kadaoui_alexandre_CrossValidation

Alexandre

21/12/2020

R Markdown

Dans le cadre de notre partiel, nous devons réaliser un total de 12 travaux retracant notre parcours et notre travail durant les 30 heures de cours.
Le travail à faire est le suivant :
- Une entête comportant un titre, un lien Github avec le ou les noms des auteurs.
- Une synthese de ce travail
- Un extrait commenté avec des parties de codes clé avec explication et commentaire.
- Une évalutation du travail avec nos 5 criteres.
- Une conclusion du travail

Definition des 5 critères de notations :

1) Effort de présentation :
2) Le knit est réalisable et bien présenté.
3) Explications simples et efficaces.
4) Le Code reproductible à d'autres DataFrame avec facilité.
5) Description des fonctions utilsés et du raisonnement.

La Cross Validation

Travail réalisé par "Marko ARSIC / Rindra LUTZ" le 15/11/2020.

https://github.com/ARSICMrk/ARSIC_PSBx/blob/main/R_Travail_Sup
(https://github.com/ARSICMrk/ARSIC_PSBx/blob/main/R_Travail_Sup)

Synthese :

La Cross Validation correspond à une méthode de vérification du set de donnée utilisé lors de l'entrainement d'un modèle afin d'augmenter la fiabilité du modèle.
A cette fin on isole une partie du pool de données utilisé pour l'apprentissage afin de l'utiliser en tant qu'échantillon de test.
On pourra répéter ce processus un nombre K de fois (ou "K folds") en utilisant à chaque fois une partie différente du pool de données qui servira de test pour le modèle.
Il est important de noter qu'un "sur-apprentissage" du modèle sur un même pool de données sera dans une majeure partie des cas contre-productive. En effet, un modèle surentrainé sur un pool en particulier sera trop influencé par les bruits et/ou cas aberrants de ce pool et sera donc moins à même d'établir des prédictions de manière pertinente à partir de pools de données différents.

Extrait commenté du code :

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
# Téléchargement des données t
data("swiss")

# Inspecter les données
sample_n(swiss, 3)
```

```
##           Fertility Agriculture Examination Education Catholic Infant.Mortality
## Sion           79.3        63.1          13        13    96.83             18.1
## Yverdon        65.4        49.5          15         8     6.10             22.5
## Entremont      69.3        84.9           7         6    99.68             19.8
```

```
# Mise en place de la cross validation "cv" pour k = 10 folds, on répète ici la cross validat
ion 3 fois afin de réduire le bruit et donc la MAE et RMSE
set.seed(123)
train.control <- trainControl(method = "cv", number = 10
                              , repeats = 3)
```

```
## Warning: `repeats` has no meaning for this resampling method.
```

```
# Entraîner le modèle poour une modélisation linéaire "lm" à partir du pool de données "swis
s" en appliquant la cross validation établie plus haut
model <- train(Fertility ~., data = swiss, method = "lm",
               trControl = train.control)

# Résultats résumés, R2 correspond au coef de corrélation qui doit tendre au maximum vers 1,
 MAE (l'erreur moyenne absolue) et RMSE (l'erreur quadratique moyenne) doivent tendre au maxi
mum vers 0
print(model)
```

```
## Linear Regression
##
## 47 samples
##  5 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42, 42, 42, 42, 42, 44, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   7.424916  0.6922072  6.31218
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Evaluation du travail :

1) Effort de présentation : Le RMD ainsi que le PDF sont bien structurés et bien organiséspour être lisible à la perfection. De plus la vidéo ajoute un complément d'information intéressant et dynamique.
2) Le knit est réalisable et bien présenté : Knit réalisable sans soucis et bien mis en page
3) Explications simples et efficaces : Les explications sont claires et directes sans détourds superflus.
4) Le Code reproductible à d'autres DataFrame avec facilité : Le code est simplifié au maximum afin de ne conserver que les fonctions essentielles qui seront facilement réplicables dans n'importe quelle situation similaire
5) Description des fonctions utilsés et du raisonnement : Les fonctions clefs utilisées ici sont les fonctions **traincontrol()** ainsi que **train()** et la fonctionnalité principale est l'application d'une cross validation à K= 10 folds

Conclusion :

En conclusion, ce tutoriel complet est direct permet de s'approprier de manière simple et efficace les bases de l'entrainement d'un modèle à partir d'un pool de données et le perfectionnement de ses capacités de prédicition grâce à la Cross Validation. Le support vidéo ainsi que les schéma ajoutés au github sont des bonus d'une grande valeur afin de faciliter la compréhension et apporter plus de dynamisme et d'intéractivité au tutoriel.