Victoire_Projet2_Big_Data

June 22, 2025

Rapport de Projet : Système de Recommandation d'Images
Date : 22 juin 2025
Auteurs : Kasende Ngeleka Victoire

0.0.1 Introduction

Dans le cadre de ce projet, nous avons conçu et implémenté un système de recommandation d'images automatisé, basé sur les préférences des utilisateurs. L'objectif était de développer un pipeline complet, allant de la collecte des données à la visualisation des recommandations, en passant par l'analyse et la modélisation.

0.0.2 Objectifs du Projet

- **Collecter et traiter** un ensemble d'images et leurs métadonnées.
- **Analyser** les caractéristiques visuelles (couleurs, orientation, tags, etc.).
- **Générer des profils utilisateurs** simulés ou réels.
- **Proposer des recommandations personnalisées** via deux approches :
  - Filtrage basé sur le contenu (similarité avec les préférences de l'utilisateur).
  - Filtrage collaboratif (recommandations basées sur les choix d'utilisateurs similaires).

0.0.3 Données Utilisées

Source des images API **Picsum Photos** (libres de droits, sans attribution requise).

Caractéristiques des données

- **100 images JPEG** (800×600 px).
- **Métadonnées** (fichier JSON) :
  - Nom, dimensions, format, orientation.
  - Couleurs dominantes (K-Means), tags générés automatiquement.

Profils utilisateurs

- Préférences stockées : couleurs, orientation, tags, format.
- Génération aléatoire ou basée sur des images “likées”.

0.0.4 Méthodologie

a. Extraction des caractéristiques

- **Clustering des couleurs** (K-Means via scikit-learn).
- **Génération de tags** à partir des métadonnées.

b. Système de recommandation

1. **Content-Based Filtering** :
   - Score de similarité entre profil utilisateur et images.
2. **Collaborative Filtering** :
   - Similarité cosinus entre utilisateurs.
   - Recommandation basée sur les voisins proches.

c. Validation

- Tests unitaires.
- Visualisation des recommandations (matplotlib).

0.0.5 Résultats & Auto-Évaluation

Points forts Notre système de recommandation présente plusieurs atouts majeurs :

- Pipeline entièrement automatisé Nous avons mis en place un processus complet et fluide, depuis la collecte des images jusqu'à la visualisation des résultats, éliminant toute intervention manuelle intermédiaire.
- Architecture modulaire et robuste Le code s'appuie sur des bibliothèques Python éprouvées (scikit-learn pour le machine learning, PIL pour le traitement d'images, matplotlib pour la visualisation), garantissant à la fois stabilité et facilité de maintenance.
- Approche hybride innovante La combinaison du filtrage basé contenu (analyse des caractéristiques visuelles) et du filtrage collaboratif (similarité entre utilisateurs) permet d'offrir des recommandations plus pertinentes et diversifiées.

Limites Malgré les résultats satisfaisants, notre système présente certaines contraintes liées à son cadre expérimental :

- Absence de données utilisateurs réelles : Les profils étant générés de manière aléatoire, les recommandations n'ont pas pu être validées avec des comportements utilisateurs authentiques. Une mise en production nécessiterait une phase de collecte de données réelles pour affiner le modèle.

- Métadonnées incomplètes ou simulées : Certaines informations (comme les EXIF ou les tags) ont été partiellement synthétisées par souci de simplification, ce qui pourrait limiter la granularité des recommandations dans un contexte réel.

0.0.6 Améliorations Possibles

Pour faire évoluer notre système vers une solution professionnelle, plusieurs pistes d'optimisation s'offrent à nous :

- Développement d'une interface interactive L'implémentation d'une interface web ou mobile intuitive permettrait une expérience utilisateur plus immersive et faciliterait la collecte de données comportementales réelles.
- Enrichissement des capacités d'analyse L'intégration de modèles de deep learning (comme VGG ou ResNet) pourrait considérablement améliorer la reconnaissance des caractéristiques visuelles complexes, au-delà des simples métadonnées.
- Mécanisme d'adaptation continue La mise en place d'un système de feedback en temps réel permettrait au modèle d'affiner ses recommandations en fonction des interactions utilisateurs, créant ainsi une boucle d'amélioration continue.
- Diversification des sources d'images L'ajout de plateformes comme Unsplash ou Flickr élargirait la variété des contenus disponibles tout en fournissant des métadonnées plus riches et authentiques.

0.0.7 Conclusion

Ce projet a permis de maîtriser un pipeline complet de recommandation d'images. Bien que limité par des données simulées, l'architecture pourrait être déployée dans un cadre réel avec des utilisateurs finaux.

0.0.8 References

[1] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011. [Pour l'implémentation de scikit-learn/K-Means]

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems (NIPS), vol. 25, 2012. [Pour les approches de deep learning VGG/ResNet]

[3] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan. 2003. [Pour le filtrage collaboratif]

[4] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," Journal of the Royal Statistical Society, vol. 28, no. 1, pp. 100-108, 1979. [Pour le K-Means color clustering]

[5] Picsum Photos, "Terms of Service," 2023. [Online]. Available: https://picsum.photos/ [Pour la source des données d'images]

[6] Python Imaging Library (PIL) Developers, "Pillow Documentation," 2023. [Online]. Available: https://pillow.readthedocs.io/ [Pour le traitement d'images]

[7] J. D. Hunter, “Matplotlib: A 2D Graphics Environment,” Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, May 2007. [Pour la visualisation]

Annexes

- Dossier `images/`
- Fichiers JSON : métadonnées, profils, historique.
- Graphes d’analyse (`graphes/`).
- lien vers le dépôt GitHub du projet : GitHub Repository