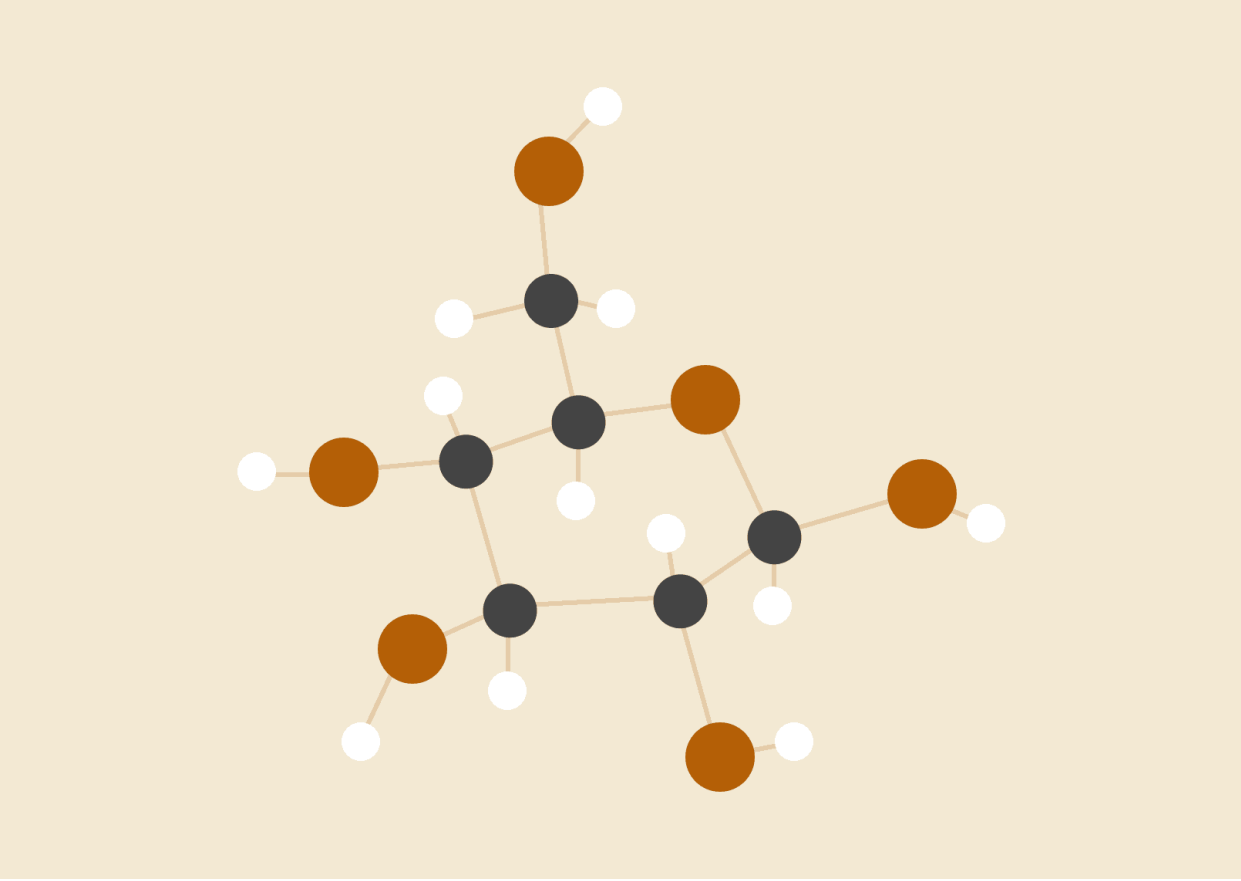
RAPPORT TP1

Cours d'apprentissage statistique de M. Dalalyan



Cordoba Muñoz Santiago
Ureña Stéphane
Watteau Rodolphe

20/10/2018
MS–DS

INTRODUCTION

Ce document a vocation à répondre aux différentes questions relatives au TP1.

Question 1

Cette commande permet de calculer l'erreur de classification.

La fonction « Apply » retourne un vecteur ou un tableau ou une liste de valeurs obtenus en appliquant une fonction aux marges d'un tableau ou d'une matrice. Voici ce que contiennent chacun des arguments :

1. une matrice (ici cvpred)
2. indique qu'on prend les colonnes
3. la fonction qui fait la somme des classes différentes de x : autrement dit la somme des mal classés pour chaque colonne

Par définition de ce qui précède, Apply retourne un vecteur contenant les valeurs mal classées par colonne.

Source : https://stat.ethz.ch/R-manual/R-devel/library/base/html/apply.html

Question 2

La classification KNN des proches voisins ne donne pas les mêmes résultats car elle est issue d'un point de départ dans les données fixé aléatoirement : chaque classification créée est donc unique.

A chaque itération nous pouvons potentiellement obtenir des résultats plus ou moins différents.

L'intérêt de faire une centaine de simulations serait d'obtenir de nombreux résultats plus ou moins similaires dont on pourrait faire une moyenne pour avoir une meilleure estimation du k optimal. Pour être plus précis, les 100 simulations renvoient 100 vecteurs. En prenant la moyenne de chaque coordonnées des 100 vecteurs, on peut déterminer le nombre de mal classés moyens pour chaque k entre 1 et 10. On peut ainsi choisir le cas ayant le plus faible nombre de mal classés moyen.

Question 3.1

Si l'on supprime l'argument "on=1", la courbe demandée ne s'affichera pas sur le plot.

```
#Question 1 : courbe mediane

MedPrice = function(p) apply(HLC(p), 1, median)
addMedPrice = newTA(FUN = MedPrice, col = 1, legend = "MedPrice")
addT.ind = newTA(FUN = T.ind, col = "red", legend = "tgtRet")
get.current.chob<-function(){quantmod:::get.current.chob()}
candleChart(last(GSPC, "3 months"), theme = "white", TA = "addMedPrice(on=1)")
candleChart(last(GSPC, "3 months"), theme = "white", TA = "addMedPrice(on=1)")
candleChart(last(GSPC, "3 months"), theme = "white", TA = "addT.ind();addMedPrice(on=1)")
```

Illustration 1: Code qui permet d'ajouter aux chandeliers japonais la courbe des valeurs médianesde (Ci, , Hi, Li)

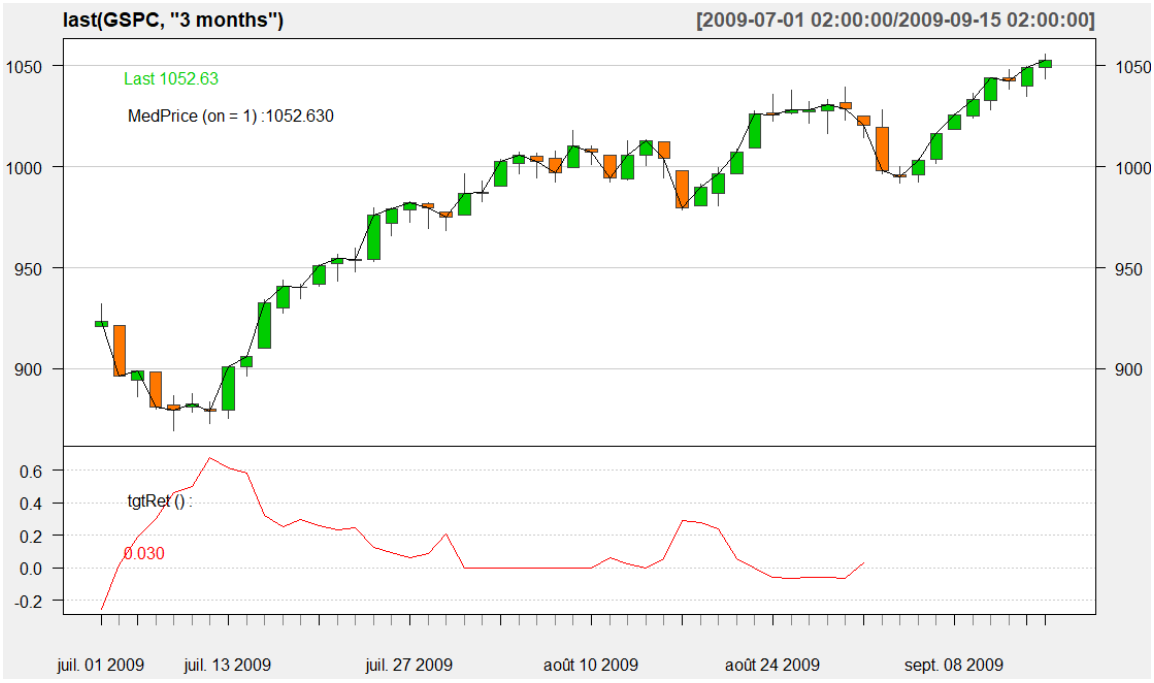


Illustration 2: Courbe des valeurs des médianes

Question 3.2

1. L'argument training.per correspont au format des dates qui sont étudiées dans la série, ce format est une suite de caractères mise sous forme de vecteur. Ce vecteur de caractères représente des dates au format ISO 8601 "CCYYY-MM-DD" ou "CCYYY-MM-DDD HH:MM:SS" de longueur 2 (il doit correspondre au format de date indélébile des données des séries chronologiques utilisées dans la modélisation).

2. « Importance » est un argument : il s'agit de l'importance des variables dans le modèle. Dans notre cs présent, le modèle de Random Forest renvoie également un objet importance : il s'agit là de la diminution moyenne de l'impureté apportée par chaque variable. Elle est calculée par l'index de Gini : la diminution pour chaque noeud est cumulée, puis une moyenne sur l'ensemble des arbres est effectuée.
   L'instruction True ou False permet de décider si on veut que le modèle renvoie cette information ou non.

Source : https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/importance

Question 3.3

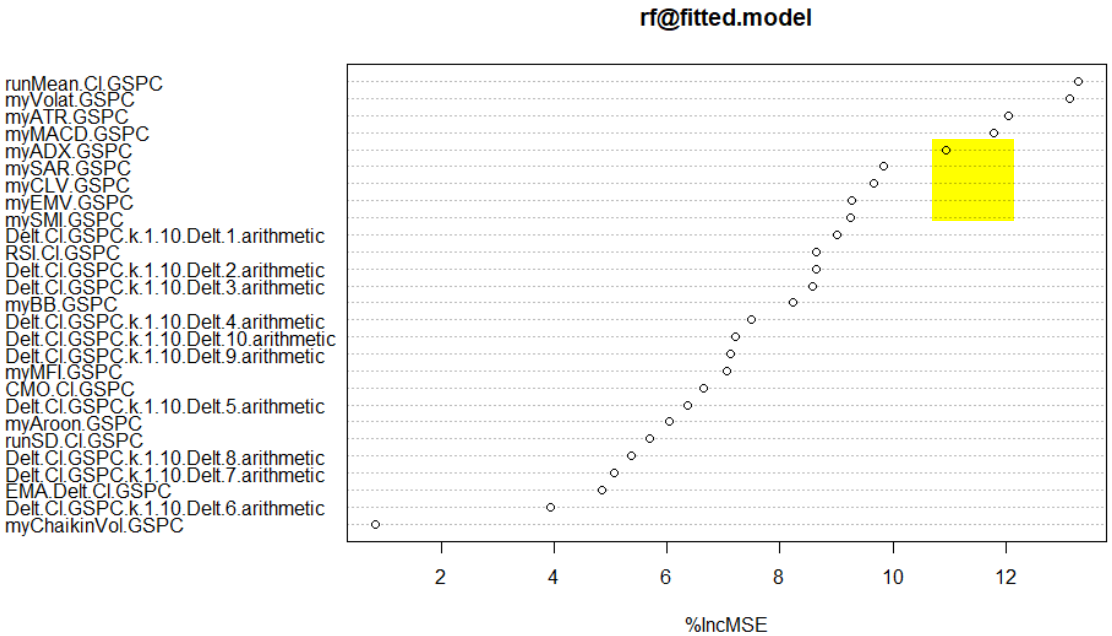


Illustration 3: Pourcentage d'augmentation de l'erreur quadratique due à la suppression d'une variable explicative

Le graphique ci-dessus traduit le pourcentage d'augmentation de l'erreur quadratique due à la suppression d'une variable explicative. Ainsi, on voit que les variables sont ordonnées par ordre d'importance: de la plus importante à la moins importante (ordre décroissant). En effet, la première variable "runMean.CI.GSPC" génère une hausse de 13% de l'erreur quadratique lorsqu'elle est retirée du modèle. Il s'agit de la hausse la plus significative. Vient ensuite "myVolat.GSPC", avec une valeur légèrement inférieure.

Le tableau ci-dessous récapitule les 8 variables les plus importantes pour le modèle au regard du pourcentage d'augmentation de l'erreur quadratique induit par leur retrait du modèle:

Classement (importance)	Nom de la variable	Augmentation de MSE induit par leur retrait (en %)(valeurs approximatives)
1	runMean.Cl.GSPC	13
2	myVolat.GSPC	12,8
3	myATR.GSPC	12
4	myMACD.GSPC	11,8
5	myADX	11
6	mySAR	9,9
7	clv	9,8
8	emv	9,2

Illustration 4: les 8 variables ayant le plus de pertinence dans notre modèle (au regard du critère de l'illustration 1)

Question 3.4

Utilisation de la fonction specifyModel pour définir le nouveau modèle data.model ayant pour variable à expliquer T.ind(GSPC) et comme variables explicatives les 8 variables les plus pertinentes trouvées dans la question précédente. Voici la commande (on retire toutes les variables et on ne garde que les 8 pertinentes) :

```
data.model = specifyModel(T.ind(GSPC) ~ myATR(GSPC) +
                          myADX(GSPC) + myCLV(GSPC) + myEMV(GSPC) + myVolat(GSPC) +
                          myMACD(GSPC) +  mySAR(GSPC) + runMean(Cl(GSPC)))
```

Question 3.5

La fonction na.omit sert à supprimer les lignes avec des valeurs manquantes sur les colonnes spécifiées.

source : https://www.youtube.com/watch?v=-rEPm3yBUBY

https://www.rdocumentation.org/packages/data.table/versions/1.11.8/topics/na.omit.data.table

Cette commande est essentielle pour l'échantillon de test car on peut faire échouer l'ensemble des prédictions si les valeurs manquantes viennent à être trop nombreuses.

Par définition, l'échantillon test sert à mesurer la précision du modèle. Afin de tester la précision maximale de ce dernier, il peut être intéressant de supprimer les lignes des données manquantes, pouvant biaiser l'évaluation de la précision du modèle, en particulier si les données sont de type « Missing at Random (MAR) » ou de type « Missing Completely at Random (MCAR) ».

source : https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4

Question 4

Nous avons relancé l'algorithme KNN la variable signal sur l'échantillon de test Tdata.eval, en utilisant Tdata.train comme échantillon d'entraînement. Voici le code :

```
pred = knn(Tdata.train[,-1], Tdata.eval[,-1], Tdata.train[,1], k = 3)

essa = table(pred,Tdata.eval[,1])
erro = mean(pred != Tdata.eval[,1])
perfo = 1-erro
erro
perfo

> erro
[1] 0.5547325
> perfo
[1] 0.4452675
```

L'erreur est de 55%, ce qui est très élevé.

Question 5

Nous relançons la prédiction avec un arbre (rpart) cette fois-ci. Voici le code :

```
library(rpart)
signal<-Tdata.train[,1]
Q5 <- rpart(signal ~., data = Tdata.train[,-1], control = rpart.control(cp = 0.0000001,maxdepth = 3))
plot(Q5)
text(Q5, use.n=TRUE,col="blue")
prettyTree(Q5,col="navy",bg="lemonchiffon")
pred2<-predict(Q5, Tdata.eval[,-1], type="class")
s= table(pred, Tdata.eval[,1])

erro2 = mean(pred2 != Tdata.eval[,1])
perfo2 = 1- erro2
erro2
perfo2

> erro2
[1] 0.3057613
> perfo2
[1] 0.6942387
```

L'erreur est de 30% dorénavant, ce qui demeure élevé mais bien meilleur que l'erreur de classification obtenue avec le KNN. On obtient l'arbre de décision ci-dessous :

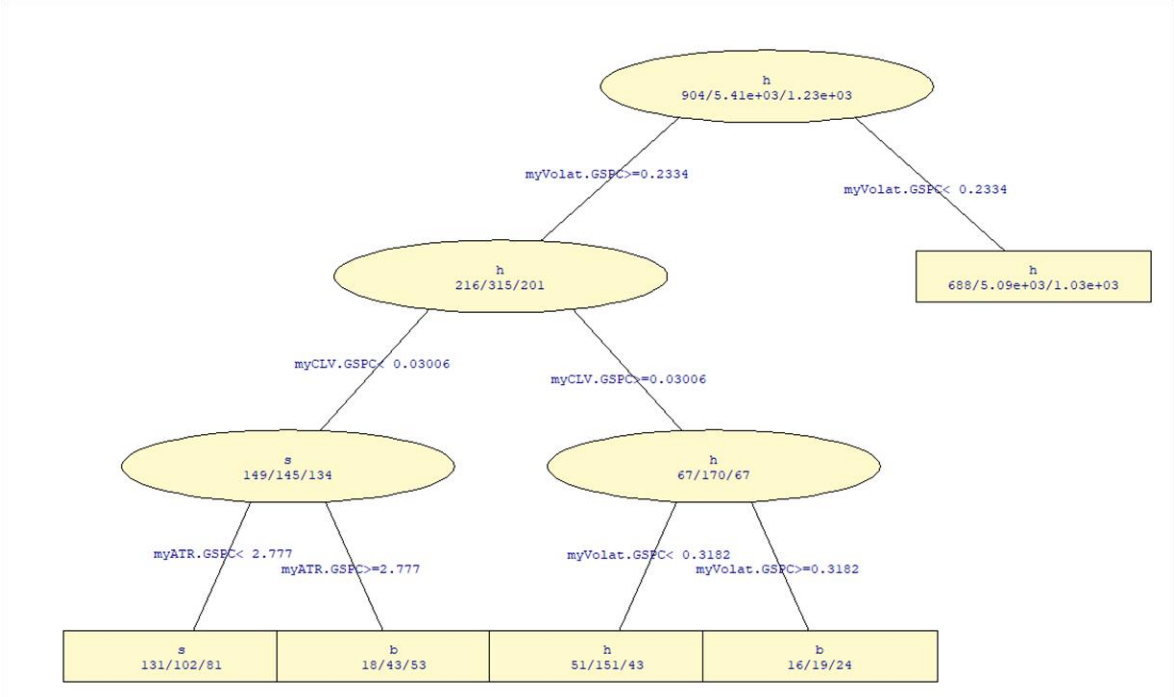


Illustration 5: les 8 variables ayant le plus de pertinence dans notre modèle (au regard du critère de l'illustration 1)

Questions 6 et 7

Nous y répondons à travers ce rapport que nous espérons à votre convenance. N'hésitez pas à nous apporter vos conseils sur la forme pour les suivants (ainsi que sur le fond bien évidemment si des choses sont à améliorer).

RÉFÉRENCES

Question 1: https://stat.ethz.ch/R-manual/R-devel/library/base/html/apply.html

Question 3.2 : https://www.rdocumentation.org/packages/randomForest/versions/4.6-14/topics/importance

Question 3.4:
https://www.youtube.com/watch?v=-rEPm3yBUBY
https://www.rdocumentation.org/packages/data.table/versions/1.11.8/topics/na.omit.data.table

Question 3.5 :
https://www.youtube.com/watch?v=-rEPm3yBUBY
https://www.rdocumentation.org/packages/data.table/versions/1.11.8/topics/na.omit.data.table
*https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4