COURS
07_dvc_dagshub_fr.html

DIFFICULTÉ
Normal

TEMPS APPROXIMATIF
1h00

DataScientest

Evaluation

Cette évaluation est composé de quatre parties. On travaillera sur un dataset qui contient les informations concernant le traitement des minéraux. Plus spécifiquement sur le processus de flottation utilisé pour concentrer la silice à partir du minerai. Le jeu de données contient divers paramètres opérationnels et leur impact sur la concentration de silice, qui est la variable cible. Le dataset que vous téléchargerez contient les informations suivantes :

- ave_flot_air_flow : Débit d'air moyen dans le processus de flottation.
- ave_flot_level : Niveau moyen dans les cellules de flottation.
- iron_feed : Quantité de minerai de fer entrant dans le processus de flottation.
- starch_flow : Débit d'amidon utilisé comme réactif dans le processus de flottation.
- amina_flow : Débit d'amine utilisé comme collecteur dans le processus de flottation.
- ore_pulp_flow : Débit de la pulpe de minerai.
- ore_pulp_pH : Niveau de pH de la pulpe de minerai, qui peut affecter le processus de flottation.
- ore_pulp_density : Densité de la pulpe de minerai, un autre paramètre critique dans le processus de flottation.
- silica_concentrate : Concentration de silice dans le produit final, qui est la variable cible.

L'objectif de la modélisation est de comprendre et de modéliser comment différents paramètres opérationnels affectent la concentration de silice dans le processus de flottation. Le jeu de données contient 1817 entrées, toutes les colonnes sont de type float64 et il n'y a pas de doublons ni de valeurs manquantes.

Les données sont accessibles à travers le lien suivant : https://datascientest-mlops.s3.eu-west-1.amazonaws.com/mlops_dvc_fr/raw.csv.

Quant à l'objectif de cet examen, il s'agit de mettre en place un workflow de la modélisation en utilisant DVC et DagsHub. Le rendu final sera sous forme de dépôt sur DagsHub que vous partagerez avec https://dagshub.com/licence.pedago en le mettant comme collaborateur avec des droits de lecture seulement. Sur la plateforme, vous enverrez un `.zip` contenant un `.md` avec votre nom, prénom, adresse mail et le lien vers votre dépôt DagsHub.

On commence par fork et cloner le dépôt : https://github.com/DataScientest-Studio/examen-dvc. Vous observerez qu'il contient l'architecture que le projet devrait avoir. L'arborescence ressemble à :

```
├── examen_dvc
|   ├── data
|   |   ├── processed
|   |   └── raw
|   ├── metrics
|   ├── models
|   |   ├── data
|   |   └── models
|   ├── src
|   └── README.md.py
```

Pensez, dans un premier temps, à créer un environment virtuel dans lequel vous travaillerez le long de l'examen.

1. Création des scripts

La première étape consiste à construire les scripts nécessaires que l'on utilisera dans le workflow de cette modélisation. On s'attend à avoir au moins 5 scripts, chacun visant une des étapes suivantes :

1. Split des données en ensemble d'entraînement et de test. Notre variable cible est `silica_concentrate` et se trouve dans la dernière colonne du dataset. L'issu de ce script seront 4 datasets (X_test, X_train, y_test, y_train) que vous pouvez stocker dans `data/processed`.

2. Normalisation des données. Comme vous pouvez le noter, les données sont dans des échelles très variés donc une normalisation est nécessaire. Vous pouvez utiliser des fonctions pré-existantes pour la construction de ce script. En sortie, ce script créera deux nouveaux datasets : (X_train_scaled, X_test_scaled) que vous sauvegarderez également dans `data/processed`.

3. GridSearch des meilleurs paramètres à utiliser pour la modélisation. Vous déciderez le modèle de regression à implémenter et les paramètres à tester. À l'issu de ce script on aura les meilleurs paramètres sous forme de fichier `.pkl` que l'on sauvegardera dans le dossier `models`.

4. Entraînement du modèle. En utilisant les paramètres retrouvés à travers le GridSearch, on entraînera le modèle en sauvegardant le modèle entraîné dans le dossier `models`.

5. Evaluation du modèle. Finalement, en utilisant le modèle entraîné on évaluera ses performances et on fera des prédictions avec ce modèle de sorte qu'à la fin de ce script on aura un nouveau dataset dans `data` qui contiendra les predictions ainsi qu'un fichier `scores.json` dans le dossier `metrics` qui récupérera les métriques d'évaluation de notre modèle (i.e. mse, r2, etc).

Chacun des scripts devra se retrouver dans le dossier correspondant (que ce soit `src/data` ou bien `src/models`). Si vous l'estimez nécessaire, vous pouvez en rajouter d'autres scripts. Faites attention de donner des noms parlants et pertinents à vos scripts.

2. Connection de votre dépôt à DagsHub

Si ce n'est toujours pas le cas et si vous êtes en train de travailler sur GitHub, connectez ce dépôt à votre compte DagsHub. Puis, faites de DagsHub votre emplacement distant pour le suivi de la donnée. N'oubliez pas d'adapter votre `.gitignore` en fonction de vos besoins.

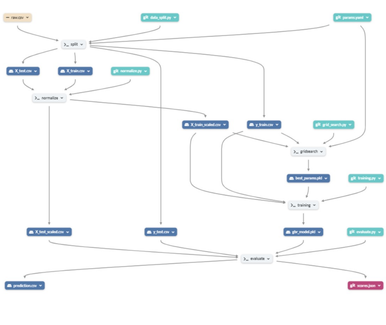
3. Pipeline DVC

À l'aide des commandes DVC vues dans le cours, mettez en place une pipeline qui reproduira le workflow de votre modèle. Il faudra bien utiliser les scripts que vous avez mis en place lors de l'étape 1.

4. Rendu

Pour rendre l'examen sur la plateforme vous enverrez un `.zip` contenant un `.md` avec votre nom, prénom, adresse mail et le lien vers votre dépôt DagsHub. Puis vous devez partager votre dépôt avec https://dagshub.com/licence.pedago en le mettant comme collaborateur avec des droits de lecture seulement. Pour valider l'examen on s'attend à trouver dans ce dépôt :

- Les 5 scripts de preprocessing, modélisation et évaluation du modèle détaillés dans l'étape 1.
- Un dossier `.dvc` avec un fichier de `config` explicitant les informations par rapport à l'emplacement distant.
- Un fichier `.pkl` dans l'onglet _models_ de DagsHub avec le modèle entraîné.
- Un fichier `.json` dans le dossier `metrics` avec les métriques d'évaluation du modèle.
- Un fichier `dvc.yaml` avec les étapes de la pipeline DVC ainsi qu'un fichier `dvc.lock` avec les informations de la sauvegarde.
- L'onglet _data_ devra bien afficher les données.
- Le schéma de la pipeline crée par DagsHub qui doit ressembler à :



Cours terminé ?

Marquer comme terminé