ESTIA/MBDS

Projet Data Analytics

Victor AZALBERT / Jérôme CHAMBORD / Justin LASSALLE

21/10/2020

Table des matières

Objectifs

L'objectif est de construire un modèle de prédiction de la catégorie de véhicules (ou du modèle de véhicule) la plus susceptible de convenir à un client en fonction de ses caractéristiques (âge, sexe, statut marital, nombre d'enfants, etc.). Les principales étapes consisteront à :

- Répartir les véhicules et/ou les clients en différentes catégories correspondant chacune à différents besoins.
- Mettre au point un modèle de prédiction de la catégorie de véhicules qui répondent aux besoins des clients à l'aide des approches de classification supervisée.

Méthode Utilisée

1. Analyse Exploratoire des Données
2. Application des catégories de véhicules définies au fichier *Immatriculations.csv*
3. Fusion des fichiers *Clients.csv* et *Immatriculations.csv*
4. Création d'un modèle de classification supervisée pour la prédiction de la catégorie de véhicules

Travail Effectué

1. Analyse exploratoire des données

1.1. Sauvegarde des Données dans Oracle

La première étape a été la connexion à notre base de données Oracle, avec l'utilisation des drivers mis à dispositions. (Extrait issu du script extractionOracle.R)

```
##classPath : add path to drivers jdbc

drv <- RJDBC::JDBC(driverClass = "oracle.jdbc.OracleDriver", classPath =
                     Sys.glob("C:/Users/v.azalbert/Documents/Semestre_5/DataScience/drivers/*"))


#Connexion OK
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)))",
                  "AZALBERT2B20", "AZALBERT2B2001")
```

Ensuite nous avons extraits les données des fichiers csv et les écrire dans des tables de notre pdb Oracle. Exemple avec le fichier catalogue.csv (Extrait issu du script extractionOracle.R) :

```
#Enregistrement de la table Catalogue dans la DB Oracle
catalogue <- read.csv("../DATA/data_initial/Catalogue.csv", header = TRUE,
                      sep = ",", dec = ".")


dbWriteTable(conn,"Catalogue",catalogue,
             rownames=FALSE, overwrite = TRUE, append = FALSE)
```

1.2. Nettoyage des Données

1.2.1 Nettoyage des Données sous SQL

1.2.2 Nettoyage des Données sous R

Nous avons nettoyé les données de la table Client_8 sous R, pour cela nous avons créer une table client en lisant le fichier csv. Nous affichons les données récupérées :

```
> client <- read.csv("../DATA/data_initial/Clients_8.csv", header = TRUE, sep = ",", dec = ".")
> #lecture du data_frame
> str(client)
'data.frame':   100000 obs. of  7 variables:
 $ age               : chr  "20" "59" "21" "21" ...
 $ sexe              : chr  "M" "F" "M" "M" ...
 $ taux              : chr  "1010" "422" "207" "409" ...
 $ situationFamiliale: chr  "En Couple" "En Couple" "En Couple" "En Couple" ...
 $ nbEnfantsACharge  : chr  "4" "4" "0" "4" ...
 $ X2eme.voiture     : chr  "false" "false" "false" "false" ...
 $ immatriculation   : chr  "7396 VP 43" "7546 VN 65" "5235 IZ 58" "3303 QQ 51" ...
>
```

Nous pouvons constater que toutes les variables sont en char pour remédier à cela, nous exécutons ces lignes de codes qui permet la conversion :

```
> client$age <- as.integer(client$age)
Warning message:
NAs introduits lors de la conversion automatique
> client$taux <- as.integer(client$taux)
Warning message:
NAs introduits lors de la conversion automatique
> client$situationFamiliale <- as.factor(client$situationFamiliale)
> client$nbEnfantsACharge <- as.integer(client$nbEnfantsACharge)
Warning message:
NAs introduits lors de la conversion automatique
> client$X2eme.voiture <- as.logical(client$X2eme.voiture)
> #lecture du data_frame
> str(client)
'data.frame':   100000 obs. of  7 variables:
 $ age               : int  20 59 21 21 26 39 69 29 23 34 ...
 $ sexe              : chr  "M" "F" "M" "M" ...
 $ taux              : int  1010 422 207 409 500 201 1068 485 1339 248 ...
 $ situationFamiliale: Factor w/ 9 levels " ","?","Célibataire",..: 5 5 5 5 3 5 5 3 5 5 ...
 $ nbEnfantsACharge  : int  4 4 0 4 0 0 0 0 1 2 ...
 $ X2eme.voiture     : logi  FALSE FALSE FALSE FALSE FALSE TRUE ...
 $ immatriculation   : chr  "7396 VP 43" "7546 VN 65" "5235 IZ 58" "3303 QQ 51" ...
>
```

Grâce à la librairie FunModeling, nous pouvons afficher le nombre de NA et de zéro présent dans les variables.

```
> #visualiser les valeurs nulles et NA
> df_status(client)
             variable q_zeros p_zeros q_na p_na q_inf p_inf      type unique
1                 age       0    0.00  170 0.17     0     0   integer     68
2               sexe        0    0.00    0 0.00     0     0 character      9
3               taux        0    0.00  226 0.23     0     0   integer   1001
4 situationFamiliale        0    0.00    0 0.00     0     0    factor      9
5   nbEnfantsACharge    45092   45.09  199 0.20     0     0   integer      6
6      X2eme.voiture    86883   86.88  223 0.22     0     0   logical      2
7    immatriculation        0    0.00    0 0.00     0     0 character  99985
>
```

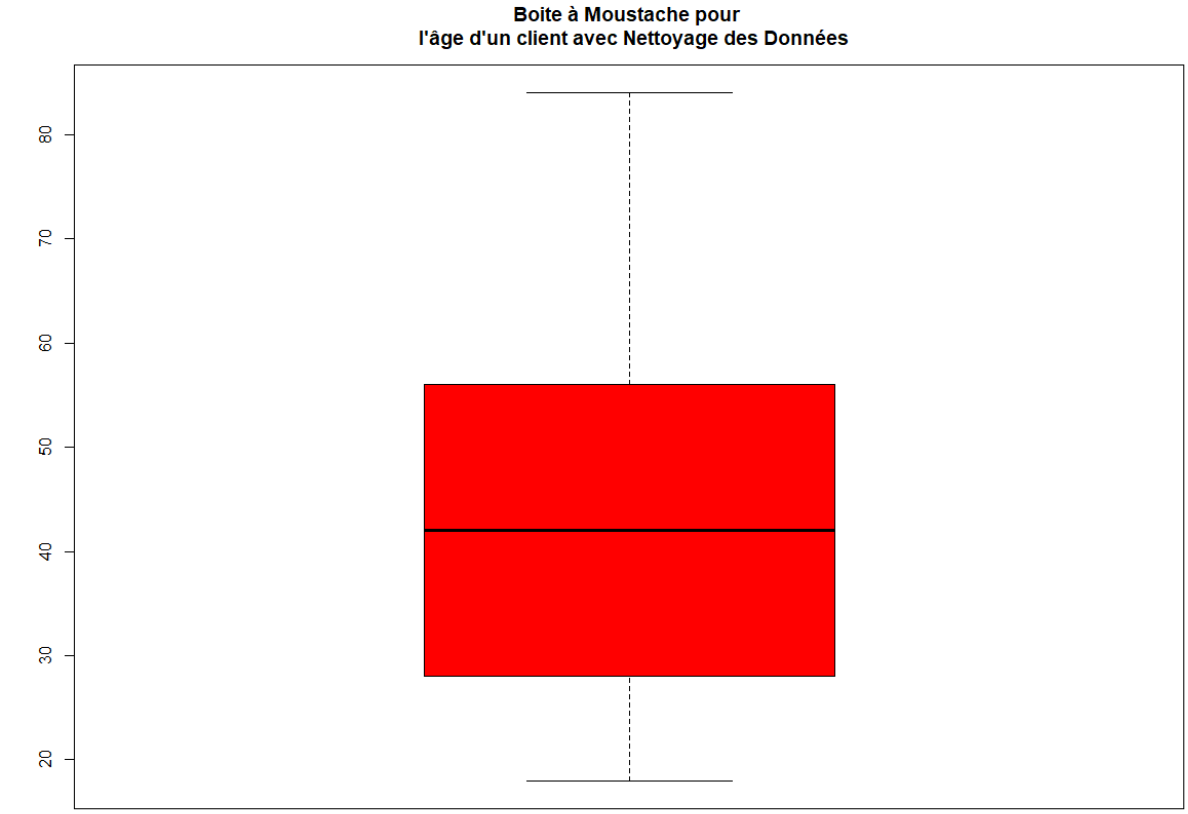
Nous avons décider de supprimer les valeurs avec des NA présents car elles nous sont d'aucune utilité et que leur nombre n'est pas significatif. Pour cela, nous créer une table client_Sans_NA.

```
> #Suppression des NA
> client_Sans_NA <- na.omit(client)
> summary(client_Sans_NA)
      age            sexe                taux          situationFamiliale nbEnfantsACharge X2eme.voiture
 Min.   :-1.00   Length:99185       Min.   :  -1.0   En Couple  :63459    Min.   :-1.000   Mode :logical
 1st Qu.:28.00   Class :character   1st Qu.: 420.0   Célibataire:29648    1st Qu.: 0.000   FALSE:86359
 Median :42.00   Mode  :character   Median : 520.0   Seule      : 4784    Median : 1.000   TRUE :12826
 Mean   :43.66                      Mean   : 606.4   Marié(e)   :  615    Mean   : 1.246
 3rd Qu.:56.00                      3rd Qu.: 823.0   Seul       :  285    3rd Qu.: 2.000
 Max.   :84.00                      Max.   :1399.0   ?          :  124    Max.   : 4.000
                                                     (Other)    :  270

 immatriculation
 Length:99185
 Class :character
 Mode  :character
```

Le summary de cette table, nous montre des valeurs aberrantes, comme par exemple -1 pour l'âge. Nous allons donc récupérer tous les index ou les valeurs de l'âge est compris entre -1 et 17ans car à cet âge, il est impossible de posséder une voiture.

```
#Suppression des valeur aberrantes, âge négatifs jusqu'à majorité
outliers <- c(-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17)
outlier_idx <-which(client_Sans_NA$age %in% outliers)
clients_nettoyes <- client_Sans_NA[-outlier_idx,]
```



Nous vérifions grâce à une boite à moustache qu'il n'existe plus de valeurs aberrantes et nous effectuons le même processus pour toutes les variables qui en ont besoin.

1.3. Chargement des Données dans R

1.4. Reformatage des Données dans R

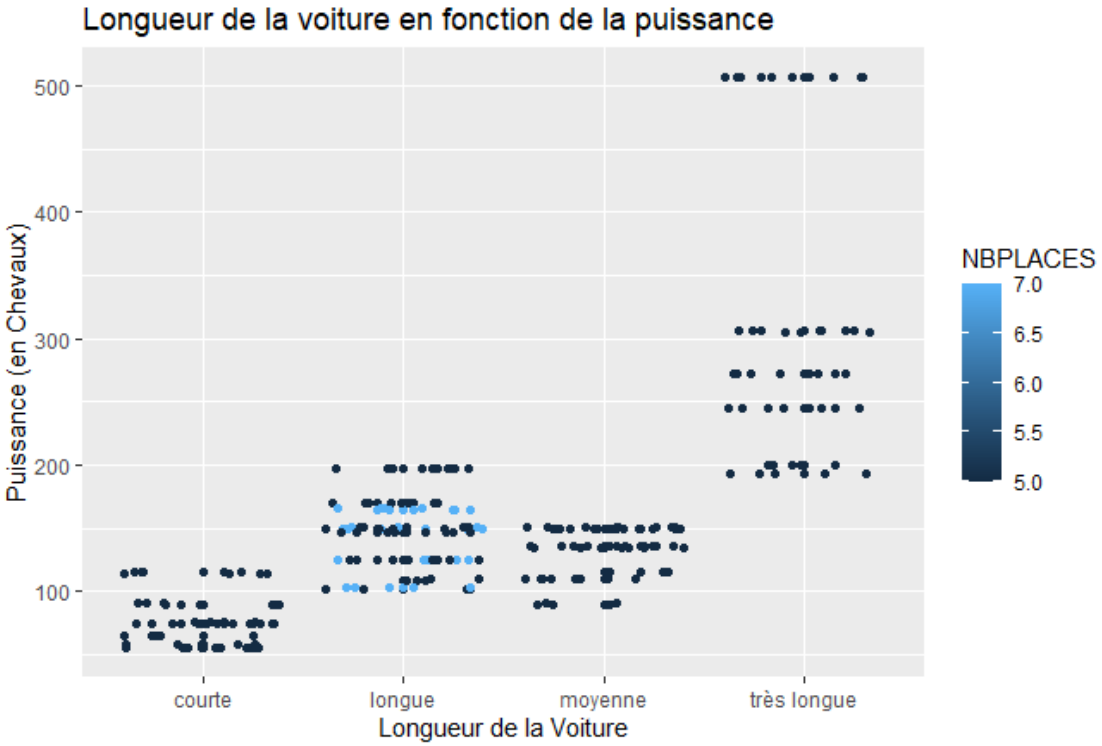
- Chargement des données : charger les fichiers .xls
  et points bonus si : chargement des fichiers de données dans la base Oracle (création des tables dans Oracle). Réaliser la connexion avec R via les drivers comme vu en cours et charger les données dans R. Une étape supplémentaire serait de réaliser le nettoyage de données sous SQL avant le chargement des données dans R.

L'analyse exploratoire des données vous permettra d'identifier d'éventuels problèmes dans les données (valeurs incohérentes, codage des valeurs manquantes, etc.) et découvrir d'éventuelles propriétés de l'espace des données (valeurs doublons, variables liées, variables d'importance particulière ou bien inutiles, etc.).

Attention : l'étape d'analyse exploratoire est ici importante car il existe des données manquantes qui nécessitent un pré-traitement / une transformation pour compléter les données dont vous aurez besoin.

Appliquez pour cela les différentes méthodes d'analyse exploratoire des données vues en cours (Statistiques descriptives, histogrammes, nuages de points, boîtes à moustaches, etc.). 2) Identification des catégories de véhicules :
Vous devez à partir des informations dans le fichier *Catalogue.csv* identifier des catégories de véhicules (citadine, routière, sportive, etc.) en fonction de leur taille, puissance, prix, etc. Ces catégories doivent correspondre à divers besoins de la part des clients (une grande voiture pour les familles nombreuses, une petite voiture pour circuler en ville, etc.). Ces catégories de véhicules constitueront les classes à prédire durant les étapes suivantes du processus.

Classification des Types de Voitures du Catalogue



Nous estimons que pour classer des voitures en plusieurs catégories, les trois critères suivants suffisent :

- La longueur du véhicule (courte, moyenne, longue, très longue)

- La puissance (min= 55, max= 507) : nous avons fixé des intervalles de puissance selon les modèles.
- Le nombre de places (min=5, Max=7)

Nous estimons que le nombre de places n'influe pas sur la classification des modèles, en effet tous les modèles comportant 3 portes font partis des véhicules courts. Ainsi la longueur du véhicule suffit pour la classification.

Type de Véhicule	Longueur du Véhicule	Intervalle de Puissance	Nombre de Places
Citadine	Courte	[55-90]	5
Citadine Sportive	Courte	[90-]	5
Routière	Moyenne	[55-100]	5
Routière Sportive	Moyenne	[100-140]	5
Routière Ultra Sportive	Moyenne	[140 -]	5
Berline	Très Longue	[55-150]	5
Berline Sportive	Très Longue	[150-250]	5
Berline Ultra Sportive	Très Longue	[250 -]	5
Familiale	Longue	[55-100]	5-7
Familiale Sportive	Longue	[100-]	5-7