2766348  C453211571    1221424   3461666 nofraud non_flag
80 616    1 caso  205956 C1149407083    3134837 3340793 C1688019098     465696     97264 nofraud non_flag
81 626    1 caso  219220  C811562535    4028495 4247716  C451111351    1974115   3940085 nofraud non_flag
82 627    1 caso  254896 C1560379655    4247716 4502611 C1509514333     256668      2415 nofraud non_flag
83 634    1 caso  338965 C1591161296    5451215 5790180  C766572210     365376         0 nofraud non_flag
 [ reached 'max' / getOption("max.print") -- omitted 1121417 rows ]
```

En analysant les données à travers des graphiques, nous avons observé une corrélation très forte :

- entre oldbalanceDest et newbalanceDest (1)
- entre oldbalanceOrg et newbalanceOrig (1)
- une corrélation non négligeable entre amount et newbalanceDest (0.5)

Nous avons également pu observer que les opérations financières non frauduleuses représentaient plus de 99% des données et un peu plus de 0.10% des opérations frauduleuses.

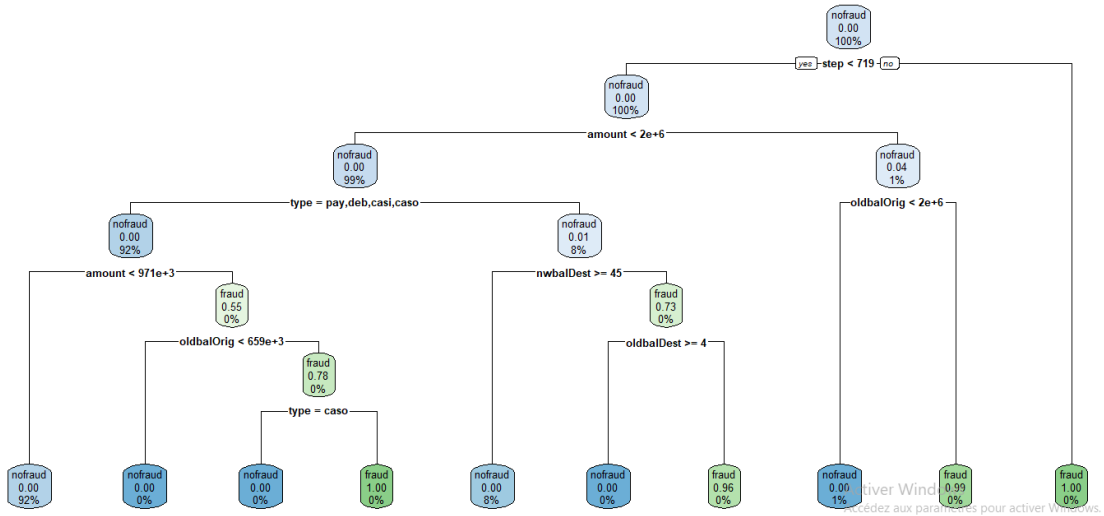
Par ailleurs, concernant les fraudes, elles sont axées principalement sur Cashout et Transfert.

4) Choix des modèles et résultats

4-1) Arbre de décision

Nous avons choisi de construire un arbre de décision afin d'avoir un visuel sur l'ensemble des données. De plus, plusieurs variables sont en chaîne de caractères.

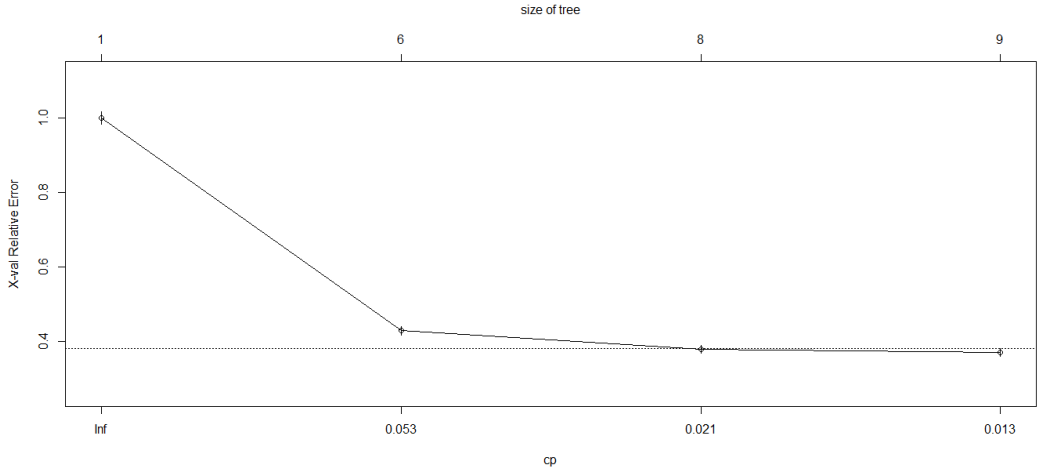
Le premier arbre que nous avons construit contient 9 feuilles.



En consultant les résultats, nous voyons que l'importance des différentes variables :

```
> summary(dtree)
Call:
rpart(formula = isFraud ~ ., data = df_train2, method = "class")
  n= 2557774

          CP nsplit rel error    xerror        xstd
1 0.09682635      0 1.0000000 1.0000000 0.017291912
2 0.02485030      6 0.3982036 0.3883234 0.010779866
3 0.01676647      8 0.3485030 0.3577844 0.010347514
4 0.01000000      9 0.3317365 0.3329341 0.009981861

Variable importance
 nwbalDest oldbalOrig       type oldbalDest       step     amount     nbOrig
        38         25         10          9          9          4          4

Node number 1: 2557774 observations,    complexity param=0.09682635
  predicted class=nofraud  expected loss=0.001305823  P(node) =1
    class counts: 2.55443e+06  3340
   probabilities: 0.999 0.001
  left son=2 (2557640 obs) right son=3 (134 obs)
  Primary splits:
      step       < 718.5     to the left,  improve=267.314500, (0 missing)
      amount     < 2025628   to the left,  improve= 48.986270, (0 missing)
      type       splits as  LRLLL, improve= 19.991900, (0 missing)
      oldbalOrig < 112958.5  to the left,  improve= 13.079390, (0 missing)
      nbOrig     < 0.5       to the right, improve=  6.044936, (0 missing)

Node number 2: 2557640 observations,    complexity param=0.09682635
  predicted class=nofraud  expected loss=0.001253499  P(node) =0.9999476
    class counts: 2.55443e+06  3206
   probabilities: 0.999 0.001
  left son=4 (2540130 obs) right son=5 (17510 obs)
  Primary splits:
      amount     < 2025628   to the left,  improve=45.50922, (0 missing)
      step       < 408.5     to the left,  improve=18.81299, (0 missing)
      type       splits as  LRLLL, improve=18.49391, (0 missing)
      oldbalOrig < 112958.5  to the left,  improve=12.11582, (0 missing)
      nbOrig     < 0.5       to the right, improve= 5.58148, (0 missing)
  Surrogate splits:
      nwbalDest < 236240900 to the left,  agree=0.993, adj=0, (0 split)
      isFlag    splits as  LR, agree=0.993, adj=0, (0 split)
```

Afin d'obtenir un arbre constitué d'un nombre de feuille optimal, nous avons décidé d'afficher un graphique qui permet d'évaluer les performances par validation croisée.



Nous remarquons par lecture graphique que les performances de notre arbre sont bonnes lorsque le nbre de feuilles dépasse 8.
C'est le cas de notre graphique. Sinon nous aurions cherché l'endroit qui minimise l'erreur afin de le faire correspondre avec le nombre de feuilles nécessaires à notre arbre pour éviter le surapprentissage.
Les résultats nous indiquent que toutes les variables sont significatives :

-plus le montant de la transaction est élevé, moins la transaction a de risque d'être frauduleuse

-plus le solde bancaire du compte d'origine est élevé, plus la transaction a de risque d'être frauduleuse

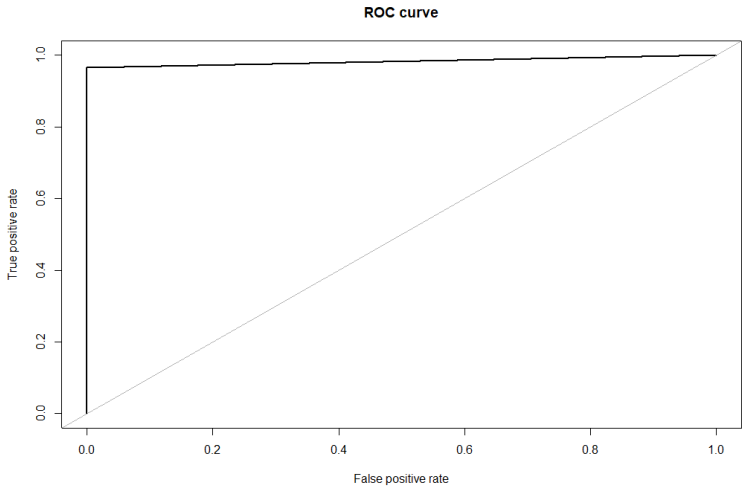
-plus le nouveau solde bancaire du compte d'origine est élevé, moins la transaction a de risque d'être frauduleuse

-plus l'ancien solde bancaire du compte destinataire est élevé, moins la transaction a de risque d'être frauduleuse

-plus le nouveau solde bancaire du compte destinataire est élevé, plus la transaction a de risque d'être frauduleuse

-lorsque la transaction est signalée, il y a plus de chance pour que l'opération soit frauduleuse

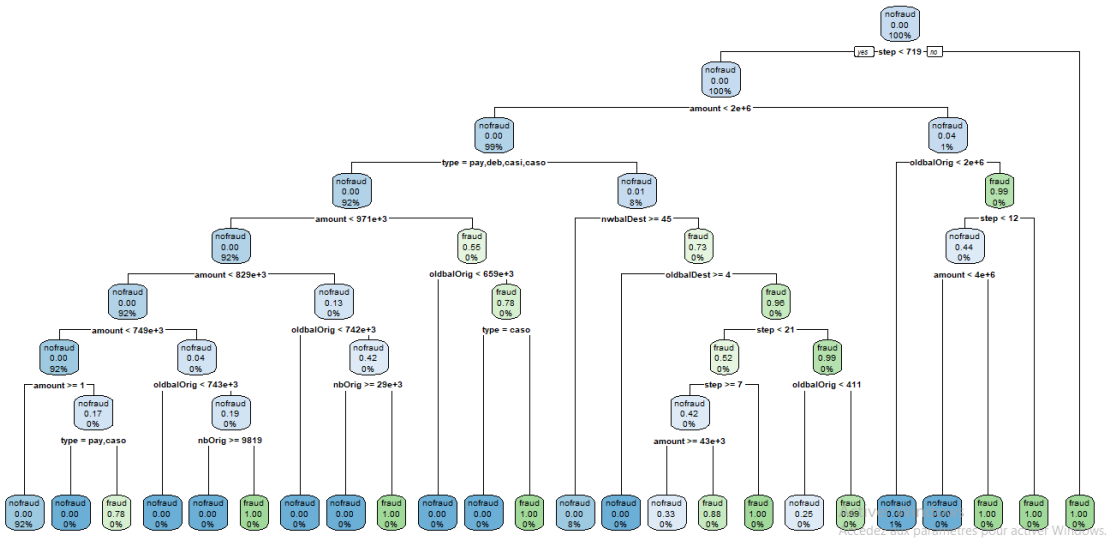
La courbe ROC du premier arbre :



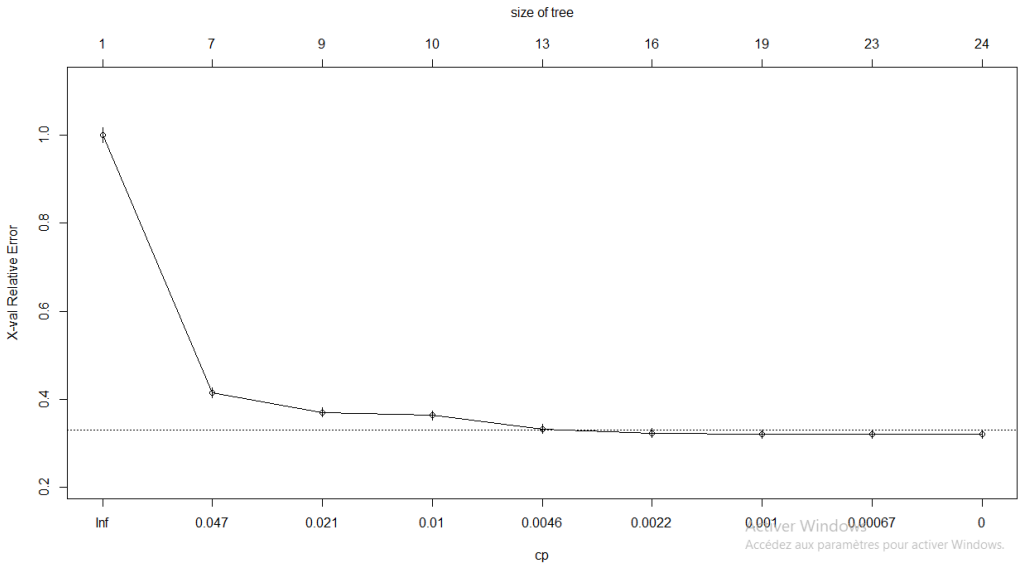
Le F1Score est de 0.79

```
> paste0("Precision: ", dtree_precision)
[1] "Precision: 0.965834428383706"
> paste0("Recall: ", dtree_recall)
[1] "Recall: 0.678044280442804"
> paste0("F1 Score: ", dtree_f1)
[1] "F1 Score: 0.796747967479675"
> paste0("AUC: ", dtree_auc)
[1] "AUC: NaN"
```

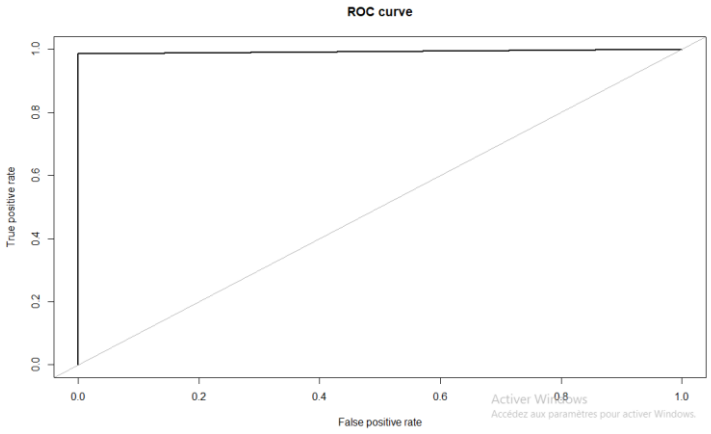
Dans le but de tenter d'améliorer le F1 score obtenu, nous avons décidé d'augmenter le nombre de feuilles de notre arbre de décision.



Le graphique qui permet de consulter les performances par validation croisée est le suivant :



La courbe ROC du deuxième arbre :



Ainsi, nous obtenu un F1score un peu plus élevé : 0.82.

```
> paste0("Precision: ", dtree_tuned_fit_precision)
[1] "Precision: 0.98578811369509"
> paste0("Recall: ", dtree_tuned_fit_recall)
[1] "Recall: 0.703874538745387"
> paste0("F1 Score: ", dtree_tuned_fit_f1)
[1] "F1 Score: 0.821313240043057"
> paste0("AUC: ", dtree_tuned_fit_auc)
[1] "AUC: NaN"
```

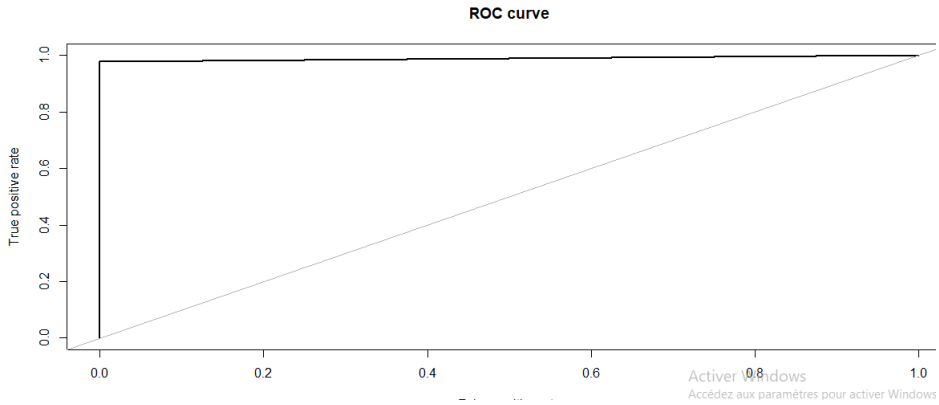
4-2) Régression logistique

Notre deuxième modèle est celui de régression logistique.

Notre choix s'est porté sur le fait de tester le modèle avec des données standardisées et des données non standardisées.
Ainsi, voici les résultats que nous obtenons avec les données non standardisées.

```
> summary(logreg)

Call:
glm(formula = isFraud ~ ., family = "binomial", data = df_train2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
 -8.49    0.00    0.00    0.00    8.49

Coefficients:
              Estimate Std. Error    z value Pr(>|z|)
(Intercept) -1.122e+15  1.020e+05 -1.099e+10   <2e-16 ***
step         7.778e+11  2.953e+02  2.634e+09   <2e-16 ***
typetra     -9.297e+13  1.761e+05 -5.280e+08   <2e-16 ***
typedeb     -8.970e+14  5.285e+05 -1.697e+09   <2e-16 ***
typecasi    -1.941e+15  1.034e+05 -1.878e+10   <2e-16 ***
typecaso     2.067e+14  1.484e+05  1.393e+09   <2e-16 ***
amount       3.156e+08  1.373e-01  2.298e+09   <2e-16 ***
oldbalorig   1.281e+09  3.383e-01  3.788e+09   <2e-16 ***
nbOrig      -1.336e+09  3.391e-01 -3.939e+09   <2e-16 ***
oldbalDest   3.357e+08  1.001e-01  3.353e+09   <2e-16 ***
nwbalDest   -3.370e+08  9.944e-02 -3.389e+09   <2e-16 ***
isFlagflag   2.516e+15  3.875e+07  6.494e+07   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance:  51037  on 2557773  degrees of freedom
Residual deviance: 178344  on 2557762  degrees of freedom
AIC: 178368

Number of Fisher Scoring iterations: 25
```

Voici la courbe ROC de cette première régression logistique :



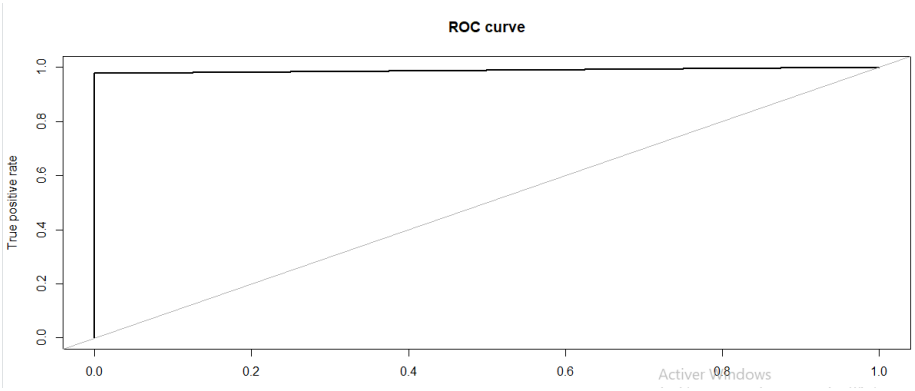
On obtient un F1score de 0.66 :

```
[1] "Precision: 0.913322632423756"
> paste0("Recall: ", logreg_recall)
[1] "Recall: 0.523459061637535"
> paste0("F1 Score: ", logreg_f1)
[1] "F1 Score: 0.665497076023392"
> paste0("AUC: ", logreg_auc)
[1] "AUC: NaN"
```

Voici les résultats que nous obtenons avec les données standardisées :

```
Call:
glm(formula = isFraud ~ ., family = "binomial", data = df_train2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
 -8.49    0.00     0.00     0.00     8.49

Coefficients:
              Estimate Std. Error    z value Pr(>|z|)
(Intercept) -9.922e+14  7.611e+04 -1.304e+10   <2e-16 ***
step         1.107e+14  4.202e+04  2.634e+09   <2e-16 ***
typetra     -9.297e+13  1.761e+05 -5.280e+08   <2e-16 ***
typedeb     -8.970e+14  5.285e+05 -1.697e+09   <2e-16 ***
typecasi    -1.941e+15  1.034e+05 -1.878e+10   <2e-16 ***
typecaso     2.067e+14  1.484e+05  1.393e+09   <2e-16 ***
amount       1.930e+14  8.398e+04  2.298e+09   <2e-16 ***
oldbalOrig   3.698e+15  9.764e+05  3.788e+09   <2e-16 ***
nbOrig      -3.903e+15  9.908e+05 -3.939e+09   <2e-16 ***
oldbalDest   1.147e+15  3.422e+05  3.353e+09   <2e-16 ***
nwbalDest   -1.245e+15  3.674e+05 -3.389e+09   <2e-16 ***
isFlagflag   2.516e+15  3.875e+07  6.494e+07   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance:  51037  on 2557773  degrees of freedom
Residual deviance: 178344  on 2557762  degrees of freedom
AIC: 178368
```

Voici la courbe ROC de cette première régression logistique :



Nous obtenons un F1score de 0.67

```
[1] "Precision: 0.841514726507714"
> paste0("Recall: ", logreg_fit_recall)
[1] "Recall: 0.551977920883165"
> paste0("F1 Score: ", logreg_fit_f1)
[1] "F1 Score: 0.666666666666667"
> paste0("AUC: ", logreg_fit_auc)
[1] "AUC: NaN"
```

Il n’y a pas de différences très significatives entre les deux modèles. Ce qui nous questionne car il est écrit dans la littérature que centrer et réduire les variables permet de réaliser une meilleure régression logistique.

4-3) Naive Bayes

Nous n'avons pas obtenu un bon F2score avec ce modèle. A vu des résultats, nous considérons qu'il n'est pas adapté.

4-4) Random Forest

Notre meilleur modèle a été le random forest.

```
> summary(modele_random)
                Length  Class  Mode
call                 4 -none- call
type                 1 -none- character
predicted      2557774 factor numeric
err.rate            60 -none- numeric
confusion            6 -none- numeric
votes          5115548 matrix numeric
oob.times      2557774 -none- numeric
classes              2 -none- character
importance           8 -none- numeric
importanceSD         0 -none- NULL
localImportance      0 -none- NULL
proximity            0 -none- NULL
ntree                1 -none- numeric
mtry                 1 -none- numeric
forest              14 -none- list
y              2557774 factor numeric
test                 0 -none- NULL
inbag                0 -none- NULL
terms                3 terms  call
```

```
> modele_random

Call:
 randomForest(formula = isFraud ~ ., data = df_train2, ntree = 20)
               Type of random forest: classification
                     Number of trees: 20
No. of variables tried at each split: 2

        OOB estimate of  error rate: 0.04%
Confusion matrix:
        nofraud fraud  class.error
nofraud 2554125    40 0.0000156607
fraud       937  2402 0.2806229410
>
```

```
> paste0("Precision: ", modele_random_precision)
[1] "Precision: 0.988693467336683"
> paste0("Recall: ", modele_random_recall)
[1] "Recall: 0.726014760147601"
> paste0("F1 Score: ", F1_Score(y_true, y_pred, positive = "fraud"))
[1] "F1 Score: 0.837234042553191"
> paste0("AUC: ", modele_random_auc)
[1] "AUC: NaN"
> roc.curve(y_pred, y_true, plotit = TRUE, add.roc = FALSE,
+           n.thresholds=100)
Area under the curve (AUC): 0.994
```

Voici la liste des F1 score de tous nos modèles utilisés :

```
> perfs[order(-f1_score),]
              precision    recall   f1_score auc
modele_random 0.9950311 0.7368905 0.84672304 NaN
dtree_tuned   0.9805447 0.6954922 0.81377826 NaN
dtree         0.9656489 0.6982521 0.81046450 NaN
logreg_fit    0.8415147 0.5519779 0.66666667 NaN
logreg        0.9133226 0.5234591 0.66549708 NaN
nbClassifier  0.0405602 0.2051518 0.06772969 NaN
```

Après avoir sélectionné notre meilleur modèle (random forest), nous l'avons utilisé sur l'échantillon test (df_test2) et obtenons les résultats suivants :

```
> paste0("Precision: ", modele_random_precision)
[1] "Precision: 0.977556109725686"
> paste0("Recall: ", modele_random_recall)
[1] "Recall: 0.687719298245614"
> paste0("F1 Score: ", modele_random_f1)
[1] "F1 Score: 0.807415036045314"
> paste0("AUC: ", modele_random_auc)
[1] "AUC: NaN"
```

5) Modèle multilinéaire

```
states <- as.data.frame(projetstats[,c("step", "amount", "oldbalanceOrg", "newbalanceOrig",
                                       "oldbalanceDest","newbalanceDest","isFraud","isFlaggedFraud")])
cor(states)

states_train <- as.data.frame(df_train_numric[,c("step", "amount", "oldbalanceOrg",
"newbalanceOrig",
                                                 "oldbalanceDest","newbalanceDest","isFraud","isFlaggedFraud")])
cor(states_train)

states_validation<- as.data.frame(df_validation_numric[,c("step", "amount", "oldbalanceOrg",
"newbalanceOrig",
                                                 "oldbalanceDest","newbalanceDest","isFraud","isFlaggedFraud")])
cor(states_validation)

states_test <- as.data.frame(df_test_numric[,c("step", "amount", "oldbalanceOrg", "newbalanceOrig",
                                                 "oldbalanceDest","newbalanceDest","isFraud","isFlaggedFraud")])
cor(states_test)



isFraud1<-projetstats$isFraud
str(isFraud1)
fit_df_projetdatas <- lm(isFraud1 ~
step+type+amount+obOrg+nbOrig+obDest+nbDest,data=df_projetdatas)
summary(fit_df_projetdatas)
```

```
df_train_numric <- collect(sample_frac(projetstats, 0.6), replace = FALSE)
df_train_numric
fit_df_train <- lm(isFraud ~
step+type+amount+oldbalanceOrg+newbalanceOrig+oldbalanceDest+newbalanceDest,data=df_train_
numric )
summary(fit_df_test)

df_test_numric <- collect(sample_frac(projetstats, 0.2), replace = FALSE)
df_test_numric
fit_df_test <- lm(isFraud ~
step+type+amount+oldbalanceOrg+newbalanceOrig+oldbalanceDest+newbalanceDest,data=df_test_n
umric )
summary(fit_df_test)


df_validation_numric <- collect(sample_frac(projetstats, 0.2), replace = FALSE)
df_validation_numric
fit_df_validation <- lm(isFraud ~
step+type+amount+oldbalanceOrg+newbalanceOrig+oldbalanceDest+newbalanceDest,data=df_valida
tion_numric)
summary(fit_df_validation )
#-----------------------------------------les confint
confint(fit_df_projetdatas)
confint(fit_df_train)
confint(fit_df_test)
confint(fit_df_validation)

#-----------------------------------------les plot
plot(fit_df_projetdatas)
```

Nous n'avons pas pu terminer ce modèle et l'utiliser pour les prédictions.

6) Conclusion

Les problématiques auxquelles nous aurions tenté de répondre sont :

Le déséquilibre des données. Nous aurions voulu utiliser le modèle SMOTE afin de réduire la classe majoritaire. Cela aurait été un choix en connaissance du fait que cela impliquerait d'accepter de perdre certaines informations.
Au vu de la quantité de données que constitue train.csv, augmenter la classe minoritaire ne nous parait pas intéressant d'un point de vue de taille de fichier.

Nous avons tâché de ne pas modifier le fichier afin de prendre en compte un maximum d'informations. Cependant, des données comme les noms des comptes d'origine ou de destination nous ont empêché de réaliser des modèles les prenant en compte.
Ainsi, nous nous n'avons pas pu explorer la piste exprimant le fait que certains comptes auraient pu être employé plusieurs fois dans le cadre de fraude bancaire.

Finalement, nous pouvons conclure que les variables-clés pour prédire si une opération est frauduleuse ou non sont :

- le type de paiement
- le nouveau solde du destinataire
- l'ancien solde du compet d'origine

Bibliographie

https://www.youtube.com/watch?v=0Jp4gsfOLMs&list=PLblh5JKOoLUJJpBNfk8_YadPwDTO2SCbx

https://datascientest.com/acp

http://www.sthda.com/french/articles/38-methodes-des-composantes-principales-dans-r-guide-pratique/73-acp-analyse-en-composantes-principales-avec-r-l-essentiel/#:~:text=L'analyse%20en%20composantes%20principales%20est%20utilis%C3%A9e%20pour%20extraire%20et,nouvelles%20variables%20appel%C3%A9es%20composantes%20principales.

https://eric.univ-lyon2.fr/~ricco/cours/slides/classif_centres_mobiles.pdf

http://wikistat.fr/pdf/st-l-des-bi
https://openclassrooms.com/fr/courses/4379436-explorez-vos-donnees-avec-des-algorithmes-non-supervises/4379506-tp-acp-d-un-jeu-de-donnees-sur-les-performances-d-athletes-olympiques

http://eric.univ-lyon2.fr/~ricco/cours/slides/logistic_regression_ml.pdf

http://eric.univ-lyon2.fr/~ricco/cours/slides/naive_bayes_classifier.pdf

https://eric.univ-lyon2.fr/~ricco/cours/slides/intro_ds_from_dm_to_bd.pdf

https://eric.univ-lyon2.fr/~ricco/cours/slides/Apprentissage_Supervise.pdf

https://www.youtube.com/watch?v=Ssen9A9weko

https://openclassrooms.com/fr/courses/5919236-decouvrez-la-science-des-donnees-pour-les-objets-connectes/6068921-comprenez-lanalyse-en-composantes-principales#:~:text=L'objectif%20de%20l'analyse,plus%20pertinent%20des%20donn%C3%A9es%20initiales.

https://r-graph-gallery.com/199-correlation-matrix-with-ggally.html

http://wikistat.fr/pdf/st-l-des-bi

https://eric.univ-lyon2.fr/~ricco/cours/slides/classif_centres_mobiles.pdf

http://www.sthda.com/french/articles/38-methodes-des-composantes-principales-dans-r-guide-pratique/73-acp-analyse-en-composantes-principales-avec-r-l-essentiel/#:~:text=L'analyse%20en%20composantes%20principales%20est%20utilis%C3%A9e%20pour%20extraire%20et,nouvelles%20variables%20appel%C3%A9es%20composantes%20principales.
https://openclassrooms.com/fr/courses/5919236-decouvrez-la-science-des-donnees-pour-les-objets-connectes/6068921-comprenez-lanalyse-en-composantes-principales#:~:text=L'objectif%20de%20l'analyse,plus%20pertinent%20des%20donn%C3%A9es%20initiales.

https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501907-reduction-de-

dimensionnalite-en-machine-learning-definition/#:~:text=La%20r%C3%A9duction%20de%20dimensionnalit%C3%A9%20en,et%20de%20temps%20d'analyse.

http://rstudio-pubs-static.s3.amazonaws.com/74431_8cbd662559f6451f9cd411545f28107f.html

https://www.youtube.com/watch?v=aFvBhgmawcs

https://larevueia.fr/7-methodes-pour-eviter-loverfitting/

https://www.google.com/search?q=regression+logistique+r+descente+de+gradient&rlz=1C1UEAD_frFR933FR933&oq=regression+logistique+r+descente+de+&aqs=chrome.6.69i57j33i160l5j33i22i29i30l2.10329j0j7&sourceid=chrome&ie=UTF-8

https://www.youtube.com/watch?v=rawaCES1Qf8

http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/fr_Tanagra_R_compiler_package.pdf

https://www.youtube.com/watch?v=Wh427utosW4

https://www.youtube.com/watch?v=IHjro2qZtog

https://www.datanovia.com/en/fr/blog/comment-normaliser-et-standardiser-les-donnees-dans-r-pour-une-visualisation-en-heatmap-magnifique/