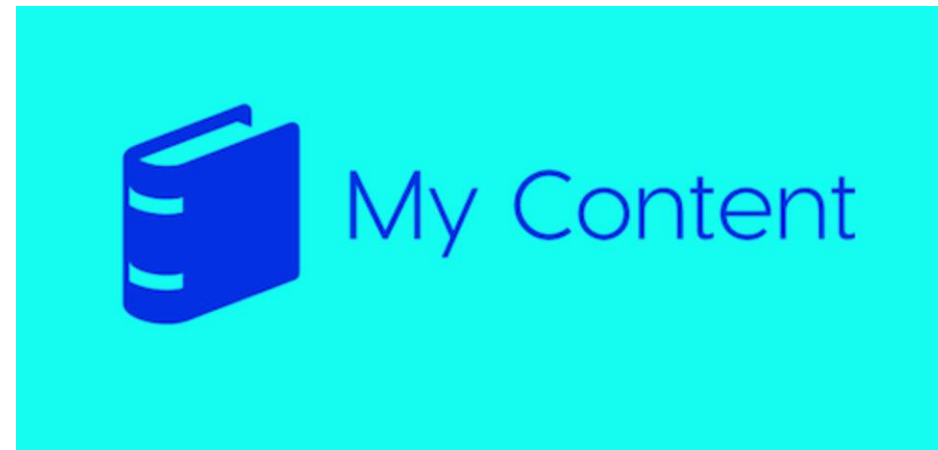


# Réalisez une application de recommandation de contenu





# Sommaire

PARTIE 1 – PROJET

PARTIE 2 – DONNÉES

PARTIE 3 – MODÉLISATION

PARTIE 4 – ARCHITECTURE

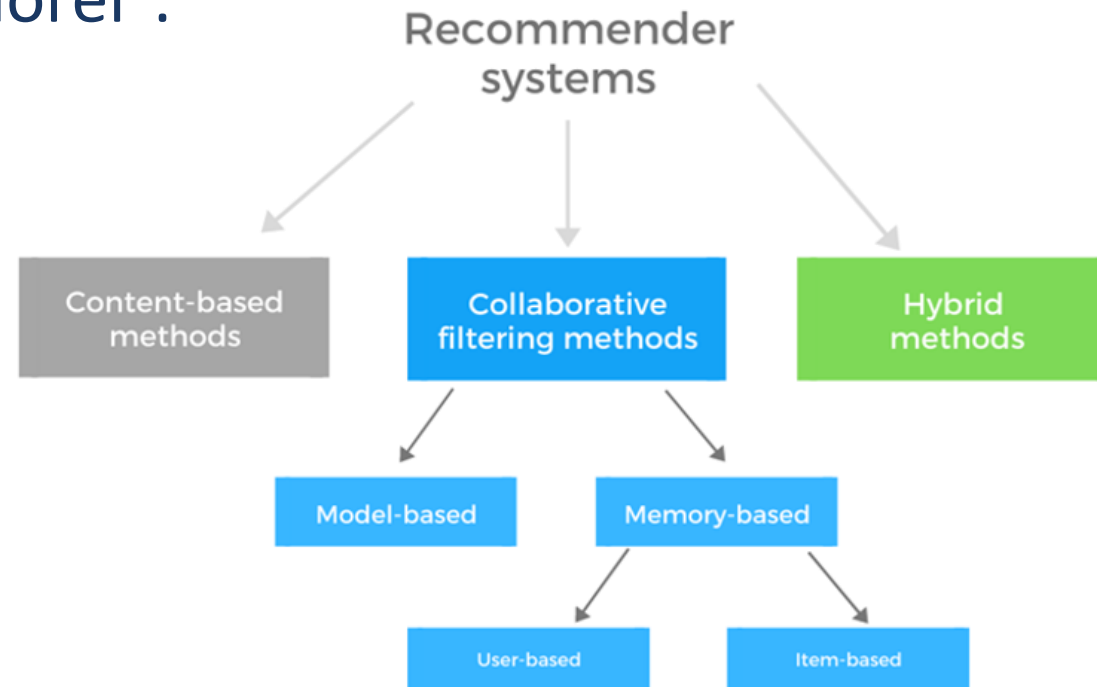
PARTIE 5 – SYSTÈME DE RECOMMANDATION

PARTIE 6 – ARCHITECTURE CIBLE

# PARTIE 1 – PROJET

# Le projet de l'entreprise

- Recommandation de contenus pour les utilisateurs
- Stade amont : développer un MVP
- Solutions à explorer :



# Le projet de l'entreprise

- Données à notre disposition : open source  
<https://www.kaggle.com/datasets/gspmoreira/news-portal-user-interactions-by-globocom>
- Système de recommandation / App : outil simple, au stade du MVP
- Architecture :
  - tester une solution *serverless*
  - s'interroger sur l'évolutivité : Quid des nouveaux utilisateurs ? articles ?

# Les outils utilisés

- Modélisation Collaborative Filtering **surprise**
- Création de l'UI 🍷 Flask
- Création des fonctions serverless et déploiement ⚡
- Stockage :
  - Emulateur (développement local) 
  - En ligne (pour le déploiement) 
- Intégration continue  **GitHub**
- Déploiement continue  GitHub Actions + ⚡

# PARTIE 2 – DONNÉES



# Dataset

- Des données relatives aux **interactions** des utilisateurs :

- `clicks_sample.csv`
- `clicks/ :`
  - `clicks_hour_000.csv`
  - `clicks_hour_001.csv`
  - ...
  - `clicks_hour_383.csv`
  - `clicks_hour_384.csv`

```
• user_id : l'identifiant de l'utilisateur
• session_id : un identifiant donné à la session dans laquelle le click a lieu
• session_start : horodatage du début de la session. De type Unix time, mais en ms
• session_size : nombre de clicks dans la session d'utilisation
• click_article_id : identifiant de l'article sur lequel le click a eu lieu
• click_timestamp : horodatage du click
• click_environment : environnement d'utilisation (1 - Facebook Instant Article, 2 - Mobile App, 3 - AMP (Accelerated Mobile Pages), 4 - Web)
• click_deviceGroup : type d'appareil utilisé (1 - Tablette, 2 - télévision, 3 - vide (inconnu), 4 - smartphone, 5 - ordinateur de bureau)
• click_os : système d'exploitation de l'appareil
• click_country : identifiant donné au pays de l'utilisateur
• click_region : identifiant donné à la région de l'utilisateur
• click_referrer_type : inconnu
```

- Des données relatives aux **articles** :

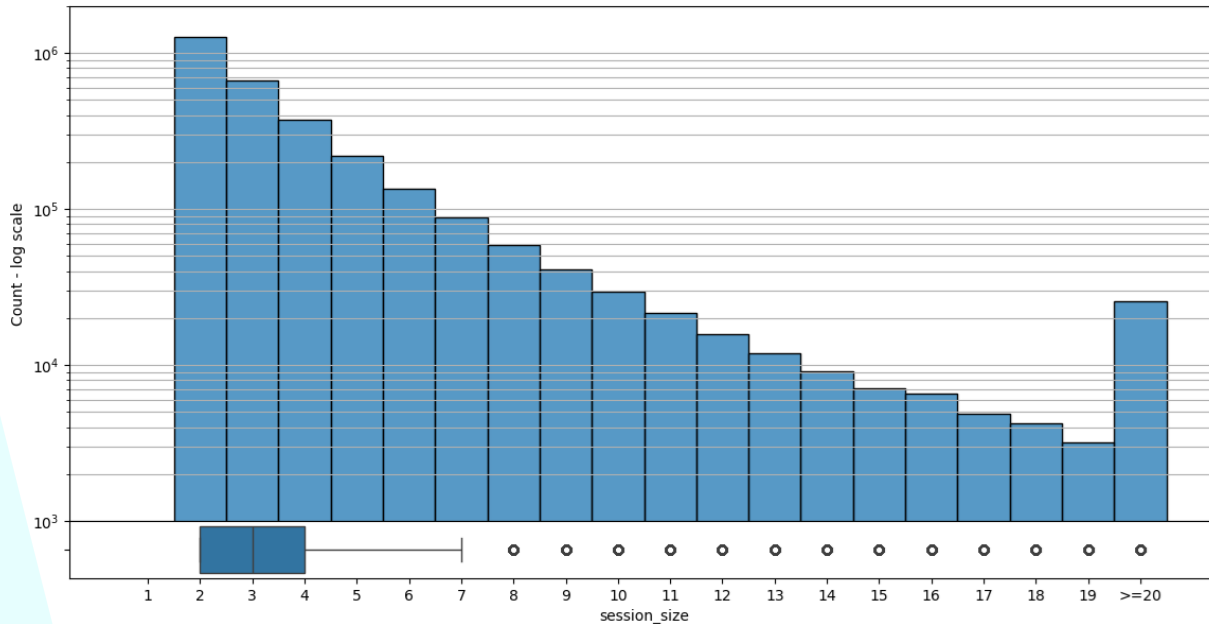
- `articles_metadata.csv`

```
• article_id : l'identifiant de l'article, le même que click_article_id
• category_id : un identifiant donné à la catégorie de l'article.
• created_at_ts : horodatage de la rédaction de l'article. De type Unix time, mais en ms
• publisher_id : l'identifiant de l'éditeur. Inutilisable car ne comporte qu'une seule valeur.
• words_count : nombre de mots dans l'article
```

- `articles_embeddings.pickle`

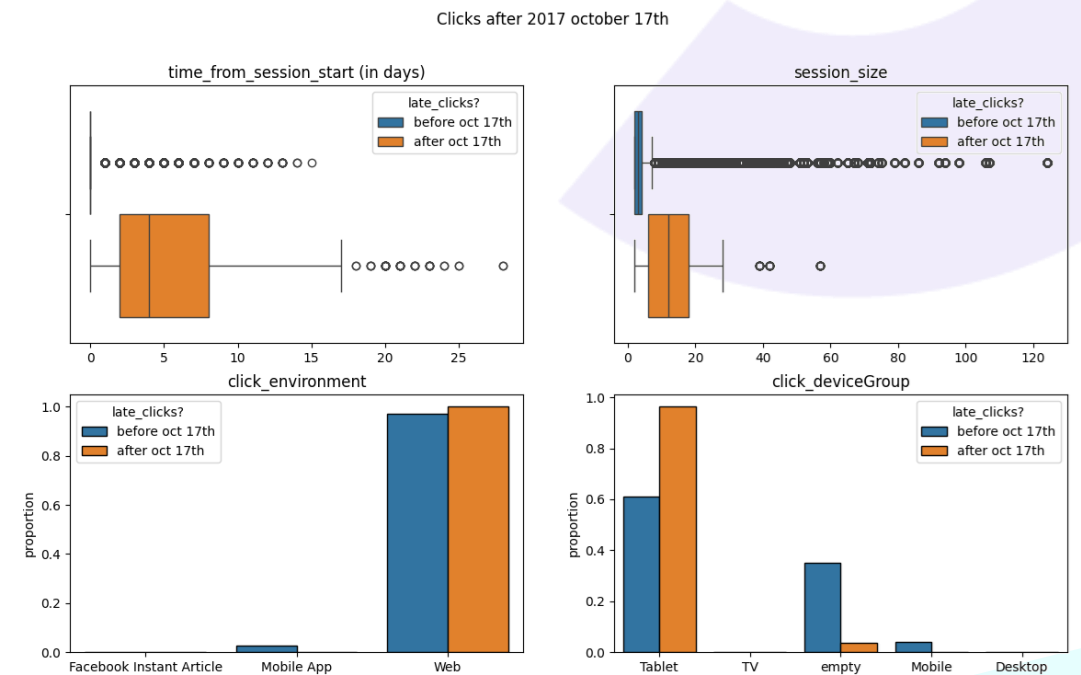
# EDA

'session\_size' - Empirical distribution

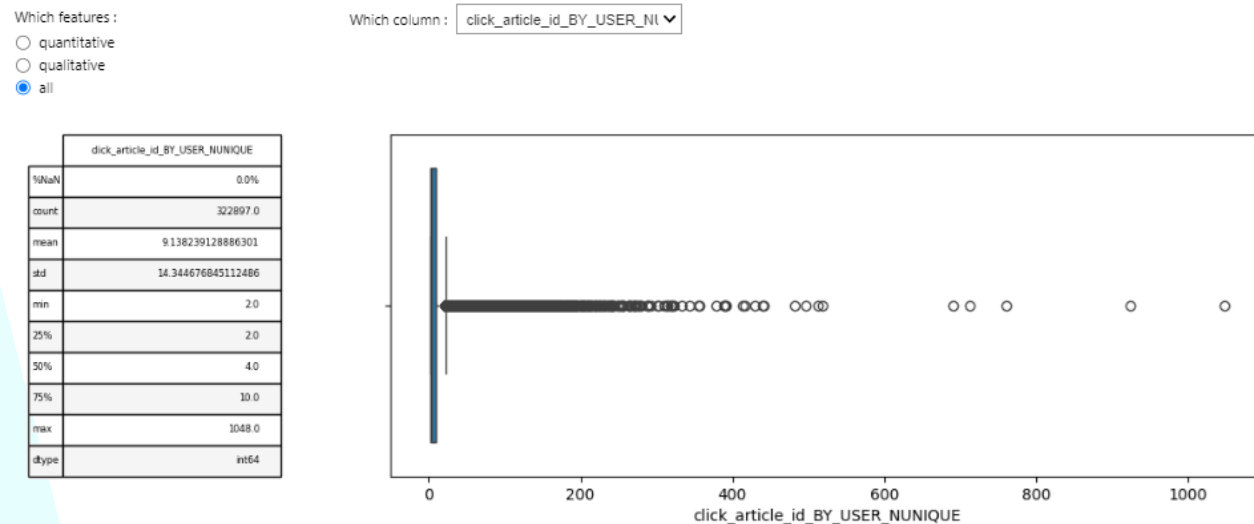


Sessions très courtes

- Sessions démarrent jusqu'au 17 oct 2017
- Ensuite, un bug ? des sessions restées ouverte ?



# EDA

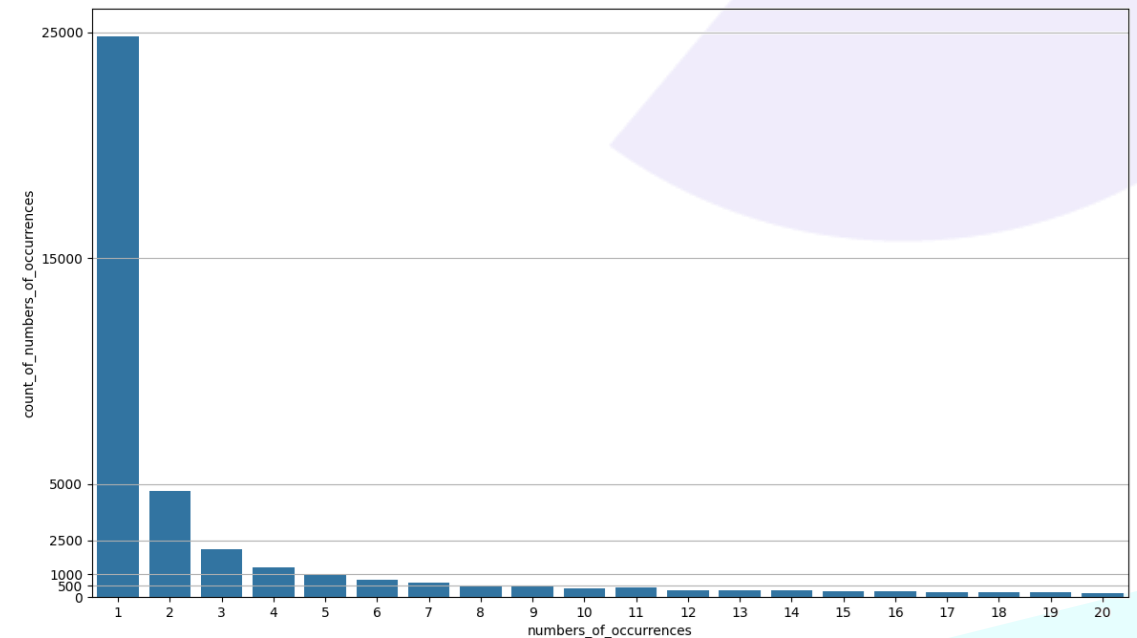


- Grande majorité des utilisateurs ont lu moins de 10 articles
- MAIS certains sont extrêmement actifs
- Prendre garde à cela dans le système de notation, pondérer

- Très grande prédominance des articles avec très peu de vues
- Prendre garde également, filtrer



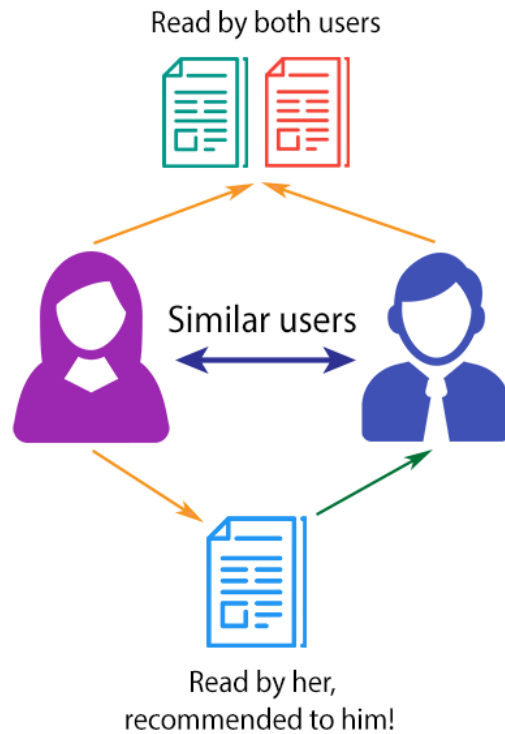
Number of occurrences of articles - Counting



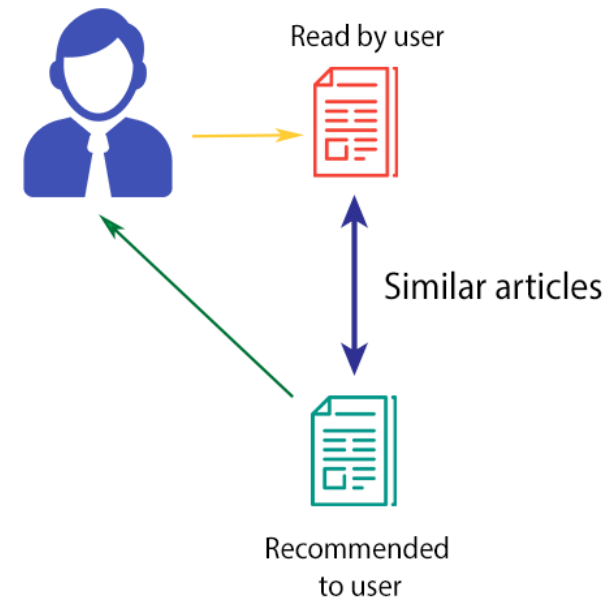
# PARTIE 3 – MODÉLISATION

# Deux approches

## COLLABORATIVE FILTERING



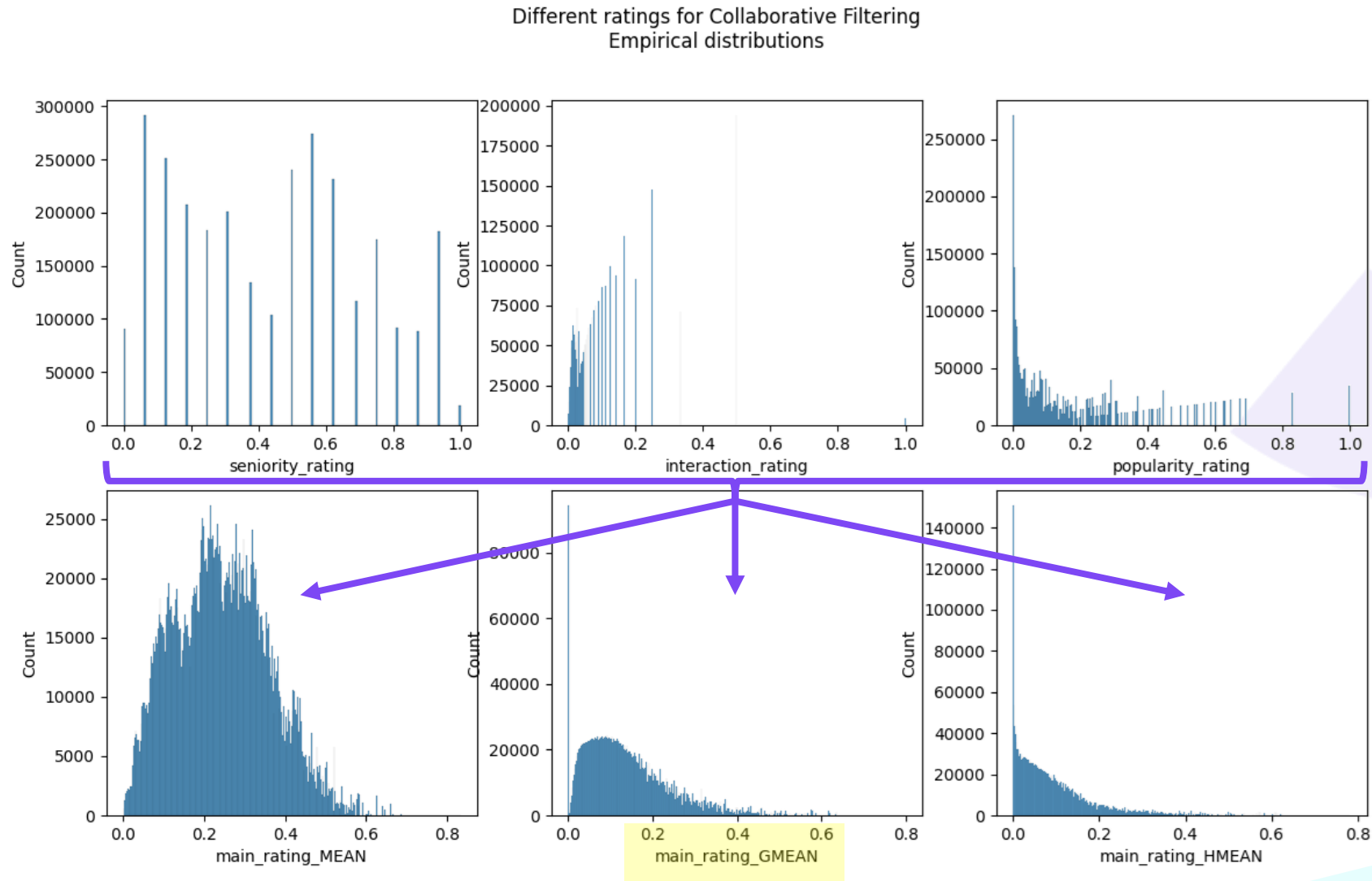
## CONTENT-BASED FILTERING



# Collaborative Filtering

- Utilisation de la librairie **surprise**
- Et de techniques de factorisation : prédire l'avis d'un utilisateur sur un article qu'il n'a jamais noté
- Mais ici : pas de *rating* explicite (pas de note, pas d'étoiles, etc.)
- ➡ créer un *rating* implicite, basé sur les interactions (les *clicks*) :
  - ~~Articles lus moins de 10 fois~~
  - Composante 1 : le fait que l'utilisateur ait lu l'article, normalisé
  - Composante 2 : l'ancienneté du click, normalisée
  - Composante 3 : popularité de l'article (nombre d'occurrences, borné à 1000 pour être ensuite normalisé)
  - Les combiner

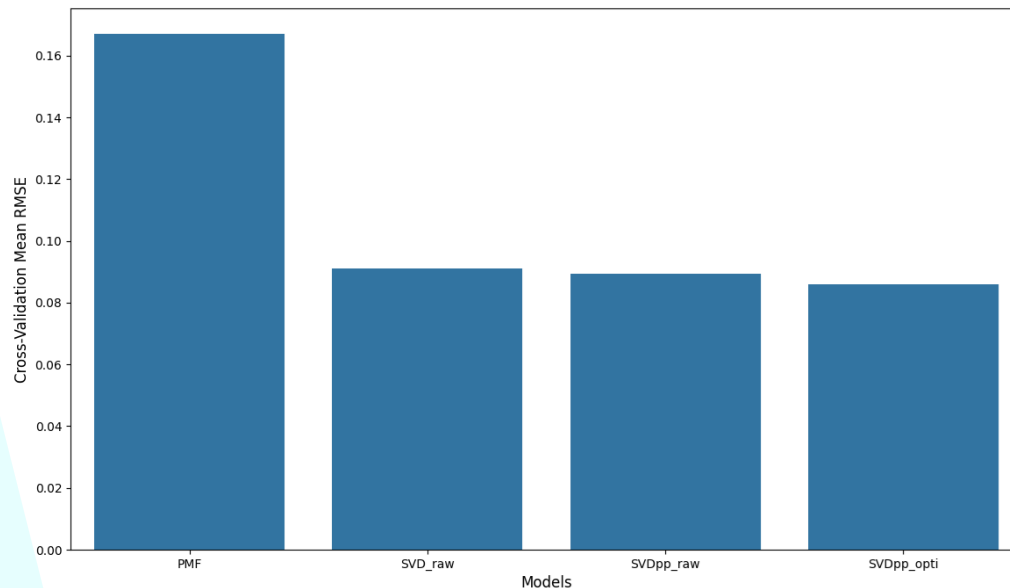
# Collaborative Filtering



# Collaborative Filtering

- 3 modèles testés
- 1 modèle optimisé

Collaborative Filtering : Cross Validation results



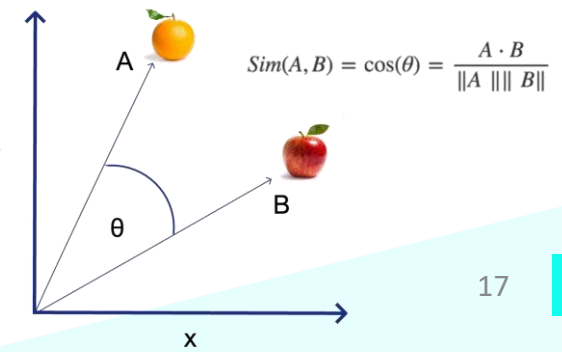
- Faire des recommandations :
  - Entraîner le modèle sur toutes les données
  - Filtrer nos données sur un `user_id`
  - Créer un dataset contenant les articles qu'il n'a **jamais** lu
  - Prédire les ratings associés
  - Garder les 5 meilleurs



# Content Base

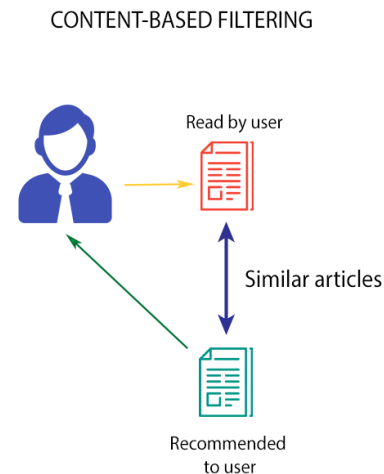
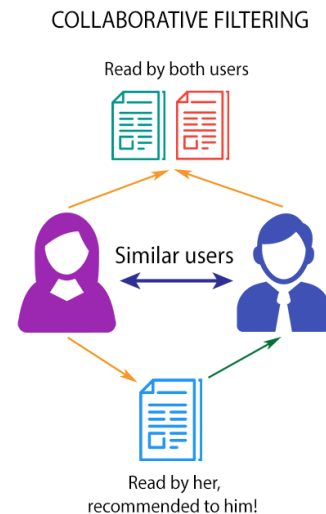
- Essayer d'obtenir les 2 catégories « préférées du moment » de l'utilisateur :
  - Composante 1 : nombre d'articles lus par catégorie, normalisé
  - Composante 2 : récence
  - Combiner les deux
  - Agréger par catégorie
  - Garder les 2 premières catégories et les articles lus dans celles-ci
- Filtrer les *embeddings* sur ces 2 catégories
- Sortir les 5 articles (3 pour la 1<sup>ère</sup> catégorie, 2 pour la 2<sup>ème</sup>) les plus similaires aux articles déjà lus

**Cosine Similarity**



# Des forces et des faiblesses

- + articles parfois très différents des articles déjà lus. Découvrir de nouvelles choses
- + tirer parti des tendances
- difficile de faire des recommandations pertinentes à un nouvel utilisateur
- difficile d'intégrer un nouvel article
- nécessite beaucoup de données



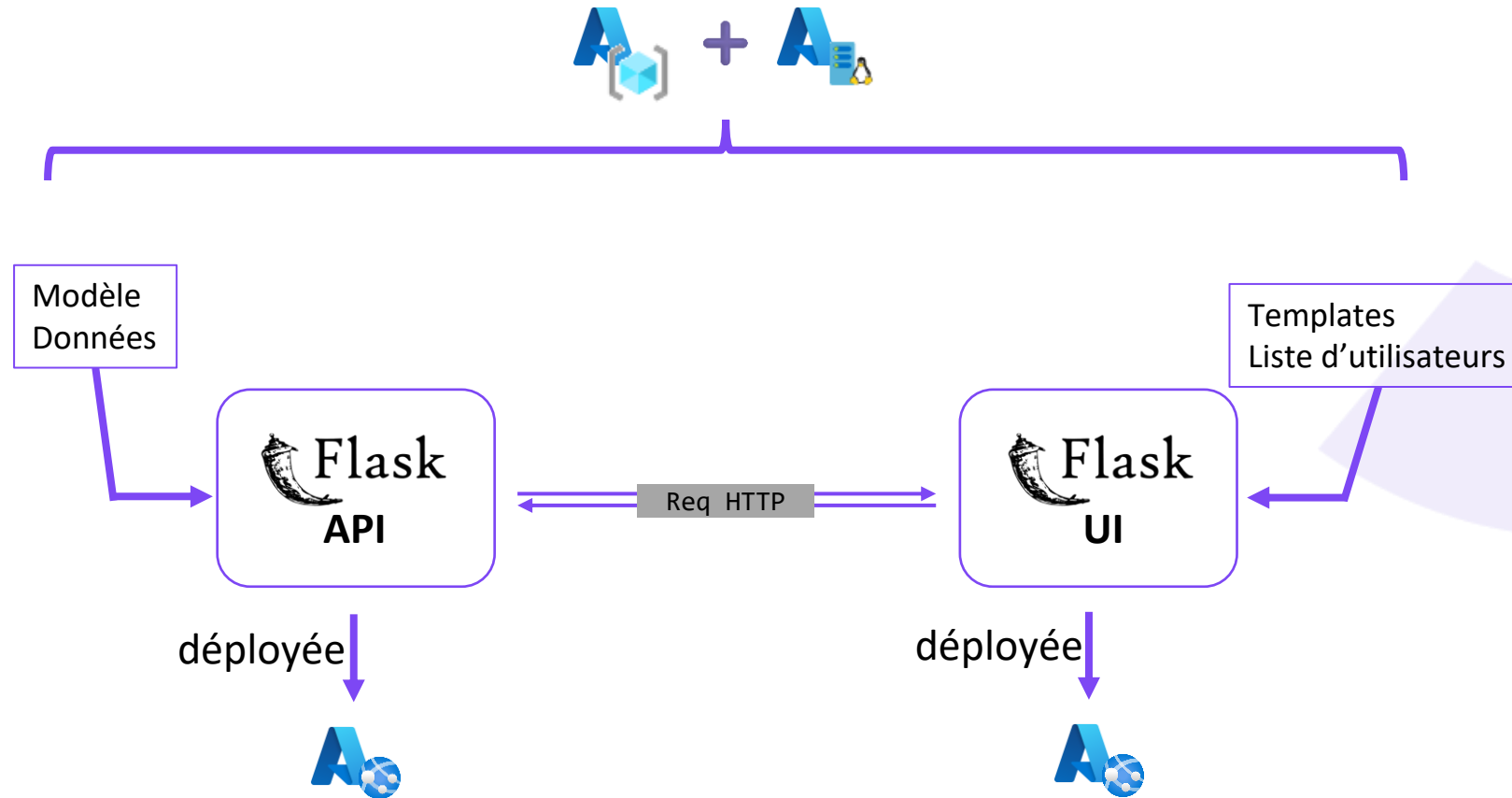
- + fonctionne même avec peu d'utilisateurs
- + permet d'approfondir ses goûts
- « enferme » l'utilisateur, ne lui fait pas découvrir des éléments nouveaux



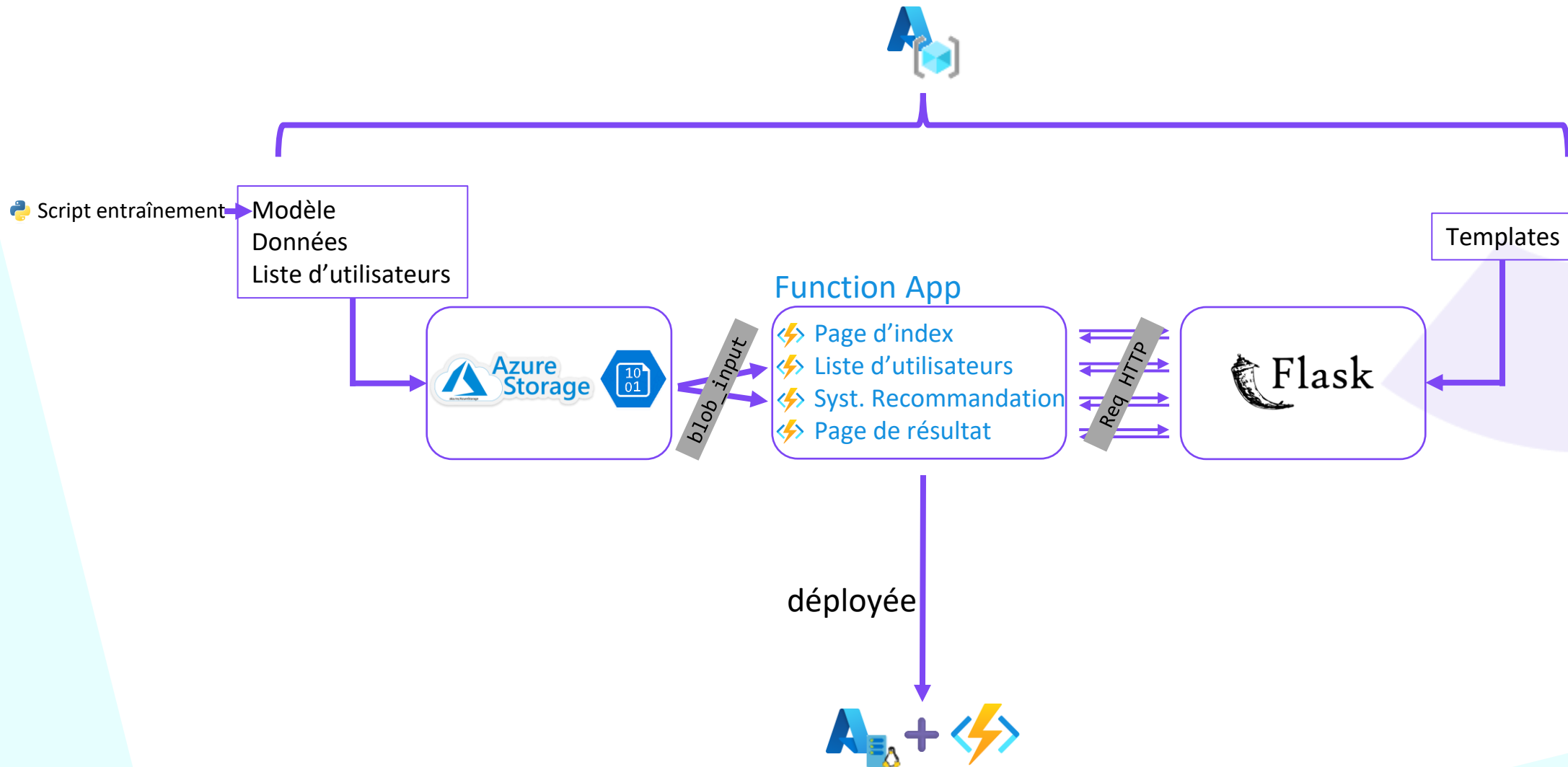
Modèle hybride

# PARTIE 4 – ARCHITECTURE

# Architecture, ce que nous n'avons pas fait...

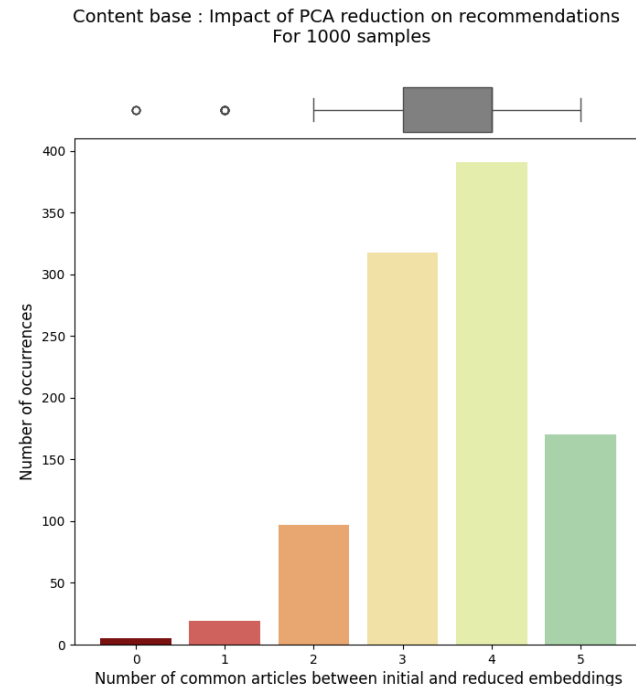


# Architecture






# Déploiement

- Au stade du MVP, **pour la mise en production**, limitation de la **taille** :
  - Des données utilisateurs → Réduction du nombre d'utilisateurs
  - Du modèle *Surprise*
  - Des *embeddings* articles → Analyse en composantes principales



# Déploiement

- classe `recommender_deployer`
- *resource group* 
- *storage account* 
- *function app service plan* 
- *function app* 
- *managed id* 
- *role assignments (user id et managed id)*
- paramètre de connexion
- publier 
- déploiement continu  +  GitHub Actions

# PARTIE 5 – SYSTÈME DE RECOMMANDATION



# Fonctionnement

## ⚡ Liste d'utilisateurs

```
ocp9app.azurewebsites.net/listofids

{"list": [7967, 8613, 12219, 13413, 14202, 15079, 16407, 18577, 21085, 23440, 24719, 25710, 27472, 31202, 47519, 53102, 53398, 53453, 61086, 63013, 67729, 77635, 81983, 82182, 88249, 93183, 98885, 98939, 102315, 104441, 105129, 105610, 108334, 118673, 122850, 125156, 127659, 128751, 129820, 131820, 133200, 136536, 136568, 138478, 139063, 139961, 141545, 145303, 147837, 150904, 151732, 151870, 153716, 159616, 162164, 163110, 166509, 171481, 172159, 175556, 176971, 181099, 181569, 183266, 186796, 189829, 192670, 193317, 196370, 198099, 201967, 205436, 205589, 208681, 210085, 219606, 224200, 225726, 228473, 229205, 230172, 232833, 233874, 236304, 241167, 245089, 251177, 252147, 256027, 262290, 263110, 278966, 283121, 283236, 283498, 293163, 294181, 299460, 308727, 318149]}
```

## ⚡ Page d'index

ocp9app.azurewebsites.net/index

Welcome to My Content !

User top 5 recommendations - Testing

Please, select a user\_id :

and the number of recommendations from Collaborative Filtering :

## ⚡ Syst. Recommendation

```
ocp9app.azurewebsites.net/recsfrommodel/7967/2

{"recs": [226401, 5292, 159563, 161100, 157192]}
```

## ⚡ Page de résultat

result - user#7967 - n\_cf#2

ocp9app.azurewebsites.net/result

Welcome to My Content !

User top 5 recommendations - Testing

Please, select a user\_id :

and the number of recommendations from Collaborative Filtering :

For user #7967, we recommend :

1 : 226401

2 : 5292

3 : 159563

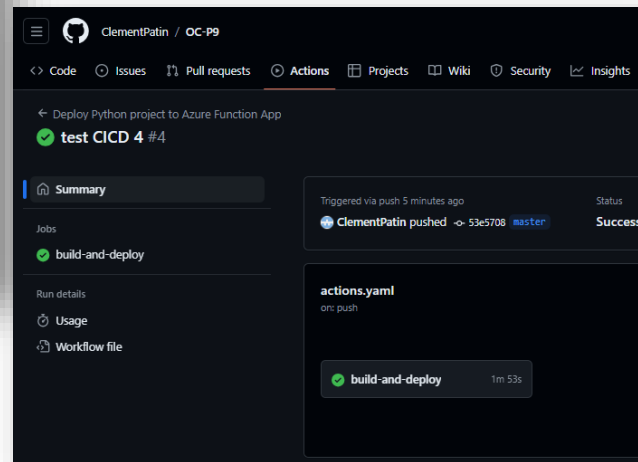
4 : 161100

5 : 157192

(2 from Collaborative Filtering, 3 from Content Base)

# CI/CD GitHub Actions

```
! actions.yaml M X
.github > workflows > ! actions.yaml
1 name: Deploy Python project to Azure Function App
2
3 on:
4   [push]
5
6 env:
7   AZURE_FUNCTIONAPP_NAME: 'ocp9app' # set this to your function app name
8   AZURE_FUNCTIONAPP_PACKAGE_PATH: 'Patin_Clement_1_application_052024'
9   PYTHON_VERSION: '3.11.8' # set this to the python version
10
11 jobs:
12   build-and-deploy:
13     runs-on: ubuntu-latest
14     environment: dev
15     steps:
16       - name: 'Checkout GitHub Action'
17         uses: actions/checkout@v4
18
19       - name: 'Setup Python ${{ env.PYTHON_VERSION }} Environment'
20         uses: actions/setup-python@v4
21         with:
22           python-version: ${{ env.PYTHON_VERSION }}
23
24       - name: 'Run Azure Functions Action'
25         uses: Azure/functions-action@v1
26         id: fa
27         with:
28           app-name: ${{ env.AZURE_FUNCTIONAPP_NAME }}
29           package: ${{ env.AZURE_FUNCTIONAPP_PACKAGE_PATH }}
30           publish-profile: ${{ secrets.AZURE_FUNCTIONAPP_PUBLISH_PROFILE }}
31           scm-do-build-during-deployment: true
32           enable-oryx-build: true
```



ClementPatin / OC-P9

<> Code Issues Pull requests Actions Projects Wiki Security Insights

Deploy Python project to Azure Function App

test CIRD 4 #4

Summary

Jobs

- build-and-deploy

Run details

Usage

Workflow file

Triggered via push 5 minutes ago

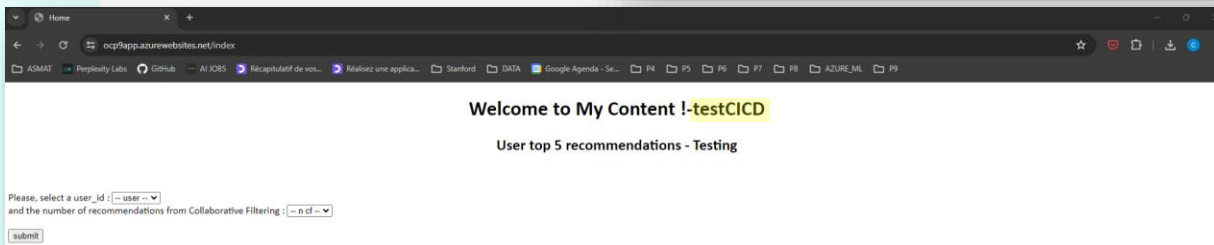
ClementPatin pushed -> 53e5708 master

Status: Success

actions.yaml

on: push

build-and-deploy 1m 53s



Welcome to My Content I-testCIRD

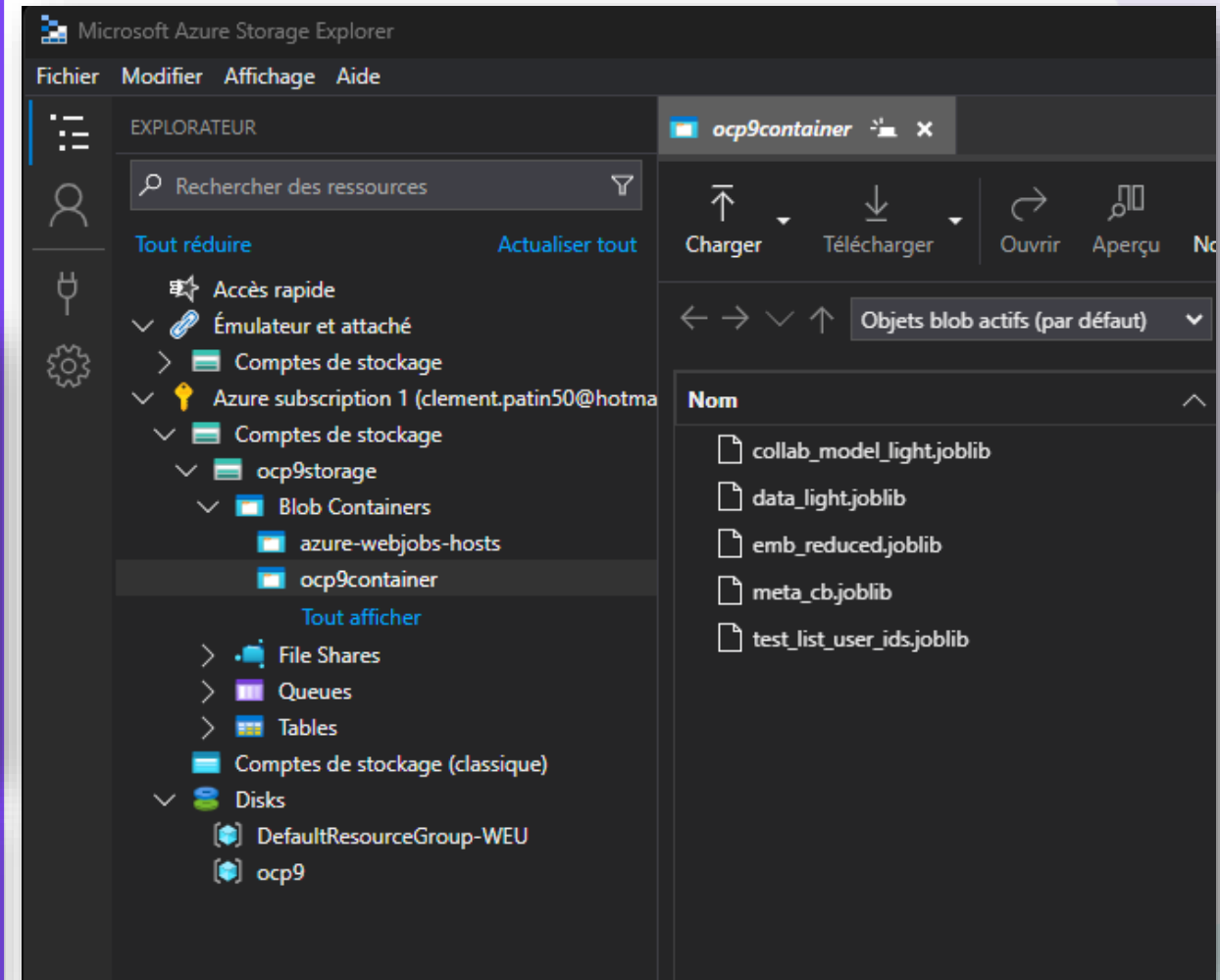
User top 5 recommendations - Testing

Please, select a user\_id: [user --]

and the number of recommendations from Collaborative Filtering: [n d --]

submit

# Blob Storage



Microsoft Azure Storage Explorer

Fichier Modifier Affichage Aide

EXPLORATEUR

Rechercher des ressources

Tout réduire Actualiser tout

Charger Télécharger Ouvrir Aperçu

Objets blob actifs (par défaut)

Nom

- collab\_model\_light.joblib
- data\_light.joblib
- emb\_reduced.joblib
- meta\_cb.joblib
- test\_list\_user\_ids.joblib

Accès rapide

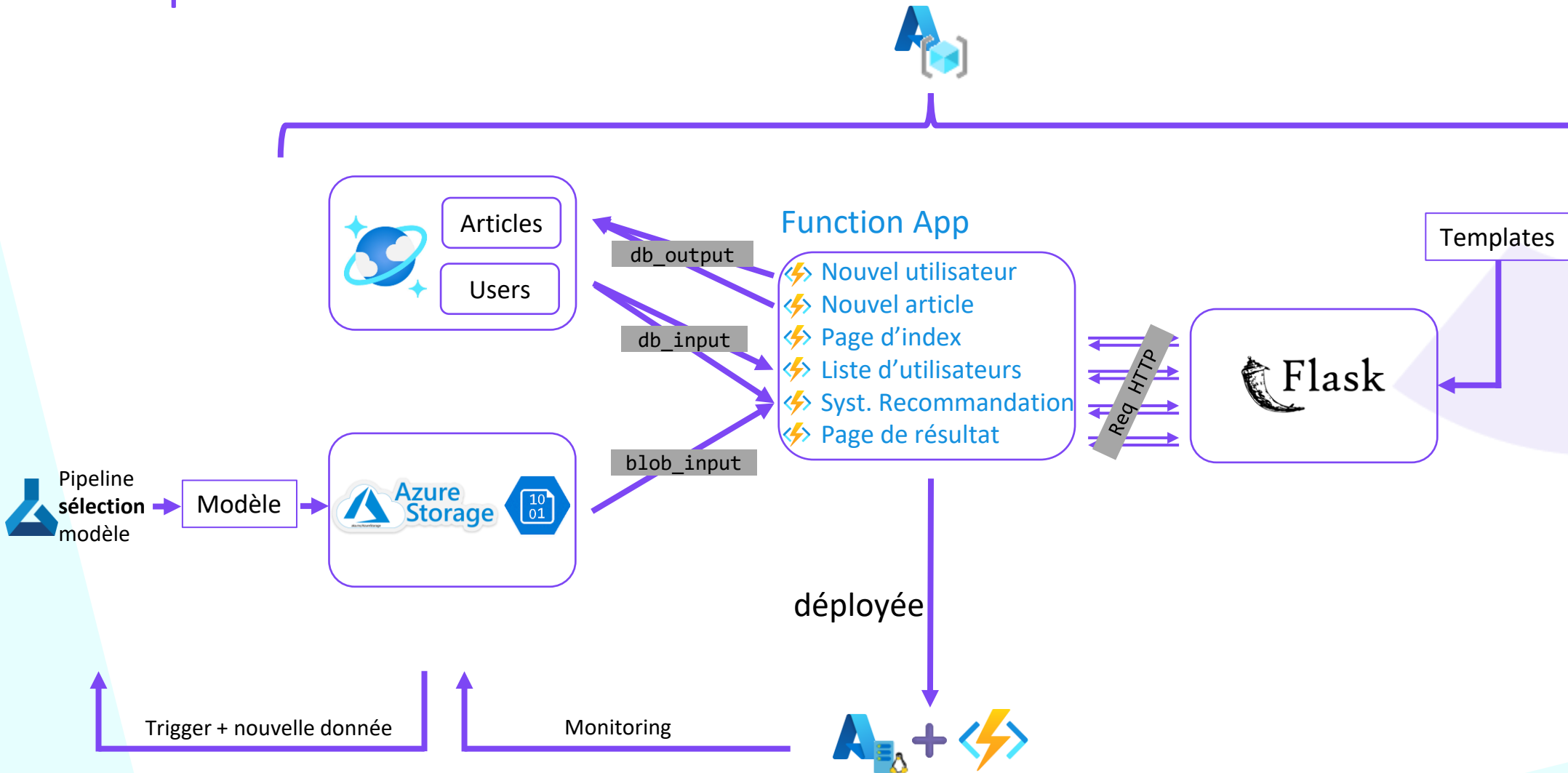
- Émulateur et attaché
- Comptes de stockage
- Azure subscription 1 (clement.patin50@hotmail.com)
- Comptes de stockage
- ocp9storage
- Blob Containers
  - azure-webjobs-hosts
  - ocp9container

Tout afficher

- File Shares
- Queues
- Tables
- Comptes de stockage (classique)
- Disks
  - DefaultResourceGroup-WEU
  - ocp9

# PARTIE 6 – ARCHITECTURE CIBLE

# Architecture cible, quid des nouveaux users ? articles ?



merci