Réalisez une application de recommandation de contenu

Liens

GitHub : https://github.com/AnodeGrindYo/OC_IA_P09.git

Application Web : https://agrecommends.azurewebsites.net

Sommaire

1. Contexte
2. Présentation du jeu de données
3. Modélisation
4. Architecture serverless
5. Déploiement
6. Architecture cible
7. Conclusion

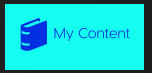
Contexte

My Content

- **My Content** est une start-up qui veut encourager la lecture en recommandant des contenus pertinents pour ses utilisateurs.

- Construction d'un **premier MVP** qui prendra la forme d'une **application** : une solution de **recommandation d'articles** et de livres à des particuliers.

Objectifs



- Utilisation d’un jeu de données disponible en ligne pour développer le MVP. (intéractions des utilisateurs avec les articles)
- Entraînement, tests et comparaison de plusieurs **modèles de recommandation**
- Architecture **serverless**.

Présentation du jeu de données

My Content

- Provient de Globo.com, contient les fichiers suivants :
  - **articles_metadata.csv** : informations sur les articles publiés
  - **articles_embeddings.pickle** : embedding de tous les articles (représentation vectorielle)
  - **clicks_sample.csv** : échantillon des intéractions des sessions des utilisateurs
  - **/clicks** : intéractions des sessions des utilisateurs (un fichier par heure)

- **364 047** articles
- **2 988 181** clics
- **322 897** utilisateurs
- **250** features d'embedding du contenu des articles

Modélisation

My Content

2 approches :

- **Content-Based :**

Basé sur les préférences de l'utilisateur et recommande des articles similaires aux articles qu'il a déjà lus.

\+ Pas besoin de données sur les autres utilisateurs

\+ Peut capturer les intérêts spécifiques d'un utilisateur

\- Dépend du nombre d'articles lus par l'utilisateur, et ne peut deviner d'autres intérêts de l'utilisateur

- **Collaborative-Filtering :**

Basé sur les préférences des autres utilisateurs ayant lus les mêmes articles, en recommandant des articles lus par les autres utilisateurs aux préférences communes.

\+ Peut permettre à l'utilisateur de découvrir de nouveaux intérêts

\+ Pas de connaissance sur le domaine requise (ici les articles), embedding automatiquement appris

\- Difficulté à recommander les nouveaux éléments, car peu ou pas d'intéraction avec ceux ci

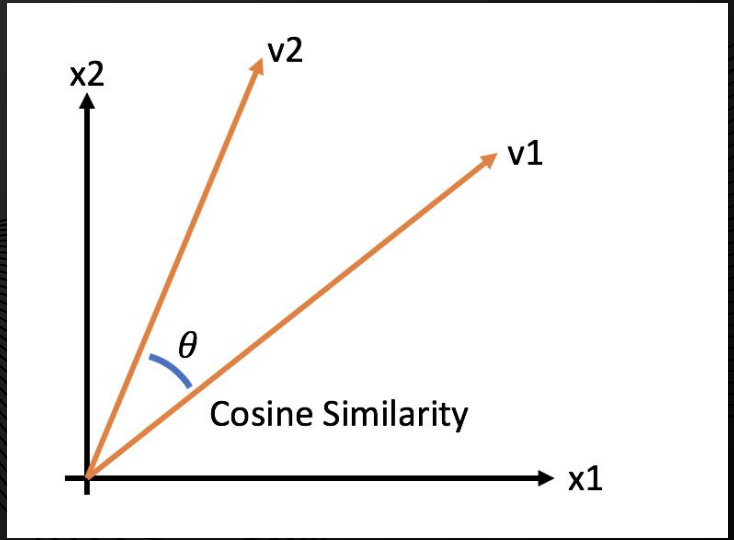
Modélisation : Content-Based

My Content

- Utilise le calcul du cosinus entre les articles lus par l'utilisateur et les autres articles, projetés via leurs features de la matrice d'embedding dans un espace vectoriel (de dimension 250).
- La prédiction du modèle retourne les n plus proches articles par rapport aux articles déjà lus.

ACP :

- Permet d'alléger le fichier d'embedding pour son utilisation dans le cloud
- On passe de 250 features à 70, en conservant une variance de 0.977.
- Résultats similaires par rapport à l'utilisation de l'embeddings original



Modélisation : Collaborative-Filtering :

My Content

- Benchmark de différents modèles (SVD, SVDpp, SlopeOne, NMF…)

Algorithm	test_rmse	test_mae	fit_time	test_time
NMF	0.003752	0.001119	28.937999	4.921168
CoClustering	0.003765	0.001128	45.746334	2.191500
BaselineOnly	0.003435	0.001391	3.512999	1.803000
SVDpp	0.003804	0.001489	16.660834	13.544000
SlopeOne	0.003673	0.001573	2.584334	4.942999
SVD	0.005112	0.001670	12.457150	5.390833
NormalPredictor	0.004463	0.002452	1.590500	1.788333

- Utilisation du modèle **NMF** de la librairie Surprise qui prend en entrée des données avec 3 colonnes :
  - ID de l'utilisateur
  - ID de l'article
  - rating

Modélisation : Collaborative-Filtering :

My Content

Fine Tuning :

- Recherche les meilleurs paramètres par validation croisée. (GridSearchCV)

- Entraînement avec les meilleurs paramètres et la totalité du jeu de données.

Recommandations :

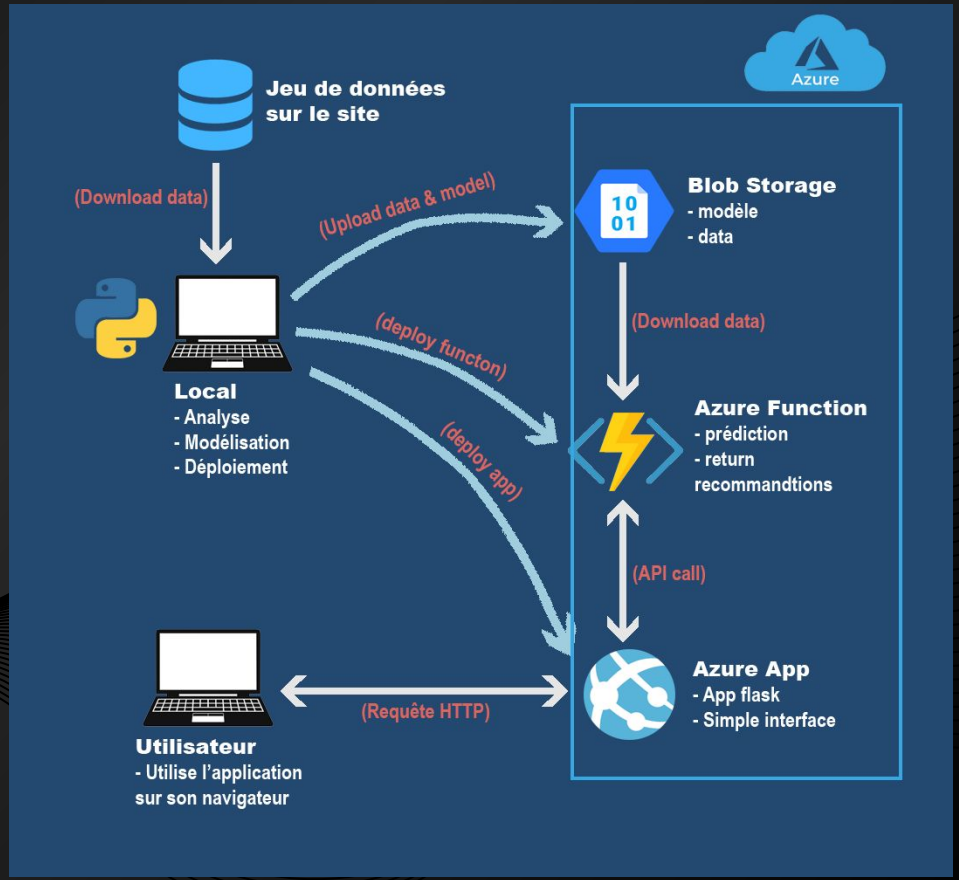
- La prédiction retourne un score compris entre 0 et 1 pour chaque article à partir d'un identifiant utilisateur, puis on retourne les n scores les plus élevés.

Architecture serverless

My Content

Fonctionnalités Azure utilisées :

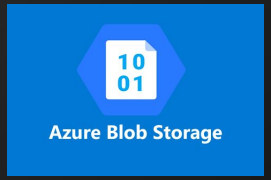
- Blob Storage
- Azure Function
- Azure App



Déploiement

Stockage sur Azure Blob :

- Modèle entraîné
- Embeddings avec ACP
- Dataframe contenant les clics pour chaque utilisateur



Déploiement sur Azure Function:

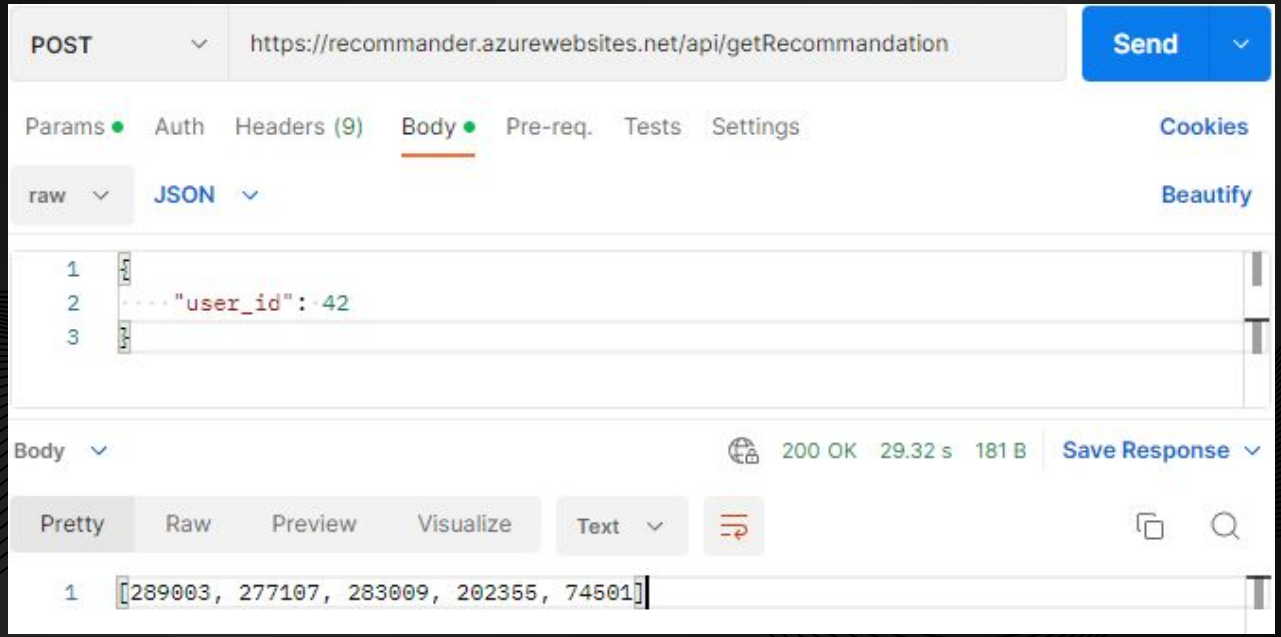
- Azure Function est la partie serverless. Notre fonction effectue la recommandation pour un utilisateur en utilisant les ressources stockées sur Azure Blob Storage.

- Cette fonction peut être utilisée via un appel API.

Déploiement

Démo d'Azure Function :

Fonction accessible à l'adresse suivante : https://recommander.azurewebsites.net/api/getRecommandation



Déploiement

Déploiement sur Azure App :

- Partie UI déployée en tant qu'application flask sur Azure App.

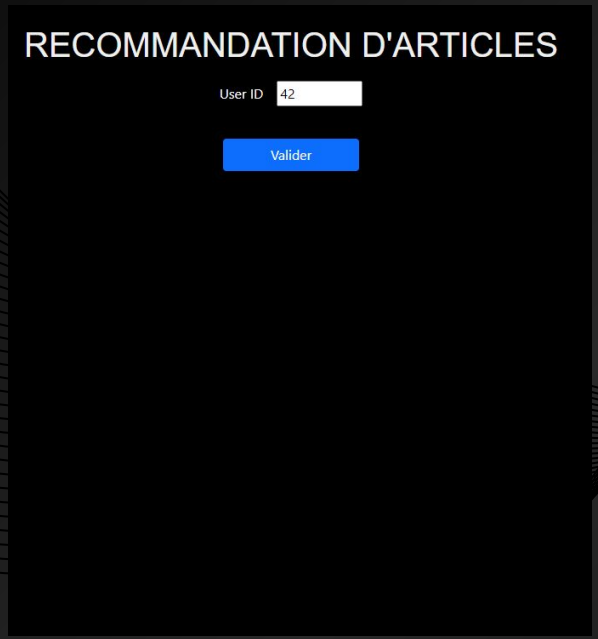
- L'application effectue l'appel à Azure Function via une requête API, en utilisant les données saisis par l'utilisateur, puis affiche le résultat.

Déploiement

My Content

Démo d'Azure App :

Application web accessible à l'adresse suivante : https://aqrecommends.azurewebsites.net

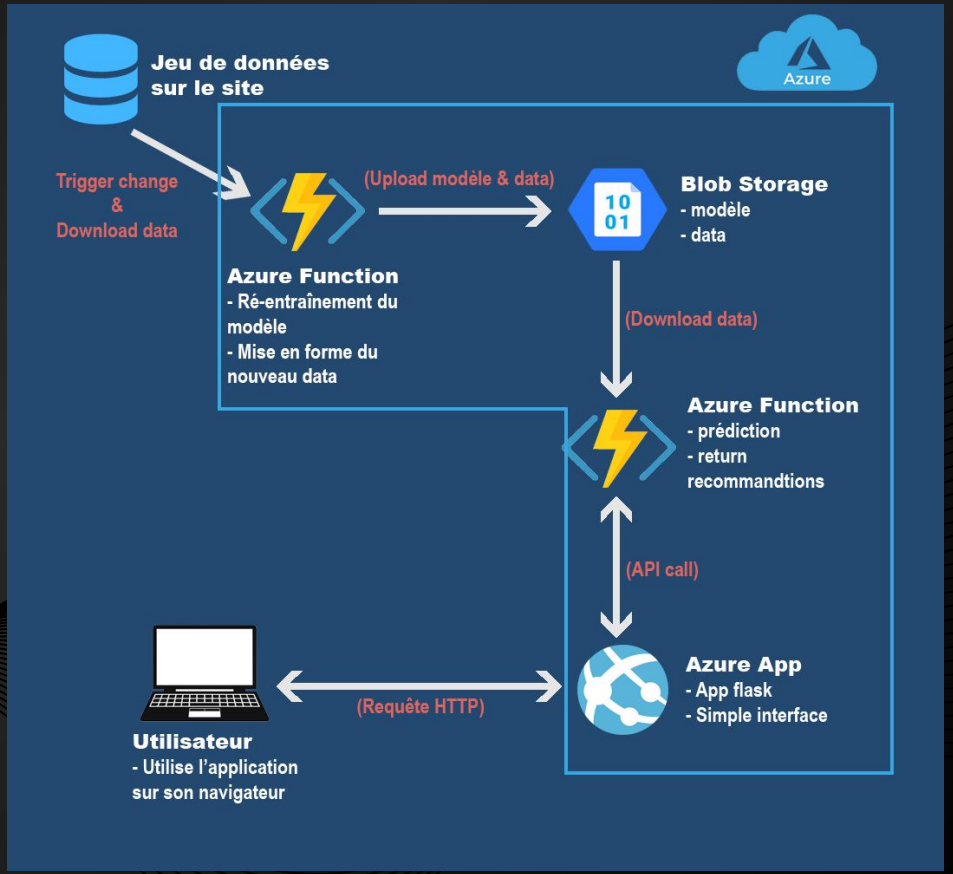


Architecture cible

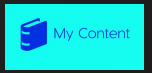
My Content

Permet de prendre en compte :

- les nouveaux utilisateurs
- les nouveaux articles.



Conclusion



Améliorations possibles :

- Modèle hybride utilisant à la fois le Content-Based et le Collaborative Filtering
- Mise en place de l'architecture cible, avec une nouvelle fonction Azure qui permettra de :
  - détecter un changement dans les données,
  - ré-entraîner le modèle
  - stockage des nouvelles données et du modèle sur Azure Blob.
- Amélioration de l'interface utilisateur