My Content

BEN ABBOU Tarik

29/07/2023

Liens

Application Web : https://p9-tarik-app-82458d7119cc.herokuapp.com

GitHub : https://github.com/Benabboutarik/p9_ingenieur_ia

Sommaire

1. Contexte

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

La start-up est en pleine construction d'un premier MVP qui prendra la forme d'une application, qui consiste en une solution de recommandation d'articles et de livres à des particuliers.

1. Contexte

Les objectifs

Utiliser un jeu de données disponible en ligne pour développer le MVP. (intéractions des utilisateurs avec les articles)

Entrainement, tests et comparaison de plusieurs modèles de recommandation

Mettre en place une architecture serverless pour un déploiement en production.

3. Modelisation

On va ici présenter les modèles qui seront utilisés. Il y a deux approches possibles pour recommander des articles aux utilisateurs :

Collaborative-Filtering

Content-Based

3. Modelisation

Content-Based

Ce modèle se base sur les préférences de l'utilisateur, en recommandant des articles similaires aux articles qu'il a déjà lu.

+ Le modèle n'a pas besoin de données sur les autres utilisateurs
Peut capturer les intérêts spécifiques d'un utilisateur

- Difficulté à représenter les caractéristiques des éléments (ici embedding déjà fourni)
Dépendant du nbr d'articles lus par l'utilisateur, et ne peut deviner d'autres préférences possibles

3. Modelisation

Content-Based

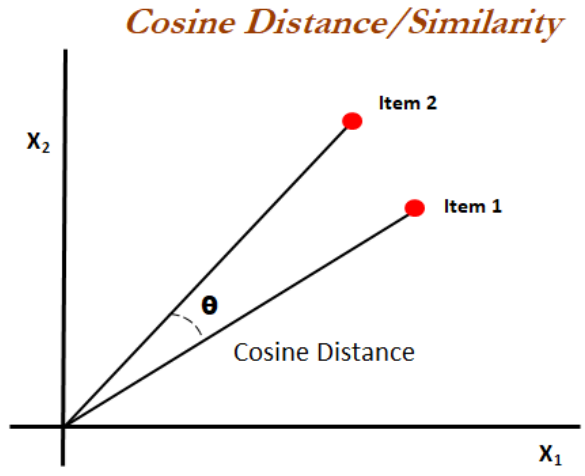
Le modèle utilise le calcul du cosinus entre les articles lus par l'utilisateur et les autres articles, qui sont projetés via leurs features de la matrice d'embedding dans un espace vectoriel (de dimension 250).

La prédiction du modèle retourne les n plus proches articles par rapport aux articles déjà lus.

Le fichier d'embedding étant volumineux, il faudrait l'alléger pour une utilisation future dans le Cloud.

L'ACP permet de passer de 250 features à 70, et de garder une variance de 0.977.

En testant le modèle utilisant l'embedding à 70 features, on obtient les mêmes résultats.



3. Modelisation

Collaborative-Filtering

Ce modèle se base sur les préférences des autres utilisateurs ayant lus les mêmes articles, en recommandant des articles lus par les autres utilisateurs aux préférences communes.

+ Peut aider l'utilisateur à découvrir de nouveaux intérêts
Pas besoin de connaissance sur le domaine (ici les articles), embedding automatiquement appris

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

3. Modelisation

Collaborative-Filtering

On utilise le modèle SVD (technique de factorisation de matrice) de la librairie surprise, qui prend un data contenant 3 colonnes :
- l'id du user
- l'id de l'article
- un "rating" qui correspond au ratio suivant :**Rating(user,article) =**

On va rechercher les meilleurs paramètres du modèle par validation croisée avec GridSerchCv.

On entraîne le modèle avec les meilleurs paramètres et la totalité du jeu de données.

	user_id	article_id	rating
0	0	68866	0.125
1	0	87205	0.125
2	0	87224	0.125
3	0	96755	0.125
4	0	157541	0.125

3. Modélisation

Collaborative-Filtering

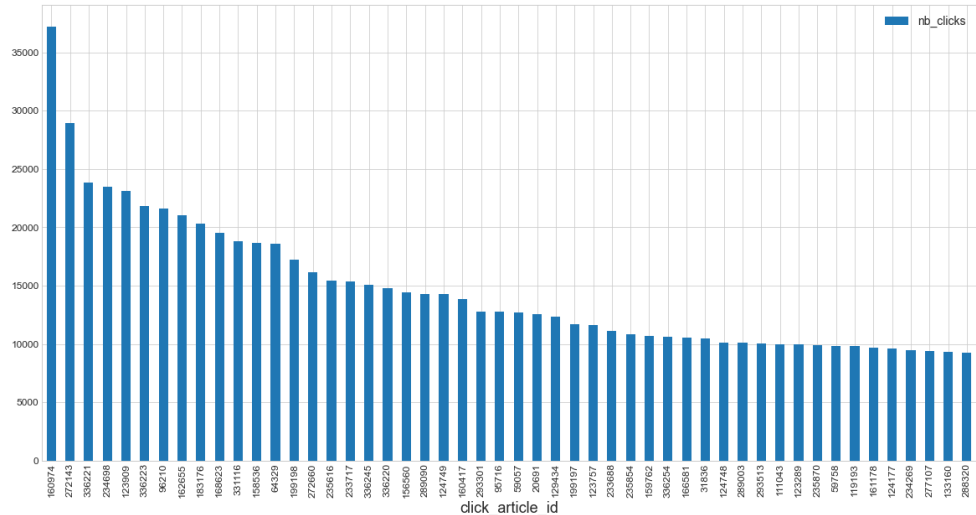
La prédiction du modèle retourne un score à partir d'un id de user et un id d'article. Ce score est compris entre 0 et 1, et est l’équivalent d’une prédiction du rating calculé précédemment.

Pour un user on va donc calculer ce score pour chaque article et retourner les n plus élevés.

3. Modélisation

Collaborative-Filtering

On voit que certains articles sont régulièrement recommandés. Il s'agit en fait des articles les plus lus



4. Deploiement

Stockage sur Azure Blob

On va stocker le modèle entraîné ainsi que les données nécessaires à son fonctionnement. Ces données seront ensuite chargées par notre Azure Function.

ACP a permis de réduire la taille de notre jeu de données d'embedding, le stockage étant limité pour l'offre gratuite.

Déploiement sur Azure Function

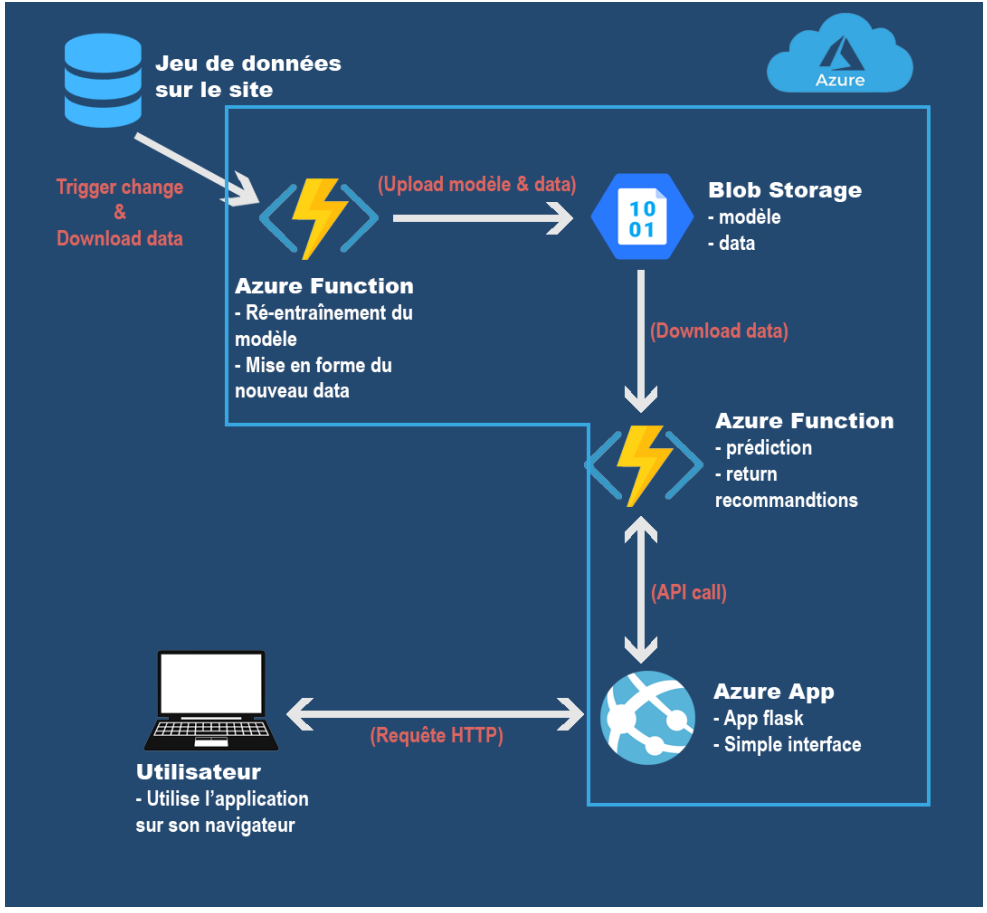
Azure Function est la partie serverless. On va déployer notre fonction, qui effectue la recommandation pour un id de user en utilisant les ressources stockées sur Azure Blob Storage

Cette fonction peut ensuite être utilisée via un appel API.

4. Deploiement

5. Architecture cible

Architecture cible permettant de prendre en compte la création de nouveaux utilisateurs et de nouveaux articles.



6. Conclusion

L'application actuelle est un premier MVP fonctionnel, offrant une base solide pour les améliorations futures. Les prochaines étapes comprennent la création d'un modèle hybride combinant Content-Based et Collaborative Filtering, la mise en place de l'architecture cible avec détection de changements de données pour ré-entraîner le modèle stocké sur Azure Blob, l'amélioration de l'interface utilisateur et le renforcement de la sécurité des échanges. Ces améliorations permettront des recommandations plus précises, une meilleure évolutivité et une sécurité accrue pour une expérience utilisateur optimale.