Projet Final

Début 9h, retour 17h30

(Aucun retard ne sera permis)

Objectif

- Simuler un projet d'entreprise réel où plusieurs développeurs travaillent sur le même projet et chacun développe une fonctionnalité dans une branche différente puis fusionne sa version finale avec la branche principale (main).
- Créer ensuite un docker qui se base sur le repository github (branche main) et se met à jour automatiquement en production.

Consignes

Un repository github contenant des templates vous aidera à avancer : xxx.

- Développer un modèle de clustering s'appuyant sur la réduction de la dimensionalité via l'ACP, l'AFC et UMAP (on retiendra un space réduit de dimension 20). Cette approche dite tandem ou séquentielle prend en entrée des données textuelles (les données NG20 comme présenté sur le template, on utilisera 2000 documents seulement) puis combine une des méthodes de réduction de dimension avec un algorithme de clustering; on choisit de retenir k-means mais tout autre algorithme de clustering pourrait être utilisé.

- Le point de départ est un repository GitHub que vous créez contenant un README et un .gitignore. Rajouter ensuite un fichier main.py qui servira à évaluer chacune des approches (ACP+kmeans, AFC+kmeans, UMAP+kmeans) à l'aide des métriques NMI, ARI et Accuracy à partir des classes connues (vous pouvez vous appuyez sur le template_main.py).

- L'objectif est de travailler en collaboration pour remplir le fichier main.py mais de manière indépendante et asynchrone en développant une approche par personne dans une nouvelle branche (vous pouvez vous appuyer sur le notebook template_branche.ipynb pour développer et tester le modèle). Une fois la méthode terminée et testée vous pouvez la fusionner avec la branche main (en utilisant un merge pull request) en rajoutant votre notebook dans un dossier appelé experiments par exemple, et en rajoutant votre approche au main.py.

- Faire un clone de la branche main finale sur une machine et créer une image docker dans le but d'executer le fichier main.py et retourner le résultat de clustering de chaque méthode.

- Pousser l'image dans docker Hub.

Rendu :

- Le lien vers github et dockerHub dans le fichier : lien
- Si vous avez créé un volume en local dans la partie docker, spécifier sur le README comment l'utiliser, et mettre la commande pour le lancer en local.

Important : Ne plus faire de push après avoir rendu le projet

Évaluation :

Le projet sera évalué en fonction de :

- Fonctionnement de la solution.
- Integration des fonctionnalités et pratiques vues en cours.
- L'idée n'est pas d'avoir les meilleurs résultats du modèle mais de répondre aux besoins en travaillant en collaboration et faciliter l'intégration de la solution finale.

Bonus :

- Développer le modèle en local avec docker et monter un volume sur le projet, ça vous permet de remplir le Dockerfile au fur et à mesure et de travailler ensemble sur la même version de python et des librairies.
- Améliorer la qualité du rendu final, par exemple : permettre à l'utilisateur de choisir le(s) modèle(s) qu'il souhaite tester, ou bien executer le modèle dans un backend. Toute autre proposition sera appréciée.
- Visualisation des données sur un plan à l'aide de l'ACP, l'AFC et UMAP.
- Rédaction d'une bonne documentation (README).
- Sauvegarder les données pour ne pas les télécharger à chaque instantiation d'un nouveau conteneur (ou bien les installer dans l'image docker).
- Réaliser une cross validation.
- Toute autre méthode de réduction de la dimension ou algorithme de clustering seraient très appréciés.

Bon courage !