LOPES D OLIVEIRA Ophélie 18.12.2018

TD Machine Learning

Pour pouvoir faire ce TD, j’ai repris les formules faite en cours pour la méthode GLM.

J’ai choisi de comparer la méthode GLM avec la méthode RF.

# Méthode Generalized linear model (GLM)

## Présentation

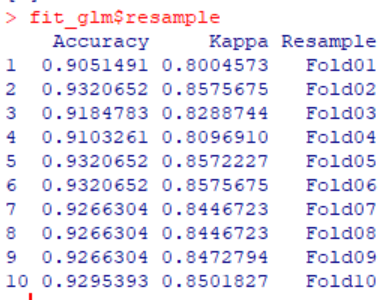
Le modèle linéaire généralisé qui s’inspire de la régression linéaire pour réaliser un modèle linéaire.

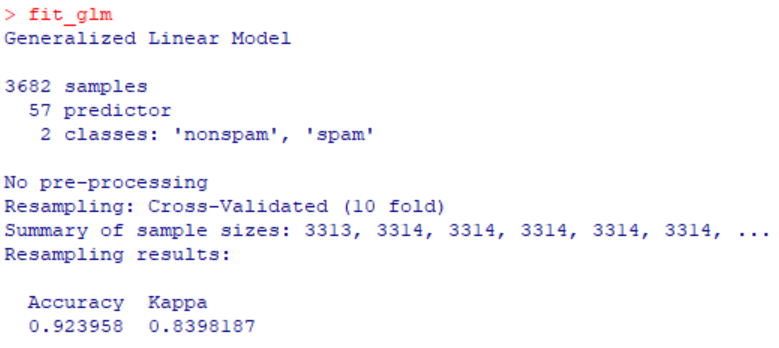
La méthode GLM a pour avantage d’être facile à interpréter et à implémenter, cependant, elle a pour inconvénient de “sous-apprendre” les données, de ne pas capturer la complexité dans les données et qu’il s’agit d’une méthode paramétrique.

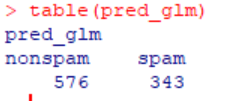
## Réalisation

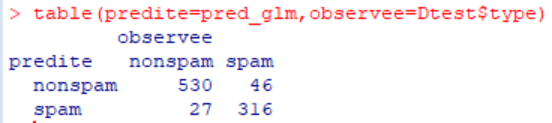
Pour réaliser cette partie, j’ai repris les éléments vu en cours, donc un paramétrage avec 10 boites. (Voir annexe - Source R)

## Résultat









On voit donc qu’il y a eu:

* 46 non spam prédit qui était en réalité des spam soit 5%
* 27 spam prédit qui n’était pas des spams soit 3%
* 530 prédiction de non spam correctes soit 58%
* 316 spam correctes soit 34%



On observe donc un taux d’erreur de 7,9%

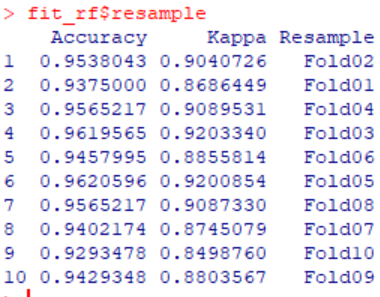
# Méthode Random Forest (RF)

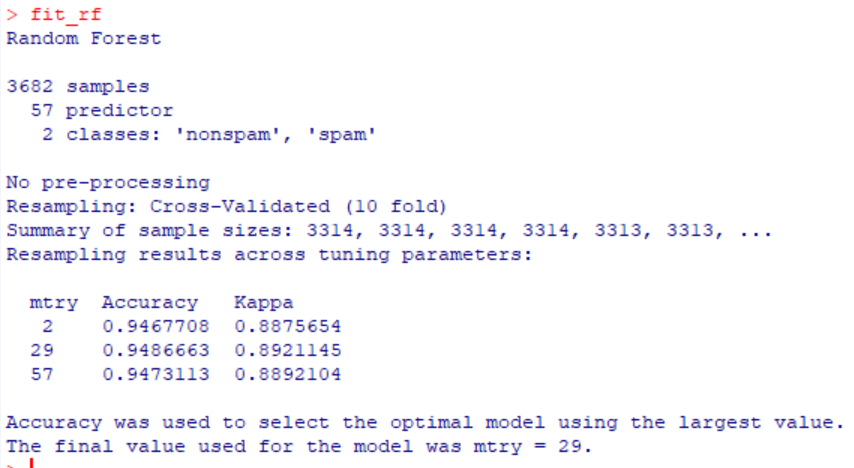
## Présentation

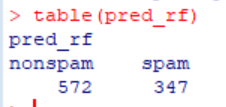
La méthode Random Forest ou Forêt d’Arbres décisionnels est une technique d’apprentissage automatique. Cet algorithme combine les concepts de sous-espaces aléatoire et de Bootstrap (technique de rééchantillonnage).

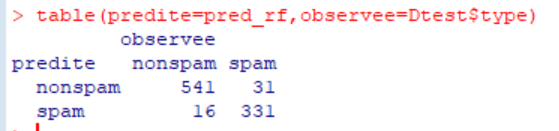
La méthode Random Forest a pour avantage de ne nécessiter quasiment aucun réglage, de contrôler la variance grâce à au ré-échantillonnage aléatoire et d’être une méthode non paramétrique. Cependant, cette méthode est modérément rapide à entraîner, mais rapide pour prédire et qu’il est moins interprétables que les arbres de décision

## Réalisation









On voit qu’il y a:

* 541 spam correct soit 59%
* 31 non spam qui était en réalité des spams soit 3%
* 16 spam qui était des non spam soit 2%
* 331 spam correct soit 36%



Il y a un taux d’erreur de 5,1%

# Comparaison

Au niveau de la prédiction:

|  |  |  |
| --- | --- | --- |
|  | Spam observé | Non Spam observé |
| Spam prédit | 37%/36%/ 34% | 0%/2%/ 3% |
| Non spam prédit | 0%/3%/ 5% | 63%/59%/ 58% |

en noir : réalité

en rouge: méthode GLM

en vert : méthode RF

On remarque que la méthode RF possède des taux plus proche de la réalité que la méthode GLM, cela peut notamment se voir par le taux d’erreur qui est de 5,1% pour la méthode RF et 7,9 pour la méthode GLM.

# Annexe : Fichier R

##CHARGEMENT DES DONNEES

#On charge les librairies

library(kernlab)

library(caret)

library(tidyverse)

#On charge les donnees

data(spam)

View(spam)

#On génère une graine

set.seed(100)

#On enregistre les donnée test

ind\_train <- createDataPartition(spam$type,p=0.8,list=F)

length(ind\_train)

head(ind\_train)

##CREATION DES DONNEES D ENTRAINEMENT

#on assigne la matrice spam qui contient toutes les lignes qui sont dans ind-train et toutes les colonnes de spam

Dtrain<-spam[ind\_train,]; dim(Dtrain)

#Creer la table des indices de test à partir de toutes les données qui sont pas dans Dtrain

Dtest<-spam[-ind\_train,]; dim(Dtest)

#paramètre d'apprentissage pour la validation croisée

param\_train<-trainControl(method="cv",number=10)

##METHODE GLM

##Validation croisee

#on lance la fonction train avec la méthode "glm"

fit\_glm<-train(type~.,data=Dtrain,method="glm",trControl=param\_train)

#détail des resultats, le taux pour chaque fold, avec le champ resample

fit\_glm$resample

#donne le taux moyen de tous les taux d'erreur

fit\_glm

#On cree la variable qui contiendra la prediction

pred\_glm<-predict(fit\_glm,newdata=Dtest)

head(pred\_glm)

table(pred\_glm)

#Compter le nb de fois où on se trompe quand on utilise les données de test avec mon algo entrainé

#affiche la matric de confusion

table(predite=pred\_glm,observee=Dtest$type)

#la moyenne des erreurs sur la variable type

mean(pred\_glm!=Dtest$type)

head(Dtest$type) --> affiche

##METHODE RF

##Validation croisee

#on lance la fonction train avec la méthode "rf"

fit\_rf<-train(type~.,data=Dtrain,method="rf",trControl=param\_train)

#Affichage des taux

fit\_rf$resample

#donne le taux moyen de tous les taux d'erreur

fit\_rf

#On cree la variable qui contiendra la prediction

pred\_rf<-predict(fit\_rf,newdata=Dtest)

head(pred\_rf)

table(pred\_rf)

#Compter le nb de fois où on se trompe quand on utilise les données de test avec mon algo entrainé

#affiche la matric de confusion

table(predite=pred\_rf,observee=Dtest$type)

#la moyenne des erreurs sur la variable type

mean(pred\_rf!=Dtest$type)

#Afficher les 6 premieres lignes

#on observe qu'il y a 1 erreur

head(Dtest$type)

head(pred\_rf)

# SOURCE

(Consulté le 13/12/2018) <https://www.institutdesactuaires.com/global/gene/link.php?doc_id=9419&fg=1>

(Consulté le 13/12/2018)

<https://fr.wikipedia.org/wiki/Mod%C3%A8le_lin%C3%A9aire_g%C3%A9n%C3%A9ralis%C3%A9>

(Consulté le 13/12/2018)

<https://fr.wikipedia.org/wiki/For%C3%AAt_d%27arbres_d%C3%A9cisionnels>