

Analyse de la Clientèle d'un Concessionnaire Automobile pour la Recommandation de Modèle

Rendu Final

Groupe 5



Quan ZHANG
Yajuan LUO
Yue ZHAO
Chaymae FAZAZI-IDRISSI

Table des matières

Contents

1	Description	3
2	Préparation	4
2.1	Nettoyage de la base de données	4
2.1.1	Nettoyage dans fichiers .csv	4
2.1.2	Nettoyage dans Oracle sous SQL	4
2.2	Charger les données dans R	8
2.3	Fusion des fichiers	8
2.3.1	Fusion entre Catalogue.csv et Immatriculations.csv	9
2.3.2	Fusion entre Clients.csv et la fusion comme 3.2.1	10
3	Petit essai	12
3.1	Renommer tableFinal	12
3.2	Apprentissage supervisé grâce aux différents classifieurs	12
3.2.1	Arbre de Décision	12
4	Construire le modèle	14
4.1	Le modèle de la prédiction du PRIX	14
4.1.1	Fusion des données	14
4.1.2	Visuellement	15
4.1.3	Apprentissage supervisé grâce aux différents classifieurs	16
4.1.4	Application de la méthode	19
4.2	Le modèle de la prédiction du LONGUEUR(ZHAO YUE)	21
4.3	Le modèle de la prédiction du NBPORTES(CHAYMAE)	21
4.4	Le modèle de la prédiction du COULEUR(LUO YAJUAN)	21
4.5	Le modèle de la prédiction du OCCASION	22
4.5.1	Fusion des données	22
4.5.2	Visuellement	22
4.5.3	Apprentissage supervisé grâce aux différents classifieurs	23
4.5.4	Application de la méthode	27
5	Conclusion	28
5.1	Erreurs	28
5.2	NBPLACES	28
5.3	La manque de la mélange de la base de données	29

1 Description

Pour ce projet on doit concevoir un outil qui va permettre à un concessionnaire automobile de cibler les véhicules qui peuvent intéresser ses clients plus précisément :

- un outil rapide qui peut évaluer le type de véhicule qui peut intéresser ses clients en se basant sur les caractéristiques de chaque client et les différents besoins
- envoyer une documentation précise sur le véhicule le plus adéquat pour des clients sélectionnés par son service marketing

Pour cela on doit faire une méthode de gestion de projet et un plan de mise en œuvre, pour l'analyse aussi il va nous falloir les techniques de data mining, machine learning et deep learning qu'on va utiliser dans notre projet pour répondre à cette problématique en se basant sur des fichiers de données qu'on a à disposition (Catalogue, Immatriculations, Marketing) et les données des clients.

2 Préparation

2.1 Nettoyage de la base de données

2.1.1 Nettoyage dans fichiers .csv

Nous allons trouver et corriger les erreurs de syntaxe:

- Nous remplaçons (è, é) par (e, e) dans tous les fichiers .csv;
- Dans fichier clients.csv: Nous remplaçons (Masculin, Homme, Féminin) par (M, M, F, F);

Pour l'instant, nous ne traitons pas ces valeurs vides et inconnus (grâce à `filter()` dans `library(dplyr)` sous R ou bien nettoyage sous SQL, nous allons sélectionner les données correctes), par exemple:

535	48 F	159	En Couple		FALSE	9652 KH 90
536	36 M	1270	celibataire	0	FALSE	6249 OO 45
537	52 M	594	celibataire	0	FALSE	7653 EU 72
538	41 F	1385	En Couple ?		FALSE	5239 SI 80

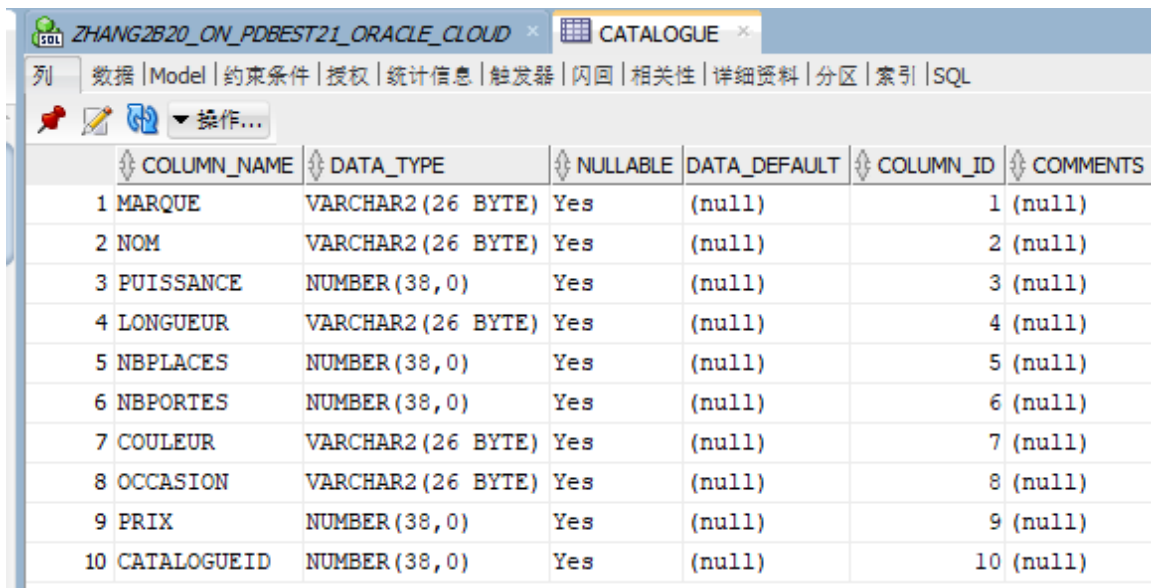
Figure 1: exemple des valeurs vides et inconnus

2.1.2 Nettoyage dans Oracle sous SQL

Nous changeons "2eme voiture" par "deuxiemeVoiture". Quand nous chargeons les données, nous allons cocher pour que les données puissent nullable(default null).

Ensuite, nous allons nettoyer chaque table:

CATALOGUE



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	MARQUE	VARCHAR2(26 BYTE)	Yes	(null)	1 (null)	
2	NOM	VARCHAR2(26 BYTE)	Yes	(null)	2 (null)	
3	PUISSANCE	NUMBER(38,0)	Yes	(null)	3 (null)	
4	LONGUEUR	VARCHAR2(26 BYTE)	Yes	(null)	4 (null)	
5	NBPLACES	NUMBER(38,0)	Yes	(null)	5 (null)	
6	NBPORTES	NUMBER(38,0)	Yes	(null)	6 (null)	
7	COULEUR	VARCHAR2(26 BYTE)	Yes	(null)	7 (null)	
8	OCCASION	VARCHAR2(26 BYTE)	Yes	(null)	8 (null)	
9	PRIX	NUMBER(38,0)	Yes	(null)	9 (null)	
10	CATALOGUEID	NUMBER(38,0)	Yes	(null)	10 (null)	

Figure 2: Table: CATALOGUE

Il s'agit d'un tableau très basique, nous pouvons donc constater qu'il n'y a pas d'erreurs dans ce tableau. Donc nous changeons rien dedans.

CLIENTS

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	AGE	NUMBER(38,0)	Yes	(null)	1 (null)	
2	SEXE	VARCHAR2(26 BYTE)	Yes	(null)	2 (null)	
3	TAUX	VARCHAR2(26 BYTE)	Yes	(null)	3 (null)	
4	SITUATIONFAMILIALE	VARCHAR2(26 BYTE)	Yes	(null)	4 (null)	
5	NBENFANTSACHARGE	NUMBER(38,0)	Yes	(null)	5 (null)	
6	DEUXIEMEVOITURE	VARCHAR2(26 BYTE)	Yes	(null)	6 (null)	
7	IMMATRICULATION	VARCHAR2(26 BYTE)	Yes	(null)	7 (null)	

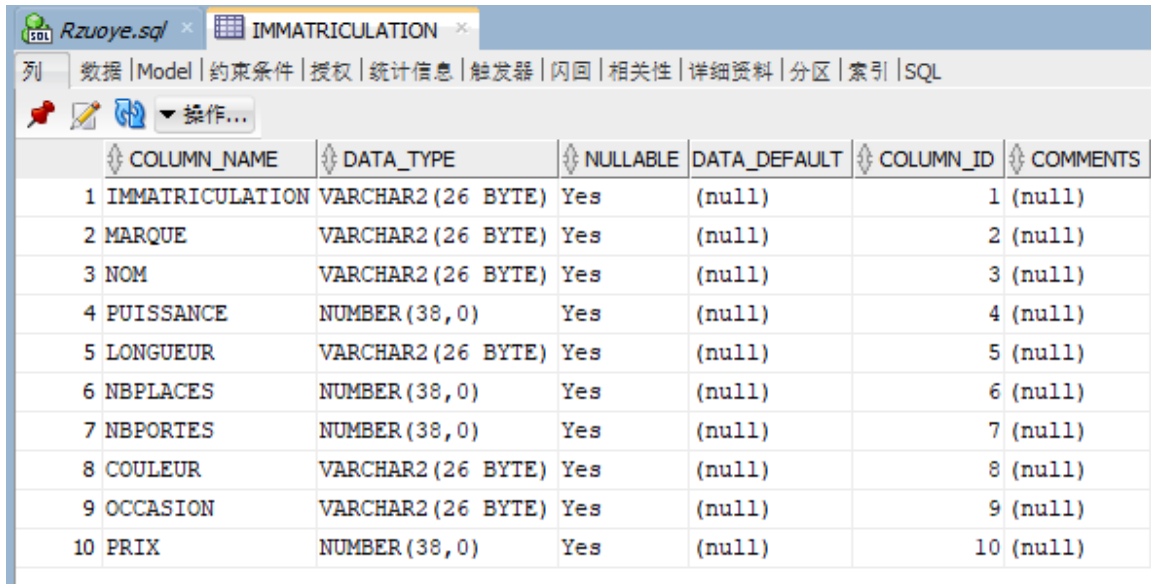
Figure 3: Table: CLIENTS

Les opérations dans table clients sous SQL:

```

1  --Pourque Age soit dans [18, 84]
2  select * from clients where age < 18 or age > 84;
3  DELETE from clients where age < 18 or age > 84;
4
5  --Domain de valeurs 'SEXE': 'F','M'
6  select * from clients where SEXE != 'M' and SEXE != 'F';
7  delete from clients where SEXE != 'M' and SEXE != 'F';
8
9  --Domain de valeurs 'taux': [544, 74185]
10 select * from clients where taux = ' ';
11 delete from clients where taux = ' ';
12 select * from clients where taux = '?';
13 delete from clients where taux = '?';
14 select * from clients where taux < 544 or taux > 74185;
15 delete from clients where TO_NUMBER(taux) < 544 or TO_NUMBER(taux) > 74185;
16
17 --Pour SITUATIONFAMILIALE, on fait group by pour voir les erreurs possibles
18 select sum(age),SITUATIONFAMILIALE from clients group by SITUATIONFAMILIALE;
19 delete from clients where SITUATIONFAMILIALE = '?';
20 delete from clients where SITUATIONFAMILIALE = ' ';
21 delete from clients where SITUATIONFAMILIALE = 'N/D';
22
23 --Domain de valeurs NBENFANTSACHARGE: [0, 4]
24 select * from clients where NBENFANTSACHARGE < 0 or NBENFANTSACHARGE > 4;
25 delete from clients where NBENFANTSACHARGE < 0 or NBENFANTSACHARGE > 4;
26
27 --DEUXIEMEVOITURE: TRUE or FALSE
28 select * from clients where DEUXIEMEVOITURE != 'TRUE' and DEUXIEMEVOITURE != 'FALSE';
29 delete from clients where DEUXIEMEVOITURE != 'TRUE' and DEUXIEMEVOITURE != 'FALSE';
30
31 -- on touche rien pour les valeurs IMMATRICULATION car il y a pas d'errers comme ' ', '?', 'N/D'

```

IMMATRICULATION


	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	IMMATRICULATION	VARCHAR2(26 BYTE)	Yes	(null)	1	(null)
2	MARQUE	VARCHAR2(26 BYTE)	Yes	(null)	2	(null)
3	NOM	VARCHAR2(26 BYTE)	Yes	(null)	3	(null)
4	PUISSANCE	NUMBER(38,0)	Yes	(null)	4	(null)
5	LONGUEUR	VARCHAR2(26 BYTE)	Yes	(null)	5	(null)
6	NBPLACES	NUMBER(38,0)	Yes	(null)	6	(null)
7	NBPORTES	NUMBER(38,0)	Yes	(null)	7	(null)
8	COULEUR	VARCHAR2(26 BYTE)	Yes	(null)	8	(null)
9	OCCASION	VARCHAR2(26 BYTE)	Yes	(null)	9	(null)
10	PRIX	NUMBER(38,0)	Yes	(null)	10	(null)

Figure 4: Table: IMMATRICULATION

Les opérations dans table IMMATRICULATION sous SQL:

```

1  --Afin d'utiliser Group by, on doit nettoyer au moins un column, par exemple 'PRIX'
2  --Domain de valeurs PRIX: [7500, 101300]
3  select * from IMMATRICULATION where prix < 7500 or prix > 101300;
4  --on touche pas si'l y a pas d'erreurs
5
6  --Pour les valeurs marque:
7  select sum(prix), MARQUE from IMMATRICULATION group by MARQUE;
8  --on touche pas si'l y a pas d'erreurs
9
10 --Pour les valeurs nom:
11 select sum(prix), nom from IMMATRICULATION group by nom;
12 --on touche pas si'l y a pas d'erreurs
13
14 --Domain de valeurs PUISSANCE: [55, 507]
15 select * from IMMATRICULATION where PUISSANCE < 55 or PUISSANCE > 507;
16 --on touche pas si'l y a pas d'erreurs
17
18 --Pour les valeurs longueur:
19 select sum(prix), longueur from IMMATRICULATION group by longueur;
20 --on touche pas si'l y a pas d'erreurs
21
22 --Domain de valeurs NBPLACES: [5, 7]
23 select * from IMMATRICULATION where NBPLACES < 5 or NBPLACES > 7;
24 --on touche pas si'l y a pas d'erreurs
25
26 --Domain de valeurs NBPORTES: [3, 5]
27 select * from IMMATRICULATION where NBPORTES < 3 or NBPORTES > 5;
28 --on touche pas si'l y a pas d'erreurs
29

```

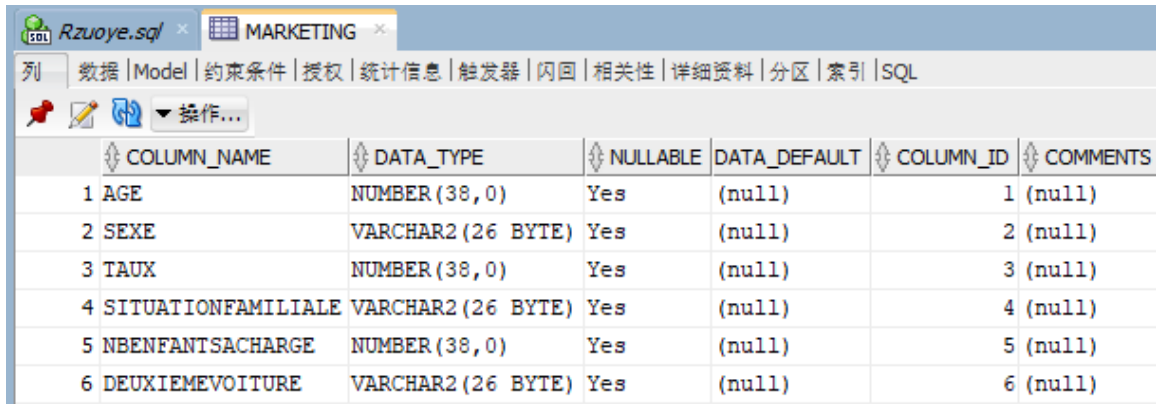
```

30 --Pour les valeurs COULEUR:
31 select sum(prix), COULEUR from IMMATRICULATION group by COULEUR;
32 --on touche pas si'l y a pas d'erreurs
33
34 --Pour les valeurs OCCASION:
35 select sum(prix), OCCASION from IMMATRICULATION group by OCCASION;
36 --on touche pas si'l y a pas d'erreurs

```

Après vérification, il n'y a aucune erreur dans ce tableau

MARKETING



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	AGE	NUMBER(38,0)	Yes	(null)	1	(null)
2	SEXE	VARCHAR2(26 BYTE)	Yes	(null)	2	(null)
3	TAUX	NUMBER(38,0)	Yes	(null)	3	(null)
4	SITUATIONFAMILIALE	VARCHAR2(26 BYTE)	Yes	(null)	4	(null)
5	NBNFANTSACHARGE	NUMBER(38,0)	Yes	(null)	5	(null)
6	DEUXIEMEVOITURE	VARCHAR2(26 BYTE)	Yes	(null)	6	(null)

Figure 5: Table: MARKETING

Les opérations dans table MARKETING sous SQL:

```

1 --Domain de valeurs 'taux': [544, 74185]
2 select * from MARKETING where taux < 544 or taux > 74185;
3 delete from MARKETING where taux < 544 or taux > 74185;

```

IL FAUT PAS OUBLIER:

```

1 COMMIT;

```

2.2 Charger les données dans R

```

1 install.packages("RJDBC")
2 library(RJDBC)
3
4 drv <- RJDBC::JDBC(driverClass = "oracle.jdbc.OracleDriver", classPath = Sys.glob("C:/Users/12506/
   OneDrive/Desktop/ESTIA_3A/R/Oracle/drivers/*"))
5
6 ##classPath : add path to drivers jdbc
7
8 #Connexion OK
9 conn <- dbConnect(drv, "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=144.21.67.201) (
   PORT=1521)) (CONNECT_DATA=(SERVICE_NAME=pdbest21.631174089.oraclecloud.internal)))", "ZHANG2B20",
   "ZHANG2B2001")
10
11 allTables <- dbGetQuery(conn, "SELECT owner, table_name FROM all_tables where owner = 'BABEAU2B20'")
12
13 tableCatalogue <- dbGetQuery(conn, "select * from Catalogue")
14 tableClients <- dbGetQuery(conn, "select * from Clients")
15 tableIm <- dbGetQuery(conn, "select * from IMMATRICULATION")
16 tableMar <- dbGetQuery(conn, "select * from MARKETING")
17 View(tableCatalogue)
18 View(tableClients)
19 View(tableIm)
20 View(tableMar)

```

Data		
▶ allTables	5 obs. of 2 variables	📄
▶ conn	Formal class JDBCConnection	🔍
▶ drv	Formal class JDBCDriver	🔍
▶ tableCatalogue	270 obs. of 10 variables	📄
▶ tableClients	11065 obs. of 7 variables	📄
▶ tableIm	1048575 obs. of 10 variables	📄
▶ tableMar	9 obs. of 6 variables	📄

Figure 6: Les tables apres les nettoyages

2.3 Fusion des fichiers

- Le fichier Clients.csv contient les informations sur les clients ayant les véhicules vendus cette année.
- Le fichier Immatriculations.csv contient les informations sur les véhicules vendus cette année.
- le fichier Catalogue.csv identifier des catégories de véhicules.

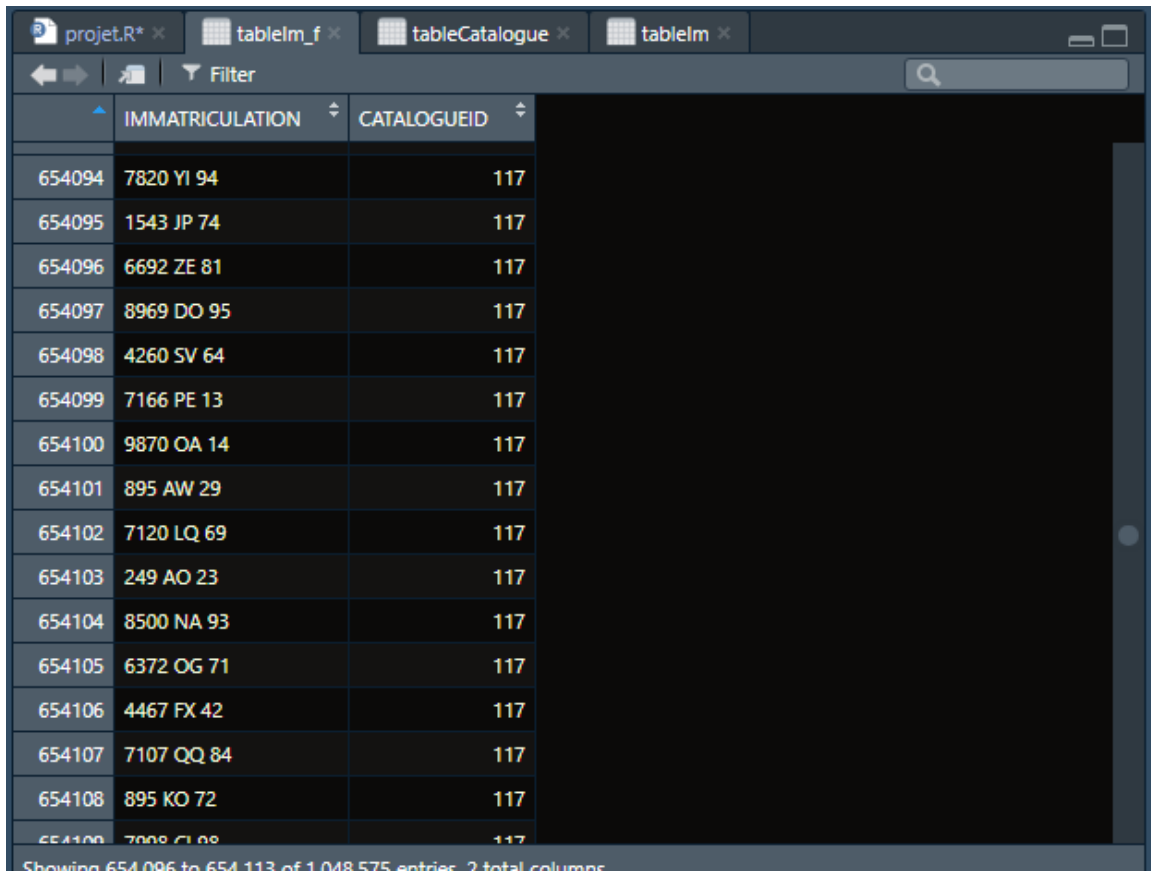
Afin d'éviter une prédiction introuvable (Différents types de voitures ont des paramètres différents) et donner la recommandation aux clients, nous prédisons un paramètre chaque fois (marque, nom, puissance....).

Pour la recommandation de modèle, au début de ce projet, nous avons ajouté un column "catalogueId" dans fichier Catalogue.csv. Grâce à ce column, la fusion entre Catalogue.csv et Immatriculations.csv nous permet d'obtenir une relation entre **IMMATRICULATION** et **CATALOGUEID** et la fusion entre Immatriculations.csv et Clients.csv nous permet d'obtenir une relation entre **les paramètres de clients** et **CATALOGUEID**.

Nous allons analyser les relations entre les informations sur les clients et CATALOGUEID et predire pour la Recommandation de Modèle pour les clients sélectionnés par le service marketing dans Marketing.csv.

2.3.1 Fusion entre Catalogue.csv et Immatriculations.csv

```
1 ##charger tous les libraries possible utilises
2 library(rvest)
3 library(ggplot2)
4 library(dplyr)
5 library(scales)
6 library(maps)
7 library(mapproj)
8 library(plotly)
9 library(rpart)
10 library(rpart.plot)
11 library(C50)
12 library(tree)
13 library(ROCR)
14 library(randomForest)
15 library(e1071)
16 library(naivebayes)
17 library(nnet)
18 library(kknn)
19
20 tableIm.f <- merge(tableIm, tableCatalogue, by = c("MARQUE", "NOM", "PUISSANCE", "LONGUEUR", "NBPORTES", "COULEUR", "OCCASION", "PRIX"))
21
22 tableIm.f <- tableIm.f[c(9, 12)]
23 View(tableIm.f)
```



	IMMATRICULATION	CATALOGUEID
654094	7820 YI 94	117
654095	1543 JP 74	117
654096	6692 ZE 81	117
654097	8969 DO 95	117
654098	4260 SV 64	117
654099	7166 PE 13	117
654100	9870 OA 14	117
654101	895 AW 29	117
654102	7120 LQ 69	117
654103	249 AO 23	117
654104	8500 NA 93	117
654105	6372 OG 71	117
654106	4467 FX 42	117
654107	7107 QQ 84	117
654108	895 KO 72	117
654109	7008 CJ 08	117

Figure 7: Fusion entre Catalogue.csv et Immatriculations.csv

2.3.2 Fusion entre Clients.csv et la fusion comme 3.2.1

```

1 tableFinal <- merge(tableClients, tableIm_f, by = "IMMATRICULATION")
2 View(tableFinal)
3
4 ## supprimer column "IMMATRICULATION"
5 tableFinal <- subset(tableFinal, select=-IMMATRICULATION)
6 View(tableFinal)

```

Finalement, nous avons obtenu une table qui contient tous les champs nous interesent:

	AGE	SEXE	TAUX	SITUATIONFAMILIALE	NBENFANTSACHARGE	DEUXIEMEVOITURE	CATALOGUEID
10452	24	M	1273	celibataire	0	FALSE	266
10453	74	M	552	En Couple	4	FALSE	151
10454	22	M	587	En Couple	2	FALSE	219
10455	69	F	1344	En Couple	3	FALSE	152
10456	58	F	884	En Couple	1	TRUE	141
10457	30	M	567	celibataire	0	FALSE	241
10458	69	M	1201	En Couple	2	FALSE	70
10459	25	M	1014	En Couple	2	FALSE	184
10460	49	M	972	En Couple	2	FALSE	181
10461	39	M	580	Seule	1	FALSE	215
10462	18	F	1335	En Couple	2	FALSE	184
10463	41	M	575	En Couple	1	FALSE	220
10464	38	M	722	En Couple	2	FALSE	232
10465	62	F	1233	En Couple	3	TRUE	152
10466	77	F	584	En Couple	2	FALSE	146
10467	31	M	588	En Couple	1	TRUE	264

Figure 8: La table finale

Ensuite, nous allons analyser les relations parmi les informations de clients et modèle de voiture.

3 Petit essai

Comme nous l'avons fait au début, nous avons ajouté une colonne ID à la catégorie. Notre idée est de créer directement un modèle, qui nous permet de prédire directement un ID, à travers différents algorithmes en apprentissage supervisé.

3.1 Renommer tableFinal

```
> table(tableEssaiFinal$CATALOGUEID)
```

1	2	3	6	7	22	23	24	26	28	29	36	37	38	39	40	46
72	74	54	72	62	36	52	1	50	46	54	1	3	2	1	2	21
47	48	49	50	61	62	63	64	65	66	67	68	69	70	76	77	78
21	23	30	29	31	40	30	31	39	84	84	84	94	74	60	79	67
79	80	86	87	88	89	90	92	96	97	98	99	100	112	113	115	117
78	58	21	28	39	26	27	1	41	30	36	46	35	41	44	46	41
120	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151
47	3	1	3	9	1	25	29	22	14	17	52	45	49	41	107	32
152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	168	169
85	104	88	121	39	40	14	17	26	63	9	16	35	41	5	9	7
170	173	181	182	183	184	185	186	187	188	189	190	206	207	208	209	210
3	4	337	34	40	357	36	37	46	313	328	327	35	29	39	42	34
215	216	218	219	220	221	224	231	232	233	234	235	236	237	238	239	240
37	33	37	44	44	1	1	59	424	429	404	61	53	407	68	407	71
241	246	247	248	249	251	252	253	254	255	258	259	260	261	262	263	264
29	26	23	36	34	3	8	8	3	13	9	8	3	63	547	561	64
265	266	267	268	269	270											
65	500	507	42	53	499											

Comme dans cette image, bien que nous ayons cent mille données, face à 270 modèles, il semble que ce ne soit pas suffisant pour obtenir un modèle mature. Nous avons quand même décidé de continuer.

3.2 Apprentissage supervisé grâce aux différents classifieurs

Creation des ensembles d'apprentissage et de test:

```
1 tableFinal <- merge(tableClients, tableIm_f, by = "IMMATRICULATION")
2 View(tableFinal)
3
4 # Creation des ensembles d'apprentissage et de test
5 id_EA <- tableEssaiFinal[1:7388,]
6 id_ET <- tableEssaiFinal[7389:11082,]
```

3.2.1 Arbre de Décision

```
1 tableFinal <- merge(tableClients, tableIm_f, by = "IMMATRICULATION")
2 View(tableFinal)
3 # Apprentissage du classifieur de type arbre de decision rpart
4 treeEssai1 <- rpart(CATALOGUEID~., id_EA)
5 prp(treeEssai1, type=4, extra=1, box.col=c("tomato", "skyblue")[treeEssai1$frame$yval])
6 ##Warning message:
```

```

7  ##labs do not fit even at cex 0.15, there may be some overplotting
8
9  # Test du classifieur : classe predite
10 pred.treeEssai1 <- predict(treeEssai1, id_ET, type="class")
11 ##Invalid prediction for "rpart" object
12
13 # Apprentissage du classifieur de type arbre de decision C5.0
14 id_EA$CATALOGUEID <- as.factor(id_EA$CATALOGUEID)
15 treeEssai2 <- C5.0(CATALOGUEID~., id_EA)
16 print(treeEssai2)
17 ##Non-standard options: attempt to group attributes
18 plot(treeEssai2, type="simple")
19 pred.treeEssai2 <- predict(treeEssai2, id_ET, type="class")
20 table(pred.treeEssai2)
21 table(id_ET$CATALOGUEID, pred.treeEssai2)
22 ##La matrice de confusion
23 tabletreeEssai2 <- as.data.frame(table(id_ET$CATALOGUEID, pred.treeEssai2))
24 ##taux de reussir
25 colnames(tabletreeEssai2) = list("Classe", "Prediction", "Effectif")
26 sum(tabletreeEssai2[tabletreeEssai2$Classe==tabletreeEssai2$Prediction,"Effectif"])/nrow(id_ET)
27 ##Error in Ops.factor(tabletreeEssai2$Classe, tabletreeEssai2$Prediction) :
28 ##level sets of factors are different

```

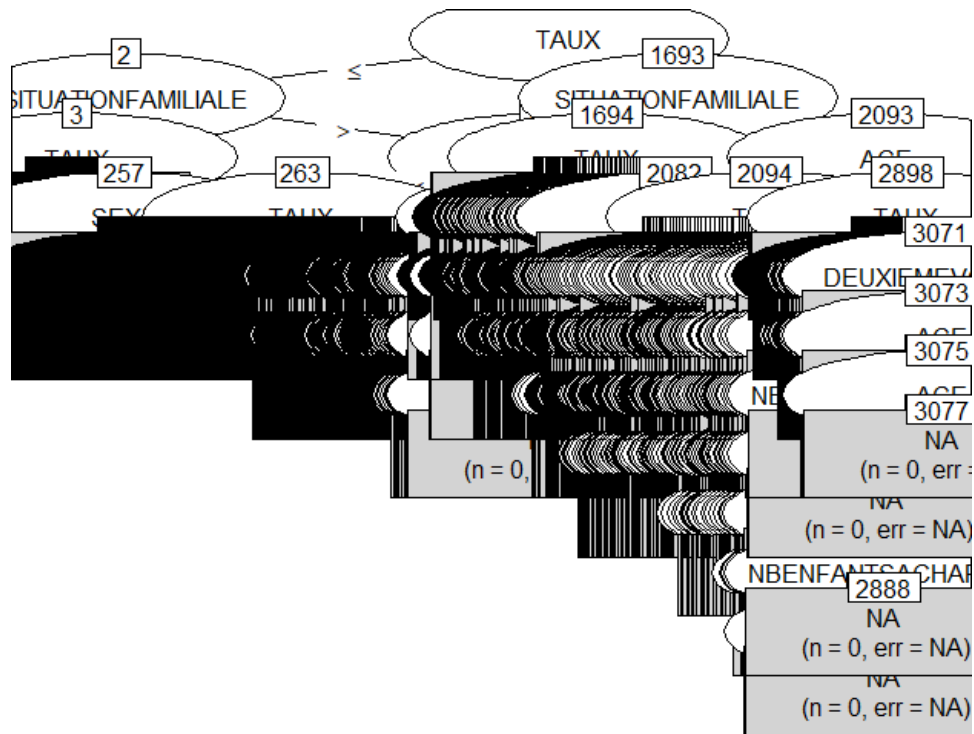


Figure 9: arbre de decision C5.0

Parce qu'il y a trop de branches, les images ne sont pas visibles. Par ailleurs, le taux de reussir n'est pas calculable. Donc, nous arrêtons d'explorer cette situation en profondeur.

4 Construire le modèle

Après avoir analysé les données du tableau du catalogue, nous avons décidé d'établir un modèle prédictif pour les paramètres de voiture suivants:

- PRIX
- LONGUEUR
- NBPORTES
- COULEUR
- OCCASION

4.1 Le modèle de la prédiction du PRIX

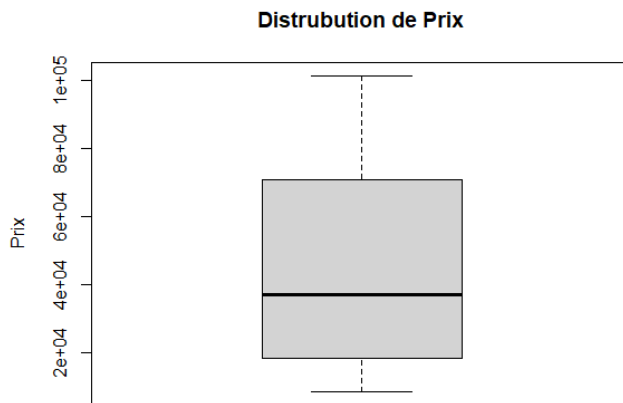
4.1.1 Fusion des données

```

1 tableImPrix <- tableIm_f[c(8, 9)]
2 tablePrixFinal <- merge(tableClients, tableImPrix, by = "IMMATRICULATION", incomparables = NA)
3 #requete pour la distribution des donnes: Prix
4 summary(tablePrixFinal$PRIX)
5 boxplot(tablePrixFinal$PRIX, data=tablePrixFinal, main="Distrubution_de_Prix", ylab="Prix")

```

On a obtenu la distribution de Prix:



PRIX	
Min.	: 8540
1st Qu.:	18310
Median :	37100
Mean :	44917
3rd Qu.:	70910
Max.	:101300

Nous avons divisé le prix en trois niveaux selon le valeur(j'ai pas encore trouvé la méthode de réaliser la meme fonction de update sous SQL, donc on a changé directement sous SQL):

```

1 update IMMATRICULATION set PRIX = 1 where PRIX <= 18310;
2 update IMMATRICULATION set PRIX = 2 where PRIX > 18310 and PRIX < 70910;
3 update IMMATRICULATION set PRIX = 3 where PRIX >= 70910;
4
5 update CATALOGUE set PRIX = 1 where PRIX <= 18310;
6 update CATALOGUE set PRIX = 2 where PRIX > 18310 and PRIX < 70910;
7 update CATALOGUE set PRIX = 3 where PRIX >= 70910;
8 commit;

```

Après recharger les tables, renommer le prix:

```

1 ##Changer
2 table(tableImPrix$PRIX)
3 ##J'ai pas trouve une methode qui nous permet de realiser la meme fonction de Update
4 tableImPrix <- within(tableImPrix,{
5   PRIX[PRIX == 3] <- "Luxe"
6   PRIX[PRIX == 2] <- "Moyen"
7   PRIX[PRIX == 1] <- "Economique"
8 })

```

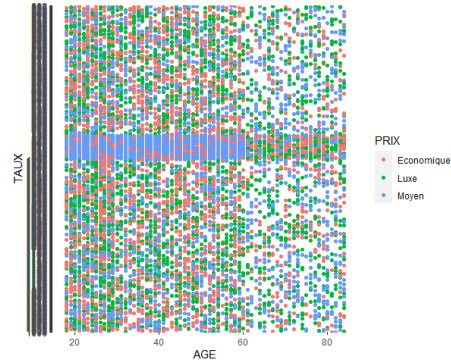
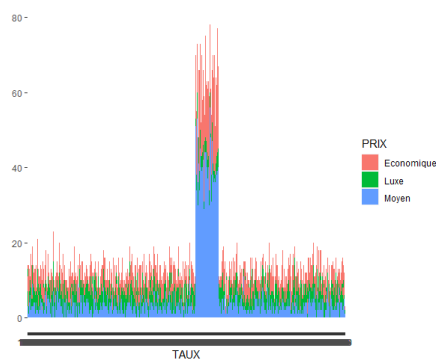
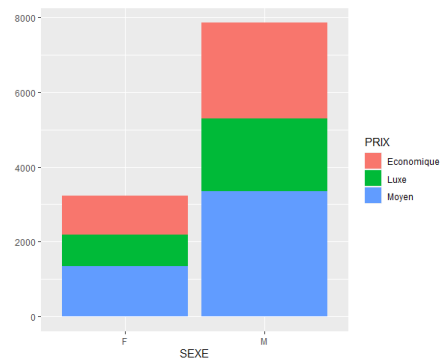
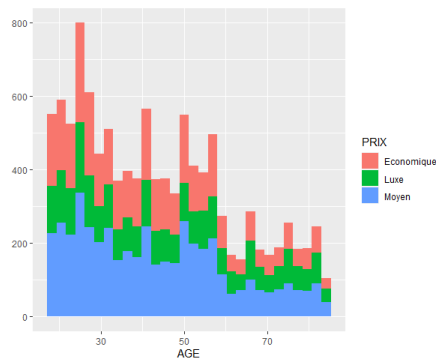
4.1.2 Visuellement

D'après notre jugement subjectif, les facteurs les plus importants affectant le prix d'une voiture sont age, sexe et taux(Capacité d'endettement du client en euros).

```

1 ## Visuellement
2 attach(tablePrixFinal)
3 tablePrixFinal <- subset(tablePrixFinal, select=-IMMATRICULATION)
4 qplot(AGE, data=tablePrixFinal, fill=PRIX)
5 qplot(SEXE, data=tablePrixFinal, fill=PRIX)
6 qplot(TAUX, data=tablePrixFinal, fill=PRIX)
7
8 table(SEXE,PRIX)
9
10 qplot(Age, TAUX, data=tablePrixFinal,color=PRIX)

```



4.1.3 Apprentissage supervisé grâce aux différents classifieurs

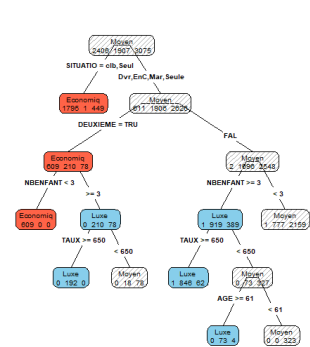
Nous allons tout d'abord choisir un arbre de decision:

Aprendissage

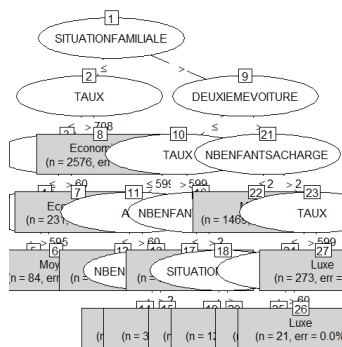
```

1  ### comparer arbre de decision
2  Prix_EA <- tablePrixFinal[1:7388,]
3  Prix_ET <- tablePrixFinal[7389:11082,]
4
5  ## Verifier tous les donnnes sont en bon format
6  str(Prix_EA)
7  Prix_EA$TAUX <- as.integer(Prix_EA$TAUX)
8  Prix_ET$TAUX <- as.integer(Prix_ET$TAUX)
9  str(Prix_EA)
10
11 ## Aprendissage
12 #rpart
13 Prixtree1 <- rpart(PRIX~, Prix_EA)
14 prp(Prixtree1, type=4, extra=1, box.col=c("tomato", "skyblue")[Prixtree1$frame$yval])
15
16 #C5.0
17 Prix_EA$PRIX <- as.factor(Prix_EA$PRIX)
18 Prixtree2 <- C5.0(PRIX~, Prix_EA)
19 plot(Prixtree2, type="simple")
20
21 #Classification and regression trees
22 Prixtree3 <- tree(PRIX~, data=Prix_EA)
23 plot(Prixtree3)
24 text(Prixtree3, pretty=0)

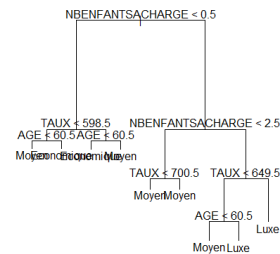
```



(e) a



(f) b



(g) c

Figure 10: Aprendissage: (a)rpart (b)C5.0 (c)Classification and regression trees

comparaison

```

1  #comparaison
2  predPrix.tree1 <- predict(Prixtree1, Prix_ET, type="class")
3  predPrix.tree2 <- predict(Prixtree2, Prix_ET, type="class")
4  predPrix.tree3 <- predict(Prixtree3, Prix_ET, type="class")
5  # Calcul des matrices de confusion

```



```

6 table(Prix_ET$PRIX, predPrix.tree1)
7 table(Prix_ET$PRIX, predPrix.tree2)
8 table(Prix_ET$PRIX, predPrix.tree3)

```

```

> table(Prix_ET$PRIX, predPrix.tree1)
      predPrix.tree1
      Economique Luxe Moyen
Economique      1187      1      1
Luxe              1     541     346
Moyen             250      32    1335
> table(Prix_ET$PRIX, predPrix.tree2)
      predPrix.tree2
      Economique Luxe Moyen
Economique      1074      1     114
Luxe              1     537     350
Moyen             115      1    1501
> table(Prix_ET$PRIX, predPrix.tree3)
      predPrix.tree3
      Economique Luxe Moyen
Economique       739      1     449
Luxe              94     551     243
Moyen             197      32    1388

```

Figure 11: matrices de confusion

ROC et Calcul de l'AUC

A cause de (**ROCR currently supports only evaluation of binary classification tasks.**), on calcule que le taux de reussi. Evidemment, C5.0 a la meilleure performance (0.842).

Ensuite, on utilise differents classifieurs suivants:

```

1  #-----#
2  # RANDOM FORESTS #
3  #-----#
4  # Apprentissage du classifieur de type foret aleatoire
5  rfPrix <- randomForest(PRIX~, Prix_EA)
6  # Test du classifieur : classe predite
7  rf_classPrix <- predict(rfPrix, Prix_ET, type="response")
8  # Matrice de confusion
9  table(Prix_ET$PRIX, rf_classPrix)
10 # Test du classifieur : probabilites pour chaque prediction
11 rf_probPrix <- predict(rfPrix, Prix_ET, type="prob")
12 # L'objet genere est une matrice
13 rf_probPrix
14 #-----#
15 # SUPPORT VECTOR MACHINES #
16 #-----#
17 # Apprentissage du classifieur de type svm
18 svmPrix <- svm(PRIX~, Prix_EA, probability=TRUE)
19 # Test du classifieur : classe predite
20 svm_classPrix <- predict(svmPrix, Prix_ET, type="response")
21 # Matrice de confusion
22 table(Prix_ET$PRIX, svm_classPrix)
23 # Test du classifieur : probabilites pour chaque prediction
24 svm_prob <- predict(svmPrix, Prix_ET, probability=TRUE)

```

```

25 # L'objet genere est de type specifique aux svm
26 svm_prob
27 # Recuperation des probabilites associees aux predictions
28 svm_prob <- attr(svm_prob, "probabilities")
29 # Conversion en un data frame
30 svm_prob <- as.data.frame(svm_prob)
31 #-----#
32 # NAIVE BAYES #
33 #-----#
34 # Apprentissage du classifieur de type naive bayes
35 nbPrix <- naive_bayes(PRIX~, Prix_EA)
36 nbPrix
37 # Test du classifieur : classe predite
38 nbPrix_class <- predict(nbPrix, Prix_ET, type="class")
39 nbPrix_class
40 table(nbPrix_class)
41 # Matrice de confusion
42 table( Prix_ET$PRIX, nbPrix_class)
43 # Test du classifieur : probabilites pour chaque prediction
44 nbPrix_prob <- predict(nbPrix, Prix_ET, type="prob")
45 # L'objet genere est une matrice
46 nbPrix_prob
47 #-----#
48 # NEURAL NETWORKS #
49 #-----#
50 # Apprentissage du classifieur de type perceptron monocouche
51 nnPrix <- nnet(PRIX~, Prix_EA, size=12)
52 nnPrix
53 # Test du classifieur : classe predite
54 nnPrix_class <- predict(nnPrix, Prix_ET, type="class")
55 nnPrix_class
56 table(nnPrix_class)
57 # Matrice de confusion
58 table(Prix_ET$PRIX, nnPrix_class)
59 # Test du classifieur : probabilites pour chaque prediction
60 nnPrix_prob <- predict(nnPrix, Prix_ET, type="raw")
61 # L'objet genere est un vecteur des probabilites de prediction
62 nnPrix_prob
63 #-----#
64 # K-NEAREST NEIGHBORS #
65 #-----#
66 # Apprentissage et test simultanes du classifieur de type k-nearest neighbors
67 knnPrix <- kknn(PRIX~, Prix_EA, Prix_ET)
68 # Resultat : classe predite et probabilites de chaque classe pour chaque instance de test
69 summary(knnPrix)
70 # Matrice de confusion
71 table(Prix_ET$PRIX, knnPrix$fitted.values)
72 # Conversion des probabilites en data frame
73 knnPrix_prob <- as.data.frame(knnPrix$prob)

```

```
> table(Prix_ET$PRIX, rf_classPrix)
      rf_classPrix
Econome 1184 1 4
Luxe 1 544 343
Moyen 247 14 1356
```

(a) RANDOM FORESTS

```
> table(Prix_ET$PRIX, svm_classPrix)
      svm_classPrix
Econome 1187 1 1
Luxe 1 516 371
Moyen 251 6 1360
```

(b) SUPPORT VECTOR MACHINES

```
> table(Prix_ET$PRIX, nbPriX_class)
      nbPriX_class
Econome 1060 103 26
Luxe 37 584 267
Moyen 301 269 1047
```

(c) NAIVE BAYES

```
> table(Prix_ET$PRIX, nnPriX_class)
      nnPriX_class
Moyen 1189
Econome 888
Luxe 1617
```

(d) NEURAL NETWORKS

```
> table(Prix_ET$PRIX, knnPriX$fitted.values)
      Econome Luxe Moyen
Econome 1105 1 83
Luxe 3 631 254
Moyen 142 203 1272
```

(e) K-NEAREST NEIGHBORS

Figure 12: Résultat des différents classifieurs

Taux de réussite:

Modèle	Taux de réussite
arbre de decision	0.842
RANDOM FORESTS	0.835
SUPPORT VECTOR MACHINES	0.829
NAIVE BAYES	0.728
K-NEAREST NEIGHBORS	0.814

4.1.4 Application de la méthode

On choisit l'arbre de décision (C5.0) et on va appliquer cette méthode:

```
1 #-----#
2 # APPLICATION DE LA METHODE arbre de decision (C5.0) #
3 #-----#
4 # Visualisation des donnees a predire
5 View(tableMar)
6
7 #=== C5.0 ===#
8 class.treeC50 <- predict(Prixtree2, tableMar, probability=TRUE)
9 # L'objet genere est de type specifique aux svm
10 class.treeC50
11 # Recuperation des probabilites associees aux predictions
12 prob.treeC50 <- attr(class.treeC50, "probabilities")
13 # Conversion en un data frame
14 prob.treeC50 <- as.data.frame(prob.treeC50)
15 resultatPrix <- data.frame(tableMar$ID, class.treeC50, prob.treeC50)
16
17 #=== ARBRE DE DECISION C5.0 ===#
18 class.treeC50 <- predict(Prixtree2, tableMar, type="class")
19 prob.treeC50 <- predict(Prixtree2, tableMar, type="prob")
20 resultatPrix <- data.frame(tableMar, class.treeC50, prob.treeC50)
21 resultatPrix <- data.frame(tableMar, class.treeC50)
22
23 # Renommage de la colonne des classes predites
24 names(resultatPrix)[7] <- "PRIX"
25
26 #-----#
27 # ENREGISTREMENT DES PREDICTIONS #
28 #-----#
```

```

29 # Enregistrement du fichier de resultats au format csv
30 write.table(resultat1, file='predictions.csv', sep="\t", dec=".", row.names = F)

```

Enfin, on a obtenu:

	AGE	SEXE	TAUX	SITUATIONFAMILIALE	NBENFANTSACHARGE	DEUXIEMEVOITURE	PRIX
1	21	F	1396	celibataire	0	FALSE	Economique
2	59	F	572	En Couple	2	FALSE	Moyen
3	64	M	559	celibataire	0	FALSE	Economique
4	79	F	981	En Couple	2	FALSE	Moyen
5	55	M	588	celibataire	0	FALSE	Moyen
6	34	F	1112	En Couple	0	FALSE	Moyen
7	58	M	1192	En Couple	0	FALSE	Moyen
8	35	M	589	celibataire	0	FALSE	Moyen
9	59	M	748	En Couple	0	TRUE	Economique

Figure 13: Le modèle de la prédiction du PRIX

- 4.2 Le modèle de la prédiction du LONGUEUR(ZHAO YUE)**
- 4.3 Le modèle de la prédiction du NBPORTES(CHAYMAE)**
- 4.4 Le modèle de la prédiction du COULEUR(LUO YAJUAN)**

4.5 Le modèle de la prédiction du OCCASION

4.5.1 Fusion des données

```

1 tableImOCCA <- tableIm.f[c(7, 9)]
2 tableOCCAFinal <- merge(tableClients, tableImOCCA, by = "IMMATRICULATION", incomparables = NA)
3 #requete pour la distribution des donnees: OCCASION
4 summary(tableOCCAFinal$OCCASION)
5 ##changer les types de donnees
6 str(tableOCCAFinal)
7 tableOCCAFinal$TAUX <- as.integer(tableOCCAFinal$TAUX)

```

4.5.2 Visuellement

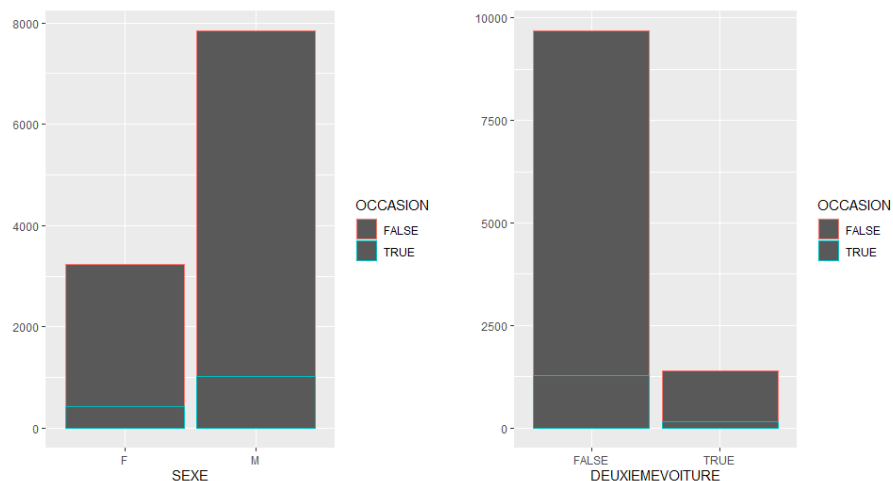
On va essayer de trouver les relations parmi les paramètres:

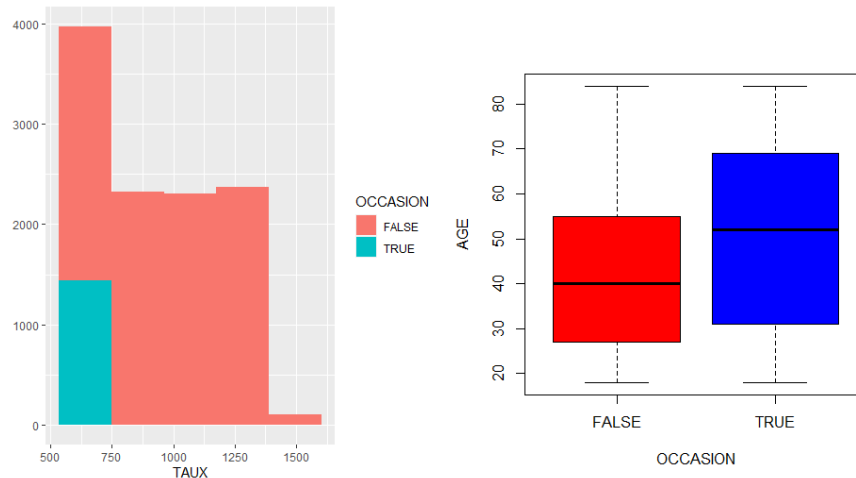
```

1 ##On trouve qu'ils y moins de voitures occasions
2 library(ggplot2)
3 qplot(OCCASION, data=tableOCCAFinal)
4 table(tableOCCAFinal$DEUXIEMEVOITURE, tableOCCAFinal$OCCASION)
5 qplot(DEUXIEMEVOITURE, data=tableOCCAFinal, color=OCCASION)
6 qplot(TAUX, data=tableOCCAFinal, fill=OCCASION, bins =5)
7 boxplot(AGE~OCCASION, data=tableOCCAFinal, col=c("red", "blue"))
8 qplot(SEXE, data=tableOCCAFinal, color=OCCASION)

```

Comme dans les figures au dessous, nous avons visualisé les relations entre le sexe, la deuxième voiture, l'âge et l'état de l'occasion. Pour les relations entre les sexes et l'état de l'occasion ou bien entre si c'est la deuxième voiture et l'état de l'occasion, nous n'avons pas pu trouver une règle ou bien une tendance. Dans figure 3, nous trouvons que les personnes qui ont bonne capacités d'endettements achètent moins de voitures d'occasions. Dans figure 4, nous trouvons que Les voitures d'occasion ont un meilleur marché auprès des jeunes.





4.5.3 Apprentissage supervisé grâce aux différents classifieurs

Nous allons tout d'abord choisir un arbre de decision:

Aprendissage

```

1 #-----#
2 # Apprentissage #
3 #-----#
4 tableOCCAFinal <- subset(tableOCCAFinal, select=-IMMATRICULATION)
5 OCCA_EA <- tableOCCAFinal[1:7388,]
6 OCCA_ET <- tableOCCAFinal[7389:11082,]
7
8 #3 arbres de decision
9 OCCAtree1 <- rpart(OCCASION~., OCCA_EA)
10 OCCA_EA$OCCASION <- as.factor(OCCA_EA$OCCASION)
11 OCCAtree2 <- C5.0(OCCASION~., OCCA_EA)
12 OCCAtree3 <- tree(OCCASION~., data=OCCA_EA)
13 #RANDOM FORESTS
14 rfOCCA <- randomForest(OCCASION~., OCCA_EA)
15 # SUPPORT VECTOR MACHINES
16 svmOCCA <- svm(OCCASION~., OCCA_EA, probability=TRUE)
17 # NAIVE BAYES
18 nbOCCA <- naive_bayes(OCCASION~., OCCA_EA)
19 # NEURAL NETWORKS
20 nnOCCA <- nnet(OCCASION~., OCCA_EA, size=12)
21 # K-NEAREST NEIGHBORS
22 knnOCCA <- kkn(OCCASION~., OCCA_EA, OCCA_ET)

```

Text des classifieurs et matrice de confusion

```

1 ##Text des arbres et matrice de confusion
2 predOCCA.tree1 <- predict(OCCAtree1, OCCA_ET, type="class")
3 predOCCA.tree2 <- predict(OCCAtree2, OCCA_ET, type="class")
4 predOCCA.tree3 <- predict(OCCAtree3, OCCA_ET, type="class")
5 # Calcul des matrices de confusion
6 table(OCCA_ET$OCCASION, predOCCA.tree1)
7 table(OCCA_ET$OCCASION, predOCCA.tree2)

```

```

8 table(OCCA_ET$OCCASION, predOCCA.tree3)
9 #####
10 # FALSE TRUE
11 #FALSE 3189 37
12 #TRUE 290 178
13 ##Text des classifieurs et matrice de confusion
14 result.rfOCCA <- predict(rfOCCA,OCCA_ET, type="response")
15 table(OCCA_ET$OCCASION, result.rfOCCA)
16 #####result.rfOCCA
17 #####FALSE TRUE
18 #FALSE 3181 45
19 #TRUE 290 178
20 result.svmOCCA <- predict(svmOCCA,OCCA_ET, type="response")
21 table(OCCA_ET$OCCASION, result.svmOCCA)
22 #####result.svmOCCA
23 #####FALSE TRUE
24 #FALSE 3186 40
25 #TRUE 307 161
26 result.treeNaiveOCCA <- predict(nbOCCA,OCCA_ET, type="class")
27 table(OCCA_ET$OCCASION, result.treeNaiveOCCA)
28 #####result.treeNaiveOCCA
29 #####FALSE TRUE
30 #FALSE 2518 708
31 #TRUE 2 466
32 result.treeNnetOCCA <- predict(nnOCCA, OCCA_ET,type="class")
33 table(OCCA_ET$OCCASION, result.treeNnetOCCA)
34 #####result.treeNnetOCCA
35 #####FALSE
36 #FALSE 3226
37 #TRUE 468
38 table(OCCA_ET$OCCASION, knnOCCA$fitted.values)
39 #####FALSE TRUE
40 #FALSE 3020 206
41 #TRUE 228 240

```

CALCUL DE COURBES ROC

```

1 # Test du classifieur : probabilités pour chaque prediction
2 p.treeRpartOCCA <- predict(OCCAtree1, OCCA_ET, type="prob")
3 # Courbe ROC
4 roc.predOCCA1 <- prediction(p.treeRpartOCCA[,2], OCCA_ET$OCCASION)
5 roc.perfOCCA1 <- performance(roc.predOCCA1,"tpr", "fpr")
6
7 p.treec50OCCA <- predict(OCCAtree2, OCCA_ET, type="prob")
8 # Courbe ROC
9 roc.predOCCA2 <- prediction(p.treec50OCCA[,2], OCCA_ET$OCCASION)
10 roc.perfOCCA2 <- performance(roc.predOCCA2,"tpr", "fpr")
11
12 p.treeTreeOCCA <- predict(OCCAtree3, OCCA_ET, type="vector")
13 # Courbe ROC
14 roc.predOCCA3 <- prediction(p.treeTreeOCCA[,2], OCCA_ET$OCCASION)
15 roc.perfOCCA3 <- performance(roc.predOCCA3,"tpr", "fpr")
16
17 rf_probOCCA <- predict(rfOCCA, OCCA_ET, type="prob")

```



```

18 # Courbe ROC
19 roc.predOCCA4 <- prediction(rf_probOCCA[,2], OCCA_ET$OCCASION)
20 roc.perfOCCA4 <- performance(roc.predOCCA4,"tpr","fpr")
21
22 svm_probOCCA <- predict(svmOCCA, OCCA_ET, probability=TRUE)
23 # Recuperation des probabilites associees aux predictions
24 svm_probOCCA <- attr(svm_probOCCA, "probabilities")
25 # Conversion en un data frame
26 svm_probOCCA <- as.data.frame(svm_probOCCA)
27 # Courbe ROC sur le meme graphique
28 roc.predOCCA5 <- prediction(svm_probOCCA[,2], OCCA_ET$OCCASION)
29 roc.perfOCCA5 <- performance(roc.predOCCA5,"tpr","fpr")
30
31 nb_probOCCA <- predict(nbOCCA, OCCA_ET, type="prob")
32 # Courbe ROC
33 roc.predOCCA6 <- prediction(nb_probOCCA[,2], OCCA_ET$OCCASION)
34 roc.perfOCCA6 <- performance(roc.predOCCA6,"tpr","fpr")
35
36 nn_probOCCA <- predict(nnOCCA, OCCA_ET, type="raw")
37 # Courbe ROC
38 roc.predOCCA7 <- prediction(nn_probOCCA[,1], OCCA_ET$OCCASION)
39 roc.perfOCCA7 <- performance(roc.predOCCA7,"tpr","fpr")
40
41 # Conversion des probabilites en data frame
42 knn_probOCCA <- as.data.frame(knnOCCA$prob)
43 # Courbe ROC
44 roc.predOCCA8 <- prediction(knn_probOCCA[,2], OCCA_ET$OCCASION)
45 roc.perfOCCA8 <- performance(roc.predOCCA8,"tpr","fpr")

```

CALCUL DE L INDICE AUC

```

1 auc.treeOCCA1 <- performance(roc.predOCCA1, "auc")
2 auc.treeOCCA2 <- performance(roc.predOCCA2, "auc")
3 auc.treeOCCA3 <- performance(roc.predOCCA3, "auc")
4 auc.treeOCCA4 <- performance(roc.predOCCA4, "auc")
5 auc.treeOCCA5 <- performance(roc.predOCCA5, "auc")
6 auc.treeOCCA6 <- performance(roc.predOCCA6, "auc")
7 auc.treeOCCA7 <- performance(roc.predOCCA7, "auc")
8 auc.treeOCCA8 <- performance(roc.predOCCA8, "auc")
9
10 attr(auc.treeOCCA1, "y.values")
11 attr(auc.treeOCCA2, "y.values")
12 attr(auc.treeOCCA3, "y.values")
13 attr(auc.treeOCCA4, "y.values")
14 attr(auc.treeOCCA5, "y.values")
15 attr(auc.treeOCCA6, "y.values")
16 attr(auc.treeOCCA7, "y.values")
17 attr(auc.treeOCCA8, "y.values")

```

Choisissons un classifieur qui a la meilleur performance

Après les calculs, nous allons choisir RANDOM FORESTS comme notre modèle d'apprentissage.

Modèle	L'INDICE AUC
arbre de decision	0.9241559
RANDOM FORESTS	0.9247706
SUPPORT VECTOR MACHINES	0.9208312
NAIVE BAYES	0.9246225
NEURAL NETWORKS	0.5
K-NEAREST NEIGHBORS	0.9033746

Après avoir comparé différents arbres de décision, nous avons constaté qu'ils avaient la même structure. Donc ils ont la même matrice de confusion, la même courbe ROC et la même l'indice AUC. En fait, dans 4.5.2, nous avons déjà trouvé les deux facteurs les plus influents: TAUX et AGE.

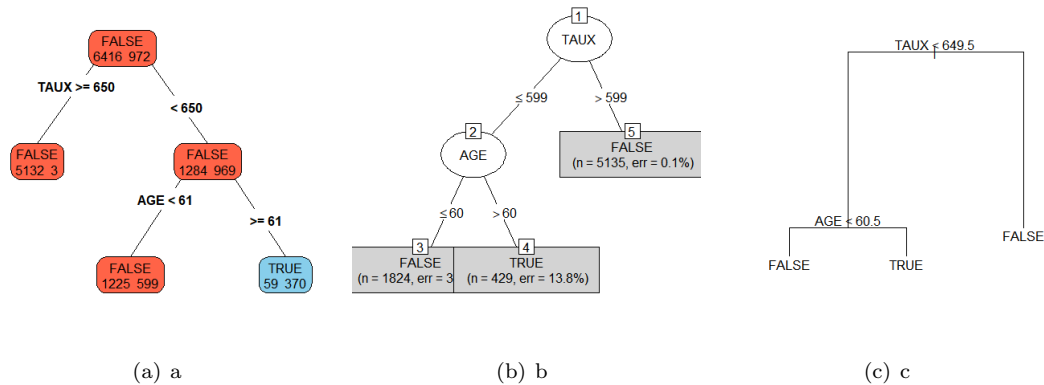


Figure 14: (a)rpart (b)C5.0 (c)Classification and regression trees

Nous pouvons aussi tracer la courbe ROC:

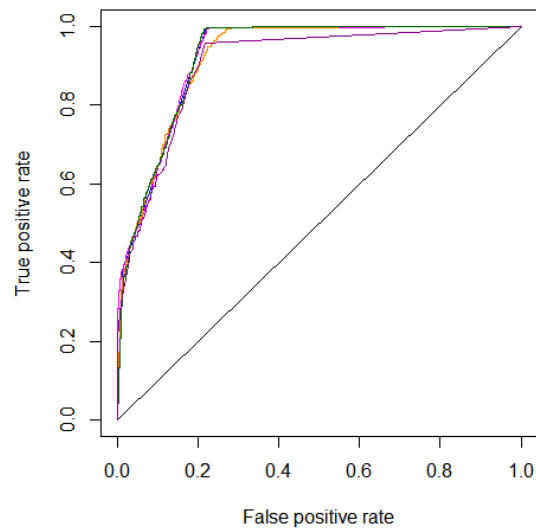


Figure 15: La courbe ROC

4.5.4 Application de la méthode

```

1 #-----#
2 # APPLICATION DE LA METHODE RANDOM FORESTS #
3 #-----#
4 # Visualisation des donnees a predire
5 View(tableMar)
6
7 #=== RANDOM FORESTS ===#
8 class.trerFOCCA <- predict(rfOCCA, tableMar, probability=TRUE)
9 resultatOCCA <- data.frame(tableMar, class.trerFOCCA)
10
11 # Renommage de la colonne des classes predites
12 names(resultatOCCA)[7] <- "OCCASION"

```

Alors nous avons le résultat pour occasion:

	AGE	SEXE	TAUX	SITUATIONFAMILIALE	NBNFANTSACHARGE	DEUXIEMEVOITURE	OCCASION
1	21	F	1396	celibataire	0	FALSE	FALSE
2	59	F	572	En Couple	2	FALSE	FALSE
3	64	M	559	celibataire	0	FALSE	TRUE
4	79	F	981	En Couple	2	FALSE	FALSE
5	55	M	588	celibataire	0	FALSE	FALSE
6	34	F	1112	En Couple	0	FALSE	FALSE
7	58	M	1192	En Couple	0	FALSE	FALSE
8	35	M	589	celibataire	0	FALSE	FALSE
9	59	M	748	En Couple	0	TRUE	FALSE

Figure 16: Le modèle de la prédiction de OCCASION

5 Conclusion

5.1 Erreurs

Après la dernière fusion, on obtient `tableFinal` qui contient plus de lignes que `tableClients`. C-à-d, il y a encore des erreurs à nettoyer. Mais nous allons ignorer ces erreurs (0.1% n'influence pas)

Data		
allTables	5 obs. of 2 variables	
conn	Formal class JDBCConnection	
drv	Formal class JDBCDriver	
tableCatalogue	270 obs. of 10 variables	
tableClients	11065 obs. of 7 variables	
tableFinal	11082 obs. of 7 variables	
tableIm	1048575 obs. of 10 variables	
tableIm_f	1048575 obs. of 2 variables	
tableMar	9 obs. of 6 variables	

5.2 NBPLACES

```
> summary(tableIm_f)
```

MARQUE	NOM	PUISSANCE
Length:1048575	Length:1048575	Min. : 55.0
Class :character	Class :character	1st Qu.: 75.0
Mode :character	Mode :character	Median :150.0
		Mean :198.9
		3rd Qu.:245.0
		Max. :507.0

LONGUEUR	NBPORTES	COULEUR
Length:1048575	Min. :3.000	Length:1048575
Class :character	1st Qu.:5.000	Class :character
Mode :character	Median :5.000	Mode :character
	Mean :4.868	
	3rd Qu.:5.000	
	Max. :5.000	

OCCASION	PRIX	IMMATRICULATION	NBPLACES.x
Length:1048575	Min. : 7500	Length:1048575	Min. :5
Class :character	1st Qu.: 18310	Class :character	1st Qu.:5
Mode :character	Median : 25970	Mode :character	Median :5
	Mean : 35767		Mean :5
	3rd Qu.: 49200		3rd Qu.:5
	Max. :101300		Max. :5

NBPLACES.y	CATALOGUEID
Min. :5	Min. : 1.0
1st Qu.:5	1st Qu.: 78.0
Median :5	Median :165.0
Mean :5	Mean :154.7
3rd Qu.:5	3rd Qu.:234.0
Max. :5	Max. :270.0

Y a que valeur 5 dans ce champs !!! Nous devrions nous demander s'il s'agit d'un problème de base de données, s'il s'agit d'un problème de base de données, nous devrions essayer de restaurer ou de supprimer cette variable directement.

5.3 La manque de la mélange de la base de données

Quand nous avons séparé notre base en 3 tiers et les 2 premiers tiers pour l'apprentissage et le dernier tier pour la prédiction, cela causera l'abaissement de la fiabilité du cadre (Les anciennes données ne sont pas aussi fiables que les données actuelles). En plus, nous devons répéter le processus Echantillonnage - Apprentissage - Prédiction plusieurs fois jusqu'à la performance de notre classifieur soit stable et le meilleur.