

Le travail de TEDDY LEANDRE est disponible ici

Théo Marié

20/12/2020

SYNTHESE DU TRAVAIL EN QUESTION

Le package Caret est important car il nous introduit à l'analyse prédictive, très intéressante en programmation. La notion de seed doit être expliquée, ainsi que l'essence même de ce qu'est l'analyse prédictive, et quelques étapes importantes, comme l'entraînement et le test. Malheureusement, le travail ici n'est pas complet, et manque d'efficacité. Il ne contient pas non plus de fonctionnalité externe qui pourraient habiller l'algorithme, pour le rendre plus efficace.

Il manque aussi beaucoup de chose : il faudrait lancer l'entraînement, et pour cela utiliser **train** expliquer comment le modèle d'entraînement est lancé, avec quoi (random forest, forest ?). Il faudrait également appliquer le jeux de test à l'entraînement, étudier ce que nous a donné l'algorithme, le comparer à otre test, puis définir quand la prédiction est bonne et enfin donner le pourcentage de réussite.

On aurait aussi pu voir des petites fonctionnalités, comme une visualisation graphique des prédictions e l'agorythme, le fait de savoir combien de temps met l'algorithme a fonctionner, ou tout simplement le renforcement de l'aléatoire.

EXTRAIT COMMENTE DES PARTIES DU CODE

```
library(caret)
# Création des ensembles de données d'entraînement et de test
set.seed(100)
# Etablir un nouvel index pour les chaque ligne 'species'
iris_index <- createDataPartition(iris$Species, p = .8,
                                  list = FALSE,
                                  times = 1)
```

Ici, le code est bon : Il va premièrement appeler le package CARET dans sa librairie, puis grâce à "createDataPartition" va créer un tableau représentant les données d'IRIS, en séparant en 2 parties (80-20).

Cependant, il n'explique pas ce qu'est SEED. Il faut rappeler qu'il est important de générer de l'aléat pour ne pas fausser notre prédiction, et donc donner une "graine à notre algorithme en tant que référence de départ .

```
# Creation d'un dataset d'entraînement
Train_iris <- iris[iris_index,]
# Création d'un dataset sur lequel se feront les tests
Test_iris <- iris[-iris_index,]
```

Ici, il est faux d'affirmer que le dataset Test a été créé pour y tester directement l'algorithme. Il faut émettre une nuance : Le Test-iris va nous servir de modèle, auquel on va étudier les réponses obtenues avec le test qui s'effectuera distinctement. Test_iris servira donc à vérifier si notre prédiction est juste, en donnant l'accuracy, c'est à dire exactitude de notre prédiction.

EVALUATION DU TRAVAIL EN QUESTION

Critère 1 : Visuellement appréciable sur pdf 2/4 Pas très visible, à ne pas négliger

Critère 2 : idées pour faire le code 1/4 Pas du tout assez

Critère 3 : Fonctionnalité du code 2/4 Marche mais n'est pas complet

Critère 4 : lisibilité du code 3/4 lisible

Critère 5 : explications données 1/4 Les explications données sont partielles et desfois fausses.

CONCLUSION

Il y a de gros manquement sur ce code. Malheureusement, alors que le début sonnait bien, il manque toute la partie intéressante du code, à savoir la probabilité de réussite de l'algorithme que l'on a entraîné.