MNIST et Fashion MNIST (Illustration dans R) de Kanlanfeyi Kabirou & Hounsinou Jordy

Tarik HAKAM

22/12/2020

1. Critères d'évaluation

1. Comportement du Rmd à l'exécution
2. Qualité de la rédaction du dossier
3. Accessibilité, didactisme et pertinence du dossier
4. Qualité et lisibilité du Rmarkdown
5. Qualité des applications permettant d'illustrer le package

2. Lien vers le document commenté

En cliquant **ici**, vous trouverez le lien menant au GitHub de Kanlanfeyi Kabirou hébergeant le fruit de sa collaboration avec Hounsinou Jordy.

3. Auteurs du document commenté

Le document évalué dans le cadre de ce rendu a été produit par Kanlanfeyi Kabirou & Hounsinou Jordy, étudiants en Master of Science Data Management à Paris School of Business.

4. Synthèse du document

Le but de ce document est de fournir une présentation des bases de données MNIST (Modified ou Mixed National Institute of Standards and Technology) et Fashion MNIST qui sont des bases d'apprentissage des technologies de Machine Learning et de Deep Learning.

Ce document présente également des applications d'algorithmes de Machine Learning afin d'illustrer leur utilité, comme par exemple :

- dans la prédiction de chiffres (pour la base de données MNIST)
- dans la prédiction d'articles (pour la base de données Fashion)

La base de données MNIST est une base de données de chiffres écrits à la main. Créée dans un 1^{er} temps pour répondre à un problème de reconnaissance de l'écriture manuscrite, elle est devenue un test standard grâce à son efficacité pour les algorithmes d'apprentissage. Elle regroupe 60,000 images d'apprentissage et 10,000 images de test.

La base de données Fashion MNIST regroupe également un jeu de données contenant 70 000 images de vêtements (articles Zalando) en niveaux de gris réparties sur 1 des 10 catégories. La répartition des données "Apprentissage-Test" est identique à la précédente.

Le processus de reconnaissance des chiffres et des articles de mode est constitué de plusieurs étapes spécifiques et utilise différents algorithmes de Machine Learning ou de Deep Learning dont principalement les 2 suivants:

- **Naive Bayes** : algorithme de classification basé sur le théorème de Bayes, lui-même basé surles probabilités conditionnelles.
- **Random Forest** : algorithme d'apprentissage automatique qui combine les concepts de sous-espaces aléatoires et de bagging. Le concept de cet algorithme est basé sur l'apprentissage de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.

Ces algorithmes serviront à entrainer les données afin de construire des modèles prédictifs qui seront ensuite eux-mêmes utilisés pour des tests de nouvelles données. Une comparaison de ces modèles est réalisé dans ce document, sur la base de leur précision, autour de la prédiction et sur de nouvelles images.

5. Extrait commenté des parties de code

La partie de code la plus significative à retenir d'après moi est la partie consistant à la construction des algorithmes et des modèles.

Dans un 1^{er} temps, les données ont été scindées en 2 groupes :

- un pour l'entrainement des données afin de construire des modèles prédictifs avec des algorithmes de Machine Learning,
- un autre afin de tester ces modèles.

```
train_mnist <-  sample_frac(mnist, 0.8)
test_mnist <- anti_join(mnist, train_mnist)
train_fashion <-  sample_frac(fashion, 0.8)
test_fashion <- anti_join(fashion, train_fashion)
```

Pour ce faire, ils ont utilisé la fonction `sample_frac`, fournie par la librairie *dplyr* et il a été choisi d'établir un ratio de 80/20, répartissant les données de la façon suivante :

- 80% des données pour l'entraînement
- 20% des données pour le test

Ensuite, ils ont construit les modèles.

Pour optimiser la durée de traitement de la prédiction de **randomForest**, les modèles ont été construit en minimisant l'ajout de paramètres au strict nécéssaire . Le nombre d'arbres de décision, **"ntree"**, a été défini à 10, car un chiffre plus élévé nécessiterait davantage de mémoire et une durée d'exécution plus longue.

Ici, ils ont utilisé deux fois le même algorithme pour les deux jeux de données.

- **Random Forest**

```
rf_MNIST <- randomForest(label ~ ., data = train_mnist, ntree = 10)
pred_MNIST1 <- predict(rf_MNIST, test_mnist)
rf_FASH <- randomForest(label ~ ., data = train_fashion, ntree = 10)
pred_FASH1 <- predict(rf_FASH, test_fashion)
```

- **Naive Bayes**

```
bayes_MNIST <- randomForest(label ~ ., data = train_mnist)
pred_MNIST2 <- predict(bayes_MNIST, test_mnist)
bayes_FASH <- randomForest(label ~ ., data = train_fashion)
pred_FASH2 <- predict(bayes_FASH, test_fashion)
```

Puis, ils ont utilisé des matrices de confusion pour évaluer les modelès construits.

```
cm_rf1 <- confusionMatrix(pred_MNIST1, test_mnist$label)
cm_rf2 <- confusionMatrix(pred_FASH1, test_fashion$label)
cm_nb1 <- confusionMatrix(pred_MNIST2, test_mnist$label)
cm_nb2 <- confusionMatrix(pred_FASH2, test_fashion$label)
```

Après les tests des modèles sur les deux bases de données, les auteurs ont procédé à l'affichage des résultats par comparaison à travers une matrice de 2×2 :

```
valeurs <- matrix(c(cm_nb1$overall["Accuracy"],
                    cm_nb2$overall["Accuracy"],
                    cm_rf1$overall["Accuracy"],
                    cm_rf2$overall["Accuracy"]),
                  ncol = 2)
colnames(valeurs)<- c("Naive Bayes", "Random Forest")
rownames(valeurs)<- c("MNIST", "Fashion MNIST")
tableau <- as.table(valeurs)
print(tableau)
```

6. Evaluation du travail suivant les 5 critères précités

1. Comportement du Rmd à l'exécution

Malheureusement, il est difficile d'exécuter le Rmd sans érreur et cela est dû à l'absence des bases de données :

- train.csv
- fashion.csv

Les deux auteurs ont omis de fournir les fichiers CSV. Afin de faciliter l'exécution du RMarkdown et la visualisation de leurs résultats, ils auraient dû les ajouter à leur GitHub, car la recherche de ces deux fichiers plats n'est pas aisé.

2. Qualité de la rédaction du dossier

La rédaction de ce document est de bonne qualité. Le language est adapté et il existe un fil conducteur dans la construction de leur pensée.

3. Accessibilité, didactisme et pertinence du dossier

La lecture de ce dossier est aisée et accessible au plus grand nombre, même aux néophytes. Les descriptions sont relativement bien explicitées et bien illustrées.

Les auteurs arrivent à faire adhérer le lecteur à leur production. Le document a une réelle vocation à transmettre une connaissance et le but, pour ma part, est atteint.

Le sujet est pertinent et a vocation à servir de mémo pour de futurs tests.

4. Qualité et lisibilité du RMarkdown

Le RMarkdown est généralement bien écrit, lisible et aéré.

J'aurais éventuellement une remarque, ainsi qu'une suggestion :

1. Les figures `![MNIST.](MnistExamples.png)` et `![Fashion MNIST.](fashionMNIST.jpeg)` s'affichent respectivement en point 3 (*Algorithmes choisis*) et 4 (*Exemple de code R*) alors qu'elles ont été introduites dans le code en point 2 (*Présentation des bases de données*).
2. J'aurais, pour ma part, réarrangé le code en ligne 158 afin que celui-ci s'affiche plus proprement dans le document PDF (*Affichage des résultats*)

Autrement, il s'agit d'un code bien réalisé.

5. Qualité des applications permettant d'illustrer le package

Les applications sont de bonne qualité, maitrisées et bien expliquées.

7. Conclusion

Selon moi, il s'agit d'un bon travail.

Les deux auteurs ont bien documenté leur dossier. C'est un travail bien construit et recherché.

Dommage que les fichiers CSV soient manquants.

Vous retrouvez ce document sur mon **GitHub**.