

Mon analyse R

Jingwen SU

12/20/2020

Introduction

À la fin du cours, tous les étudiants ont téléchargé les documents pertinents sur Github et partagé les résultats avec nous. Afin de profiter pleinement de ces packages R sur les mathématiques, je vais lire, analyser et évaluer les articles de mes collègues pour améliorer mes lacunes.

Ceci est une introduction à l'article lu dans cet article. Si vous souhaitez en savoir plus, vous pouvez rechercher des références.

- **Title:** Caret
- **Auteurs:** Jingwen SU
- **Lien sur Github:** Jingwen SU

Synthèse du travail en question

Ceci est un article d'introduction sur Caret. L'auteur a donné une brève introduction sur le contenu de l'article au début. Dites au lecteur que dans cet article, elle utilisera le package caret pour expliquer et pratiquer le prétraitement et la segmentation des données. Et d'introduire l'ensemble de données qui seront utilisées ci-dessous.

Dans la deuxième partie, l'auteur a présenté le code pour installer le package Caret. Après cela, l'auteur a commencé à démontrer le traitement des données.

Pour Prétraitement des données, il a démontré la gestion des valeurs manquantes et les types de données de conversion. Y compris la suppression des valeurs manquantes, l'introduction des fonctions `preProcess ()` et `factor ()`.

Pour la segmentation des données, elle a démontré le partitionnement des données, la validation croisée et comment segmenter des séries chronologiques.

Enfin, il a standardisé les données. Sélectionnez des variables et entraînez-vous à utiliser le modèle.

Contenu principal et explication

1.Importez le package caret

```
library(caret)
```

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

```
## Loading required package: ggplot2
```

Si nous voulons effacer le répertoire de travail avant cela, utilisez:

```
rm(list = ls())
```

2. Les fonctions

- Supprimer les valeurs manquantes

```
# dat=na.omit(dat1)
# nrow(dat)
```

- L’auteur a présenté en détail l’utilisation de la fonction `preProcess()`. Il existe deux méthodes, l’une est la **méthode médiane** (`method = “medianImpute”`) et l’autre est la **méthode kNearest Neighbor** (`method = “knnImpute”`) .

```
#imputation_k = preProcess (datTrain, method = "medianImpute")
#datTrain1 = prédire (imputation_k, datTrain)
#datTest1 = prédire (imputation_k, datTest)
```

- La fonction “**facteur**” est utilisée pour convertir le type de données.

```
# dat$Gender =factor(dat$"Gender", levels = c("F","M"),labels = c("Femme","homme"))
```

- Pour partitionner les données, nous utiliserons la fonction “**createDataPartition**”.
- Pour standardiser les données, nous utiliserons la fonction “**preProcess**”.

Evaluation et résumer

Selon mes critères, je pense que c’est un article supérieur à la moyenne.

Tout d’abord, son contenu est très riche, et un exemple complet a été démontré en détail. Elle a beaucoup de descriptions détaillées sur le contenu spécifique et l’utilisation de la fonction. Le cadre logique est très bon.

Mais le seul inconvénient est que tous ses textes sont identiques à l’exception du titre, vous ne pouvez donc pas voir le point principal d’un coup d’œil. Et aucune bibliographie n’est donnée, il est donc impossible pour les gens de mieux comprendre.