# 

# Analyse exploratoire des données

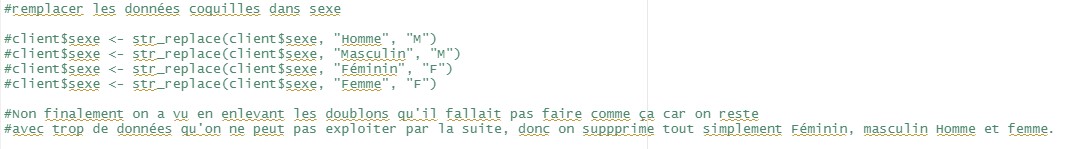
L’analyse exploratoire des données nous permet d’identifier des problèmes dans les données (valeurs incohérentes, codage des valeurs manquantes, etc.) et découvrir d’éventuelles propriétés de l’espace des données (valeurs doublons, variables liées, variables d’importance particulière ou bien inutiles,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

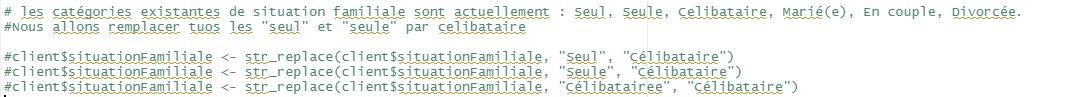
**Après avoir chargé les librairies qui nous serons utiles tout au long du projet et après avoir créé les tables sur Oracle puis nos dataframes comme expliqué dans le rapport du projet, nous nous sommes mis sur le tri et nettoyage des données des différents data frame.**

En premier lieu, nous avons fait un tri en parcourant toutes les données de client colonnes par colonnes. Nous avons remarqué de nombreuses incohérences :

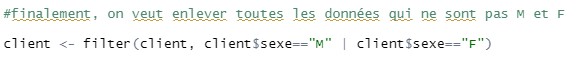
Pour le sexe, il y avait Masculin M et Homme par exemple, nous avons souhaité regrouper ces 3 données en une seule car nous étions sur du sexe.

Pareil pour Célibataire, seul et seule, nous avons pris l’initiative de tout regrouper car cela nous semblait plus cohérent que d’avoir 3 noms qui désigne la même situationFamiliale.

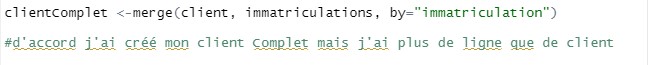




Finalement, après relecture du sujet nous nous sommes rendu compte que le CDC nous demandés des données précises que nous avons par la suite respecté. Donc chaque donnée que nous ne connaissions pas (ex : Homme) nous les avons supprimés.



Nous avons ensuite, fusionnés les fichiers Clients et immatriculations afin de vérifier les doublons. Sans surprise il y en a plusieurs car les totaux de clientComplet et Client sont différents.



ATTENTION AUX DOUBLONS

Pour clients dans les immatriculations - 18 doublons.

1er doublon

Client [client$immatriculation == "1557 AB 48",]

La même immatriculation appartient à 2 personnes totalement différente, voyons si dans immatriculations elle est double.

Nous voyons qu’elle correspond à 2 voitures différentes, ce qui est un problème car après la liaison dans client complet : 1 immatriculation créer 4 lignes dans client complet.

Ce qui fait que les 18 doublons d'immatriculations dans client créent 18\*4 lignes = 72 lignes en plus dans client Complet.



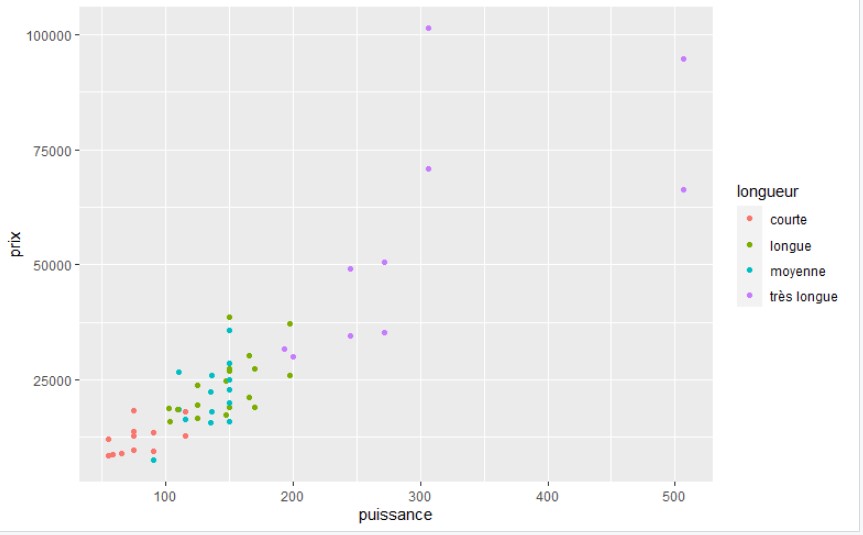
On a pu supprimer tous les doublons et donc on a re fusionné les fichiers.

# Application des catégories de véhicules

Pour entamer la fabrication des catégories de véhicules, nous avons tracer plusieurs nuages de points afin de voir s'il y a des groupes de voiture qui se forment “naturellement”. Sans surprise, la longueur et la puissance sont les critères les plus “utiles”.

Nous nous sommes essentiellement basés sur ce qplot car c’était le plus « parlant » pour faire nos catégories :

>qplot(longueur, nbPortes, data=catalogue)



Nous décidons alors de faire comme critères :

#citadines : les courtes

#sport : +de 300cv

#berline compact : moyennes

#berline : des longues mais pas de 7places

#berline confort : très longue mais supérieur à 190 et inférieur à 300

Nous remarquons des incohérences, par exemple : new beetle ce n'est pas 5 places mais 4 donc pas dans la bonne catégorie car on ne peut pas la conseiller à des familles (trop serrées à l'arrière), donc cette voiture irait plus dans la catégorie citadine

Nous nous rendons compte qu'il n'y a aucune voiture de 7 places, devons-nous supposer que les familles de 4 enfants choisiront automatiquement un monospace ?

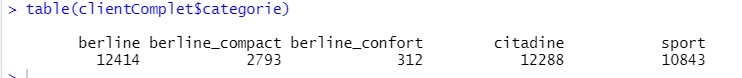
Oui du coup.

# Fusion des fichiers Clients.csv et Immatriculations.csv

Après avoir créé les catégories, nous refusionnions nos fichiers.



On vérifie que les catégories choisies sont cohérentes (pas toutes les voitures dans une seule catégorie par ex)



# Création d’un modèle de classification supervisée

Suppression des colonnes pas utiles, toutes sauf celles qui correspondent aux clients.

A la suite de pas mal d’erreurs rencontrées, nous avons décidés de supprimer la colonne taux car l’intervalle des taux était trop grand et nous posait des problèmes suivant les ordinateurs.

Création des ensembles d'apprentissage et de test :

#2/3 :

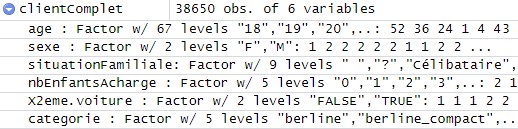
client\_EA <- clientComplet[1:25766,]

#1/3 :

client\_ET <- clientComplet[25767:38650,]

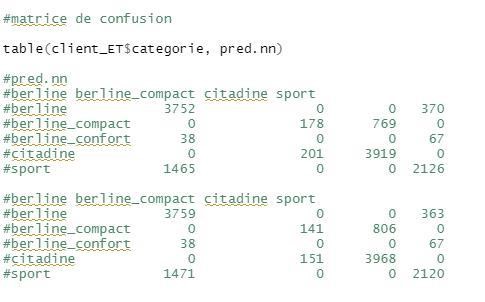
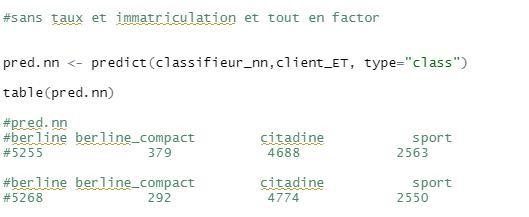
**Puis chaque classifieurs (sans taux) :**

# Sans taux



## NEURAL NETWORKS

classifieur\_nn <- nnet(categorie~., client\_EA, size=5)



Sur la deuxième matrice de confusion nous avons :

Taux d’erreur :

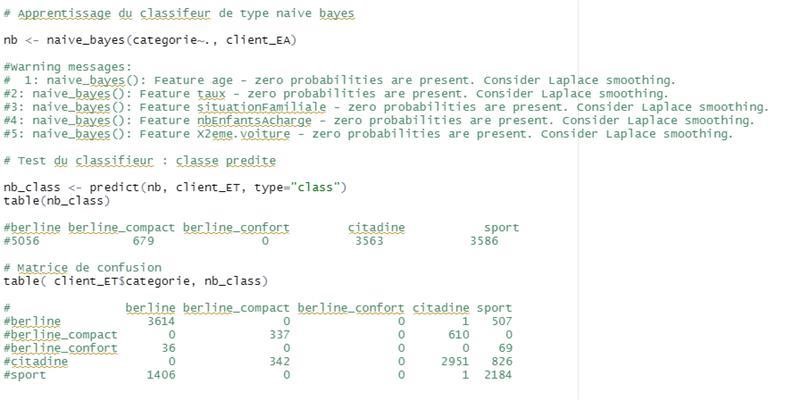
Ensemble de test : 12884

Prédictions correctes : 9988

Prédictions incorrectes : 2896

### Précision du classifieur : 9988/12884 = 77.52% Taux d’erreur : 2896/ 12884 = 22.48% nn\_auc : 0.8847

## NAIVE BAYES





Taux d’erreur :

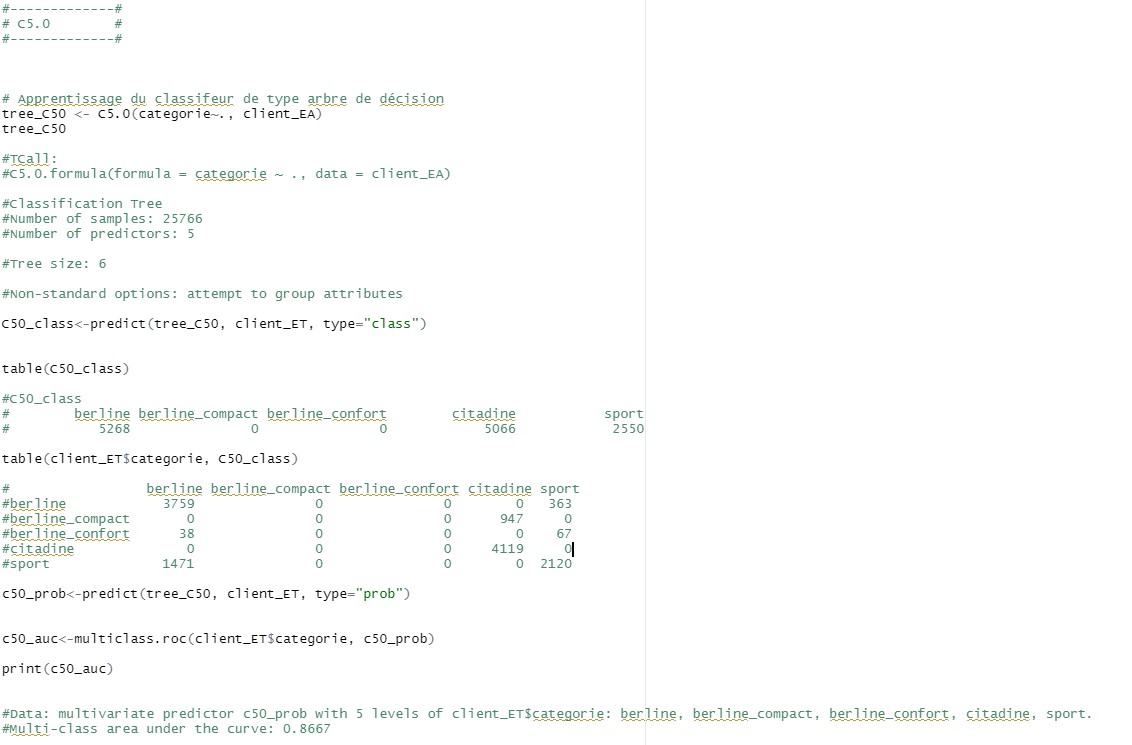
Ensemble de test : 12884

Prédictions correctes :9086

Prédictions incorrectes : 3798

### Précision du classifieur : 9086/12884 = 70.52% Taux d’erreur : 3798 / 12884 = 29.41% nn\_auc : 0.8621

## C5.0



Taux d’erreur :

Ensemble de test : 12884

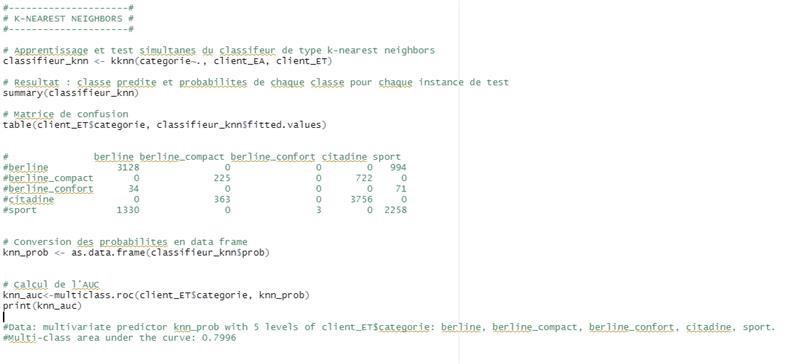
Prédictions correctes :9998

Prédictions incorrectes : 2886

**Précision du classifieur : 9998/12884 = 77.60% Taux d’erreur : 2886 / 12884 = 22.39%**

### Auc : 0.8667

## K-NEAREST NEIGHBORS



Taux d’erreur :

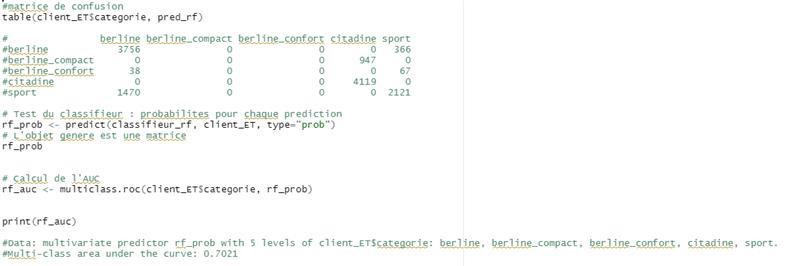
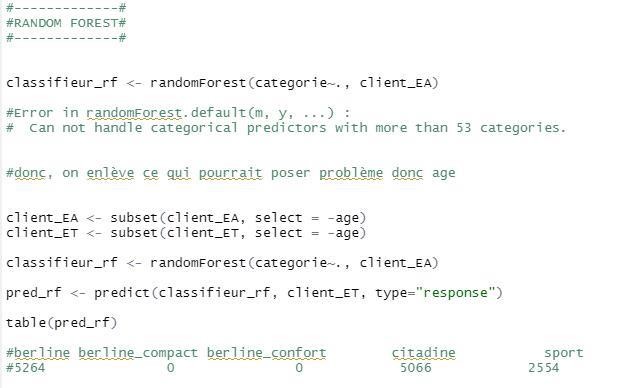
Ensemble de test : 12884

Prédictions correctes :9367

Prédictions incorrectes : 3517

### Précision du classifieur : 9367/12884 = 72.70% Taux d’erreur : 3517 / 12884 = 27.30% nn\_auc : 0.7996

## RANDOM FOREST



Taux d’erreur :

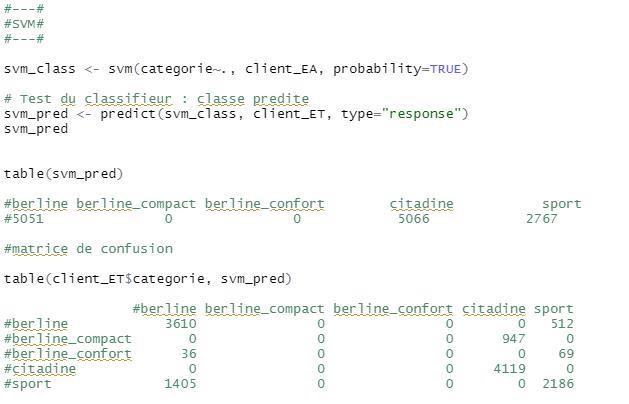
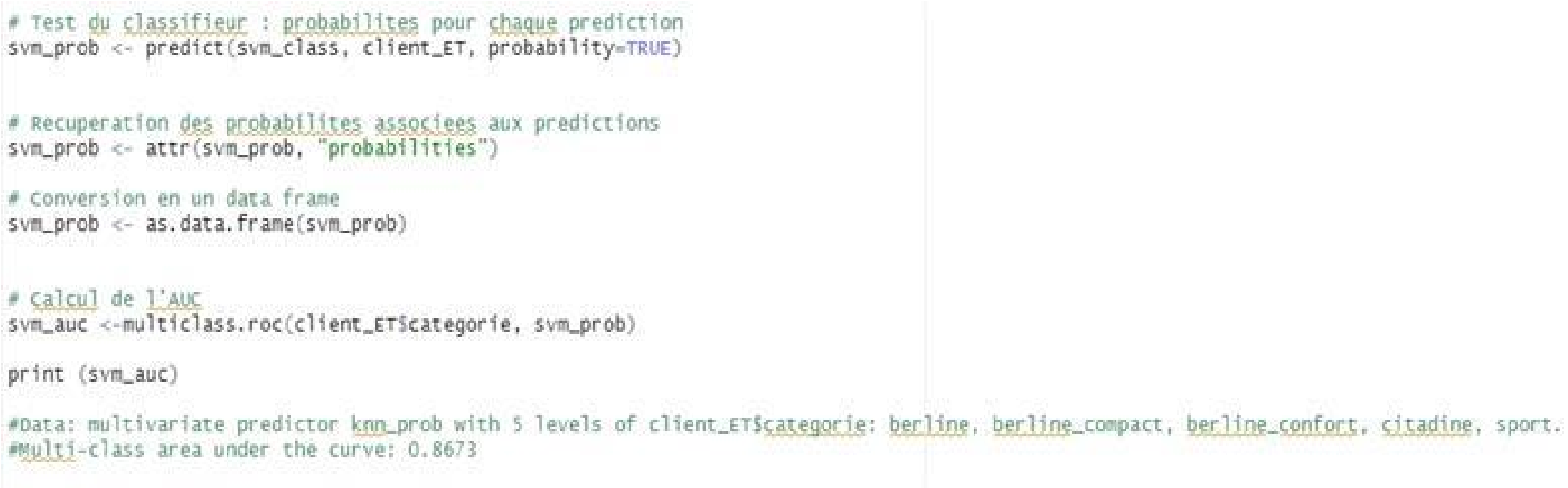
Ensemble de test : 12884

Prédictions correctes :9996

Prédictions incorrectes : 2888

### Précision du classifieur : 9996/12884 = 77.58% Taux d’erreur : 2888 / 12884 = 22.42% nn\_auc : 0.7021

## SVM



Taux d’erreur :

Ensemble de test : 12884

Prédictions correctes :9915

Prédictions incorrectes : 2969

### Précision du classifieur : 9915/12884 = 76.96% Taux d’erreur : 2969 / 12884 = 23.04% nn\_auc : 0.8673

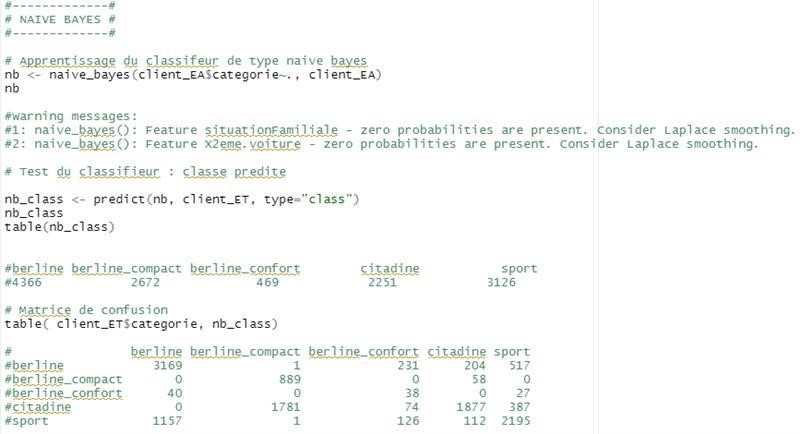
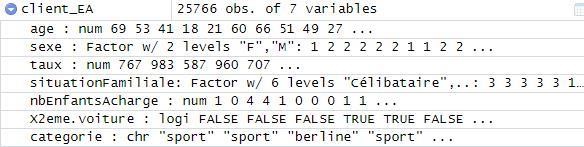
*Nous avons essayé également avec les classifieurs R-part et Tree mais cela ne marche pas puisqu’il y a trop de données pour ce type de classifieur.*

**Puis chaque classifieurs (avec taux) :**

# Avec taux

## NAIVE BAYES

Dans cette configuration, cela correspond à « avec taux » et sans mettre en « factor » les colonnes :





Taux d’erreur :

Ensemble de test : 12884

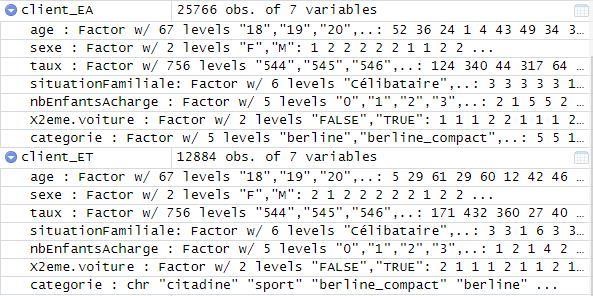
Prédictions correctes :8168

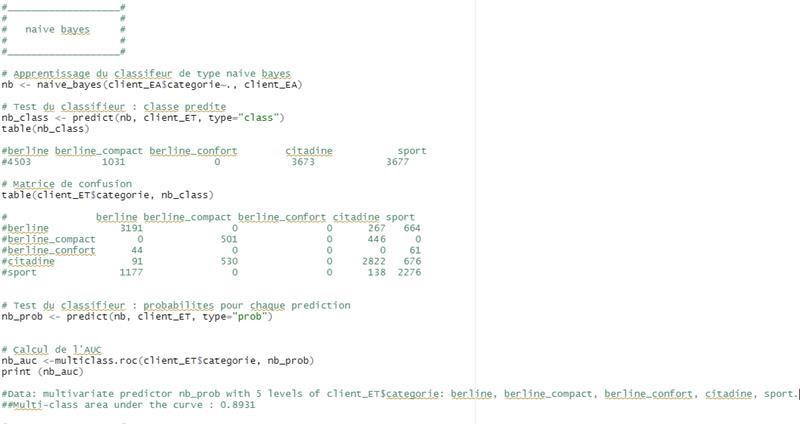
Prédictions incorrectes : 4716

**Précision du classifieur : 8168/12884 = 63.40%**

### Taux d’erreur : 4716 / 12884 = 36.60% Auc : 0.9017

Avec cette configuration, c’est-à-dire avec taux et tout en factor :





Taux d’erreur :

Ensemble de test : 12884

Prédictions correctes :8790

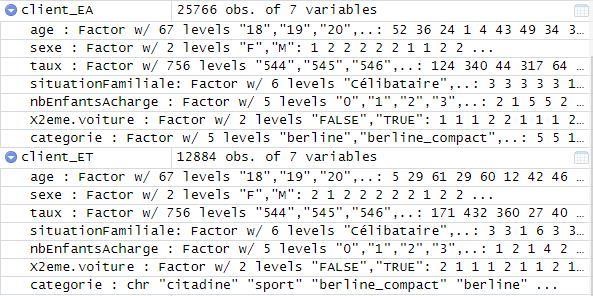
Prédictions incorrectes : 4094

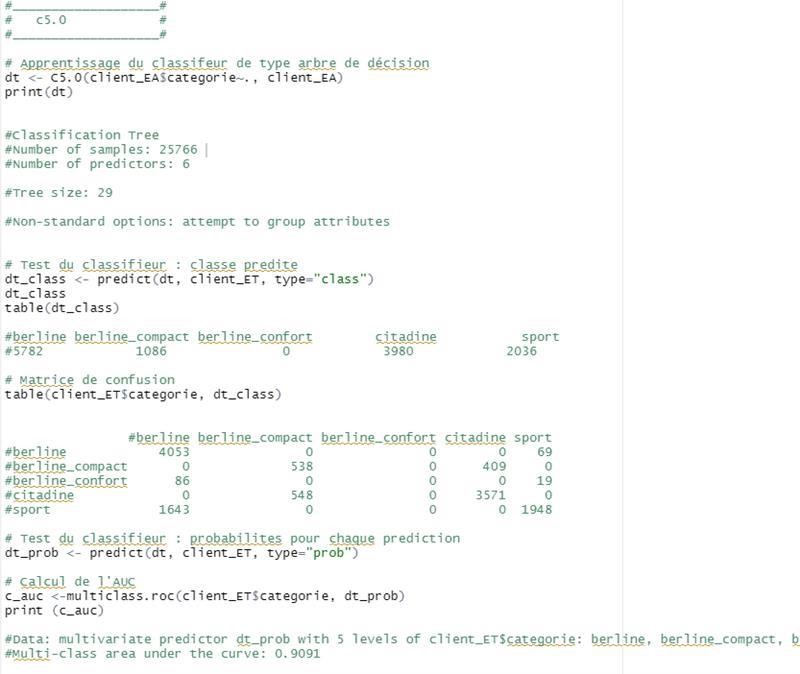
**Précision du classifieur : 8790/12884 = 68.22%**

### Taux d’erreur : 4094 / 12884 = 21.78% Auc : 0.8931

## C5.0 :

Avec cette configuration, c’est-à-dire avec taux et tout en factor :





Taux d’erreur :

Ensemble de test : 12884

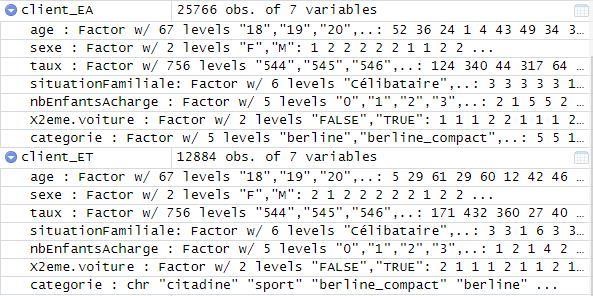
Prédictions correctes :10110

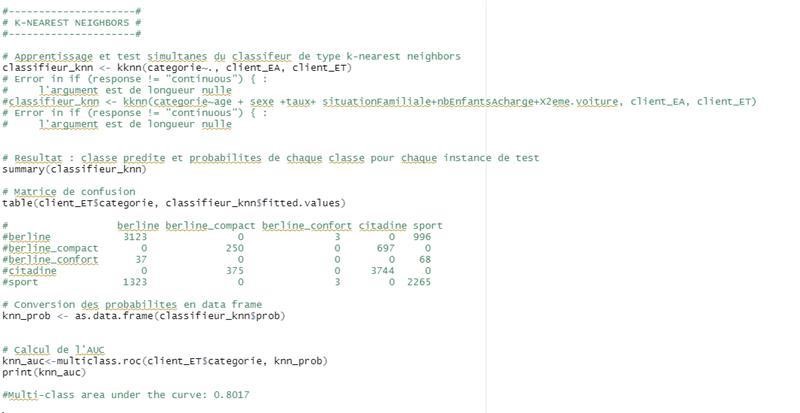
Prédictions incorrectes : 2774

### Précision du classifieur : 10110/12884 = 78.47% Taux d’erreur : 2774/ 12884 = 21.53% Auc : 0.9091

## K-NEAREST NEIGHBORS :

Avec cette configuration, c’est-à-dire avec taux et tout en factor :





Taux d’erreur :

Ensemble de test : 12884

Prédictions correctes :9382

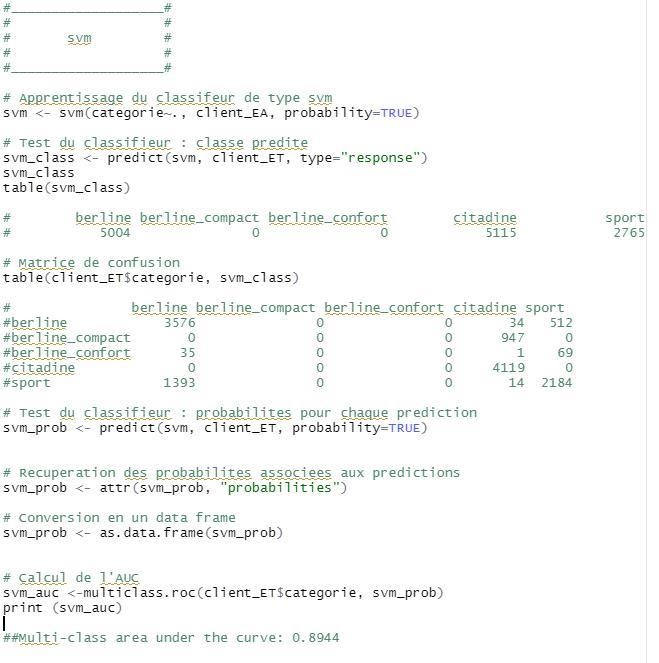
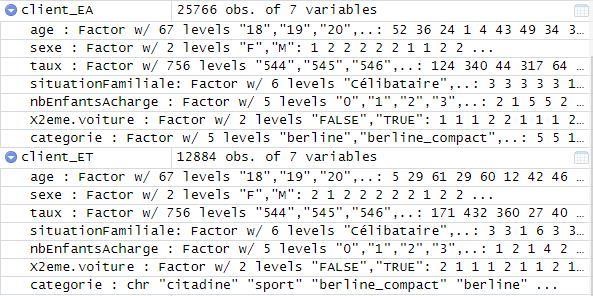
Prédictions incorrectes : 3502

**Précision du classifieur : 9382/12884 = 72.82% Taux d’erreur : 3502/ 12884 = 27.18%**

### Auc : 0.8017

## SVM :

Avec cette configuration, c’est-à-dire avec taux et tout en factor :



Taux d’erreur :

Ensemble de test : 12884

Prédictions correctes :9879

### Prédictions incorrectes : 3005 Précision du classifieur : 9879/12884 = 76.68% Taux d’erreur : 3005/ 12884 = 23.32% Auc : 0.8944

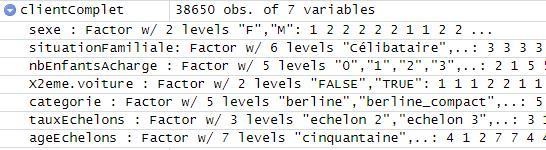
## RANDOM FOREST :

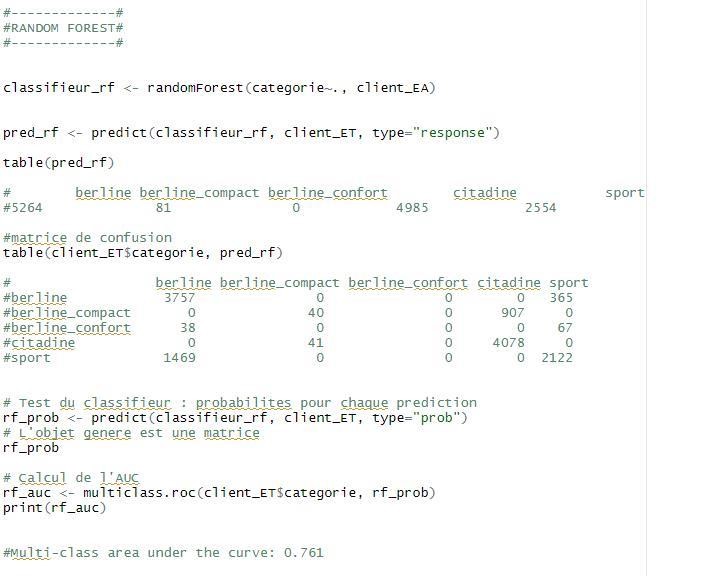
Aux vues du fait que random forest ne marche que pour 53 itérations max nous allons diviser taux en catégorie et âge en catégorie (nous avons essayé de ne pas mettre age en catégorie mais ca nous donnait un résultat moins précis) :

Pour mettre taux et âge en catégorie, nous avons fait comme ceci :



Nous avons donc cette configuration :





Taux d’erreur :

Ensemble de test : 12884

Prédictions correctes : 9997

Prédictions incorrectes : 2887

**Précision du classifieur : 9997/12884 = 77.59%**

### Taux d’erreur : 2887/ 12884 = 22.41% Auc : 0.761

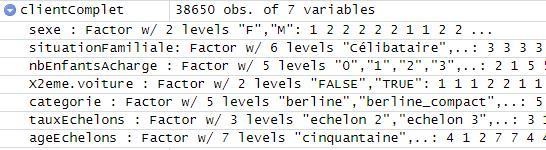
## NEURAL NETWORKS :

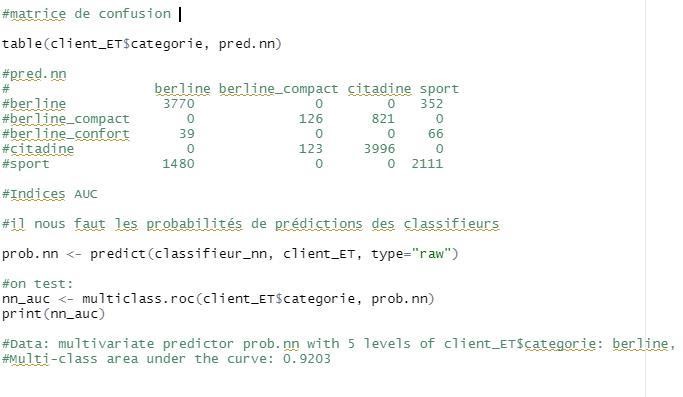
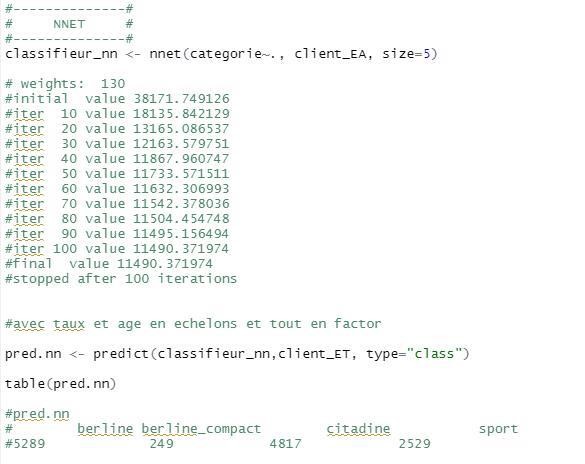
Aux vues du fait que neural networks ne marche que pour 53 itérations max nous allons diviser taux en catégorie et âge en catégorie :

Pour mettre taux et âge en catégorie, nous avons fait comme ceci :



Nous avons donc cette configuration :



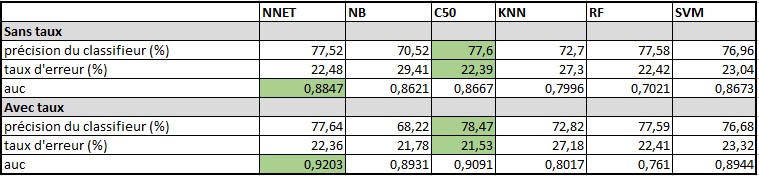


Taux d’erreur :

|  |  |
| --- | --- |
| Ensemble de test : 12884  Prédictions correctes : 10003  Prédictions incorrectes : 2881 | **Précision du classifieur : 10003/12884 =**  **77.64%**  **Taux d’erreur : 2881/ 12884 = 22.36% Auc :0.9203** |

# Classification – Prédiction de la Catégorie pour le fichier Marketing

Comparaison des résultats avec ou sans taux :



Petite conclusion :

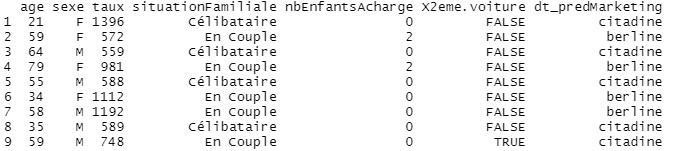
Nous avons décidé d’utiliser Auc et les taux de précision du classifieur pour choisir notre classifieur car nous ne pouvons pas tracer les courbes Roc pour les multi class. De plus nous remarquons que nous avons des meilleurs résultats avec taux.

Notre choix se porte donc sur 2 classifieurs : NNET et C50

On va donc appliquer la méthode de prédiction NNET et C50, étant donné que c’est celles qui présentent le meilleur AUC et la meilleure précision et donc la meilleure prédiction. L’ensemble d’apprentissage correspond au clientComplet et la prédiction se fera sur le dataframe marketing :

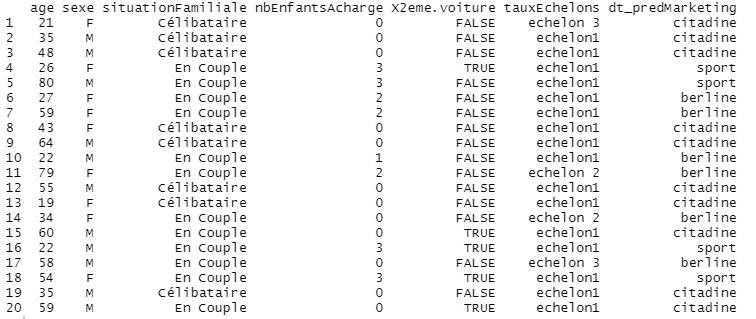
Application de nos classifieurs sur le fichier Marketing :

Résultat pour C50 :



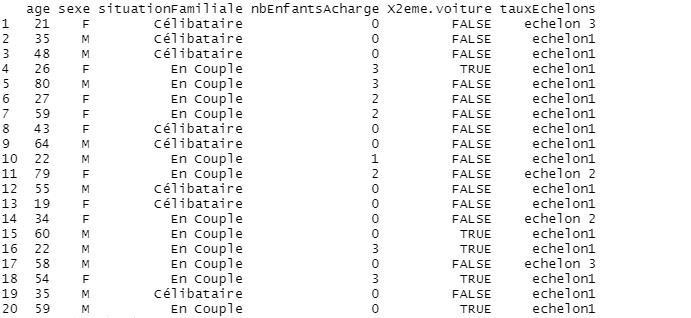
Nous remarquons que comme nous n’avions pas mis taux dans des catégories nous avons que 9 types de client sur 20 (dans le fichier marketing) qui peuvent correspondre puisque qu’il y a beaucoup de taux inférieur à 544 (et nous avions donc fait le tri dans client comme demandé dans le cahier des charges). Nous allons donc refaire une prédiction avec C50 et avec des catégories pour taux.

Résultat c50 avec taux en catégorie :



C’est bien mieux !

Résultat pour NNET :



Pour finir, nous avons enregistré nos résultats : trois fichiers, deux pour c50 (un avec taux en échelon et un sans ) et un pour nnet de prédictions dans le dossier DATA :







Conclusion : Nous remarquons qu’il n’y a que 3 catégories sur nos 5 prédites. Cela vient du fait que dans notre fichier client nous n’avions pas de personne avec des voitures 7 places donc la catégorie confort qui a 7 places est tout le temps vide mais nous restons persuadés qu’une famille nombreuse

(avec 4 enfants ou plus) choisira certainement une berline confort (7places). De plus la catégorie Berline compact n’apparait pas mais nous avons que 20 types de personnes dans le fichier Marketing. Nous aurions peut-être pu faire plus de catégories afin de lier au mieux chaque client pour chaque type de voiture mais nous sommes assez satisfaits des résultats.