

# Le travail de THEO MARIE 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 est expliquée, ainsi que l'essence même de ce qu'est l'analyse prédictive. Les étapes importantes sont clairement énoncées comme l'entraînement et le test. On aurait aussi pu voir des petites fonctionnalités, comme une visualisation graphique des prédictions et l'algorithme, le fait de savoir combien de temps met l'algorithme à fonctionner.

## EXTRAIT COMMENTÉ DES PARTIES DU CODE

```
library(caret) ## ON INVOQUE CARET
set.seed(333) ## ON PLANTE UNE GRAINE

trainIndex <- createDataPartition(iris$Species, p = 0.8,
                                   list = FALSE,
                                   times = 1)

irisEntrainement <- iris[ trainIndex,] ##Création du dataset d'entraînement
irisTest <- iris[-trainIndex,] ##Création du dataset de Test
```

Extrait bien commenté, qui nous renseigne exactement sur ce qu'on est entrain de faire.

```
Theo <- train(Species ~ .,
              data = irisEntrainement,
              method = "rf",) ##il faut ici installer le package randomforest
```

Mise en place du modèle Théo, qui va utiliser la méthode randomForest pour s'entraîner sur irisentraine-ment.

```
prediction_du_test_iris <- predict(Theo, irisTest) ## il faut ici installer le package e1071
tableau_de_prediction <- data.frame(irisTest$Species, prediction_du_test_iris )

reussite <- tableau_de_prediction[,1] == tableau_de_prediction[,2]
nombre_de_reussites <- sum(reussite)
nombre_de_ligne <- nrow(tableau_de_prediction)
exactitude_de_la_prediction <- nombre_de_reussites/nombre_de_ligne * 100
```

Nous sommes ici aux dernières lignes de code, qui nous renseigne sur notre principal but, c'est à dire l'accuracy de notre algorithme. Les lignes de code sont claires, efficaces.

## EVALUATION DU TRAVAIL EN QUESTION

Critère 1 : Visuellement appréciable sur pdf 3/4 Utilisation de couleurs, et de démarquations, espacés

Critère 2 : idées pour faire le code 2/4 Pas très novateur car vu en cours, mais suffisant pour expliquer

Critère 3 : Fonctionnalité du code 3/4 Assez complètes

Critère 4 : lisibilité du code 4/4 Code très lisible

Critère 5 : explications données /4 Explications nombreuses, étoffées, pour être pédagogue

## **CONCLUSION**

Bon code, très bonnes explications, mais on aurait pu faire mieux en rajoutant quelques fonctionnalités, comme avec le temps que l'algorithme a pris pour effectuer les calculs (`t_after - t_before`, en utilisant `sys.time()`), ou encore un graphique, avec `ggplot2`.