

Compte Rendu

Dockeriser un Arbre de décision sur python

1) Partie 1 : Un arbre de décision en script python

Ce script a été développé sur Spyder. J'ai utilisé les librairies pandas et sklearn. Le Dataset est celui qui m'a été fourni, voici ses caractéristiques :

Data Set Characteristics:	Multivariate	Number of Instances:	286	Area:	Life
Attribute Characteristics:	Categorical	Number of Attributes:	9	Date Donated	1988-07-11
Associated Tasks:	Classification	Missing Values?	Yes	Number of Web Hits:	528012

Ce script est décomposé en plusieurs parties :

- a. Loading & Preprocessing des données
 - On charge les données
 - On enlève les valeurs manquantes
 -
- b. Processing des données
 - Création de dummies pour l'entraînement du modèle (passage du type « categorical » à « binaire »)
 - Séparation des variables explicatives et de la vérité terrain
 - Normalisation des variables explicatives
 - Création du dataset de training et de test
- c. Création de l'Arbre de décision
 - Il faut penser à utiliser le paramètre `class_weight = 'balanced'` car il n'y a pas autant de lignes où la vérité terrain est "no-recurrence-events" que "recurrence-events"
 - Prints des résultats, et notamment la précision moyenne sur le dataset de test.

2) Partie 2 : Docker

Ce travail a été effectué sur Windows 10.

Ce dossier docker sera constitué de tous les fichiers dont on a besoin pour créer une image :

- Le dockerfile
- Le dataset
- Le script Python

Ainsi, on va voir les commandes docker effectuer pour réussir à Dockeriser mon script python.

3) Docker file

Ce docker file permet de faire savoir quelle version et quelles librairies de python j'ai besoin pour faire fonctionner mon script python.

```
FROM python:3
```

J'indique ici la version de python et l'environnement à utiliser.

```
ADD arbre_classification.py /
```

Je rajoute avec la commande docker ADD [mon_script.py].

```
RUN pip install pandas
```

```
RUN pip install sklearn
```

Je demande d'installer les bibliothèques pandas et sklearn avec la commande docker RUN pip install [ma_librairie].

```
COPY breast-cancer_data.csv ./breast-cancer_data.csv
```

J'importe mon dataset breast-cancer_data.csv.

```
CMD [ "python", "./arbre_classification.py" ]
```

Enfin, j'exécute mon script, ici appelé arbre_classification.py.

Voilà notre dockerfile terminé, je dois maintenant créer une image.

4) Build de l'image (une fois dans le dossier avec la dockerfile

Avec le shell windows, je me place dans mon dossier avec le docker file, le dataset, et le script.

Ensuite, je crée mon image qui s'appellera python_arbre.

```
docker build -t python_arbre .
```

5) Run de l'image

Maintenant que l'image a été créée, il faut l'exécuter en tant que container grâce à la commande suivante :

```
docker run python_arbre
```

Voilà, j'ai créé un container de mon script en python

je vais maintenant le rendre disponible sur mon compte dockerhub.

6) Commit & Push

Il va falloir faire une commande commit pour ensuite push notre container.

Je regarde l'ID du conteneur de l'image python_arbre grâce à la commande suivante :

```
docker ps -a
```

Je trouve comme résultat que l'ID est : 1cff609c8baa

```
docker commit 1cff609c8baa theodoremifa/python_arbre:v2
```

On se connecte à docker puis on push sur dockerhub avec la commande suivante :

```
docker push theodoremifa/python_arbre:v2
```

Conclusion:

Grâce à docker, n'importe qui ayant docker peut exécuter ce script python sans avoir à se soucier des configurations et de l'environnement de travail de son pc.

Voici la commande nécessaire à télécharger mon image :

```
docker pull theodoremifa/python_arbre:v2
```