Zaghouane Farès
Galmier Romain

Data Mining

Table des matières

Présentation du projet

Proposer des fleurs à différents utilisateurs en fonction de leurs préférences. Le but est de faire découvrir de nouvelles espèces notamment grâce aux couleurs mais aussi aux tranches d'âge et au sexe.

Dataset

Notre dataset est ***104 Flowers : Garden of Eden*** tiré du site Kaggle. C'est un ensemble de différentes espèces de fleurs tirées de plusieurs data bases publiques. Il comprend plusieurs dossiers. Les images sont disponibles en plusieurs tailles. Toutes les images sont au format jpeg.

Notre base de données est conséquente avec un peu plus de 95 000 images pour environ 2Gb mais nous avons grandement réduit ce nombre trop important. Dans le script nous avons gardé uniquement 5000 images pour des raisons évidentes.
Il y a un grand nombre d'espèces de fleur pour plus de diversité.

Les données stockées

La couleur est la donnée principale que nous avons décidé de récupérer. Dans un fichier JSON (data.json) nous avons stocké pour chaque image dans l'ordre :

- La taille
- L'espèce de fleur
- Le nom du fichier
- Les 3 couleurs principales de l'image

Pour les couleurs nous utilisons une base de données (colorsPalette.tsv) qui nous permet d'avoir une base de 140 couleurs différentes et pas 255^3.
Les informations de nom, taille, et espèce vont permettre de connaitre le PATH de chaque image.

Etant une base de données avec uniquement des images il n'y a aucune classification ou label pré-fait pour les images tout à donc été fait automatiquement par un programme de notre composition.

Nous stockons également toutes les préférences des utilisateur (users :Id/name, age, sexe, Couleur Préféré) dans un fichier Json (user.json).

Pour finir nous allons aussi stocker les choix des users, encore une fois, dans un Json (linkTable.json). Plus concrètement pour chaque choix fait par un user on va conserver le fait qu'il aime ou pas l'image.
Chaque ligne de cette table de correspondance sera de la forme suivante : L'id de l'user, l'id de l'image jugée et si l'user à aimer ou non l'image.

Toutes ces données vont permettre de pouvoir récupérer/charger ces "archives" plutôt que de les recrées à chaque fois.

Ainsi nous pourrons avoir une continuité a chaque exécution du programme, nous ne perdrons pas nos utilisateurs à chaque redémarrage

Les préférences utilisateurs

Pour les informations de l'utilisateur, on demande l'âge, le sexe et la couleur préféré de l'utilisateur afin de prédire au mieux les images qui peuvent correspondre aux goûts de l'utilisateur.

Pour chaque nouvel utilisateur on lui propose un petit panel d'image qui plaise aux personnes de la même catégorie qu'eux. Par exemple si un homme de 20 ans a déjà ses préférences enregistrées et qu'un nouvel utilisateur masculin de 25 ans arrive pour également découvrir de nouvelles espèces, on pourra lui proposer des fleurs plus appropriées, en piochant parmi les images que l'utilisateur déjà enregistré a bien aimé.

Les préférences sont faites en fonction d'une couleur principale et de deux couleurs secondaires

Les modèles d'exploration de données et/ou d'apprentissage machine que nous avons utilisée.

Pour déterminer les couleurs prédominantes de chaque image de fleurs nous avons utilisé la méthode des K-means. Grâce à cette méthode cela, nous avons pu extraire les trois couleurs prédominantes.

Ce modèle d'exploration de données est très indiqué dans notre situation car nous utilisons les couleurs principales et secondaires en tant qu'informations pour les préférences de l'utilisateur.

Ce modèle peut même nous permettre d'augmenter le nombre de couleurs secondaires dans le cas ou des couleur pas importantes (dans notre cas le noir des ombres par exemple) se retrouverais comme principale. C'est en parti pour cela que nous avons décidé de prendre les 3 couleurs principales, car les images n'étant pas parfaite d'un point de vue traitement de donnée, nous trouvons énormément de noir des ombres, mais également du vert, ce qui ne nous permet pas de venir différencier toutes ces fleurs. Avec les 3 couleurs principales, nous nous assurons que nous extrayons les couleurs spécifiques à chaque fleur.

Grace à ce modèle on utilise ensuite un classificateur de type Decision Tree de la bibliothèque Sklearn, grâce aux données utilisateur on entraine le Decision tree. Ensuite on prédit les images avec le classificateur pour proposer des images en rapport. Le Decition tree cherche les corrélations en les images. On base le Decision tree sur plusieurs critères comme la taille de l'image, l'espèce de plante, les trois couleurs principales de l'mage

Pour les nouveaux utilisateurs, nous avons simplement réaliser un Dataframe des utilisateurs existants, et somme venue cherches les images aimer pas des utilisateurs du même sexe et âge. Nous voulions dans un premier temps, utiliser une méthode de « collaborative filtering », mais cela n'était pas pertinent compte tenu que nous avions majoritairement des faux utilisateurs, générés aléatoirement, donc nous n'aurions pas pu extraire des données pertinentes pour faire la suggestion. Mais dans un model réel, cela serait plus pertinent

Auto-évaluation de votre travail.

Le programme de notre projet est bien construit et peut s'adapter et/ou évoluer facilement grâce à l'utilisation de classe.

Notre sujet en revanche n'est peut-être pas adapté pour une analyse de résultats de prédictions sur un petit nombre de personnes, en effet la prédiction consiste uniquement à dire si on a aimé ou pas l'image et à catégoriser les goûts des gens en fonction de leurs caractéristiques. Or le Decision tree a besoin de plus d'utilisation par de vrai utilisateurs. Il faut donc une grande variété de personne qui utilise notre modèle pour que celui-ci soit vraiment pertinent et efficace.

On a donc créé des faux utilisateurs pour tester notre programme et obtenir des métriques mais étant donné qu'on utilise des faux utilisateurs qui répondent aléatoirement, le modèle de prédiction n'a aucun sens et donc il était prévisible que les résultats aillaient être mauvais (précision 33% et grande variance). Le f1 score n'est pas pertinent étant donné que les faux utilisateurs font des choix aléatoires.

On est aussi conscient que cela aurait été mieux d'utiliser des filtres collaboratifs pour récupérer les préférences utilisateur pour déjà proposer des images pertinentes mais ce genre de model est pertinent que si ce sont de vrais utilisateurs ce qui n'est pas le cas ici.

Remarques concernant les séances pratiques, les exercices et les possibilités d'amélioration

Il faut peut-être plus de séances actives c'est-à-dire avec un peut plus de code à écrire avec l'aide de professeurs.
Mais le système de compréhension de code est très bien. Il manque aussi un petit glossaire qui résume les fonctions des bibliothèques (même si tout est disponible sur internet)
Également il aurait été appréciable d'avoir plus de séances de projet dans le but d'avoir plus d'accompagnement des professeurs au commencement du projet.

Conclusion

Ce projet a été l'oppurtunité pour nous d'appliquer dans, une situation pratique, tous les examples et les algoritmes mis en place en tp .

Nous avons pu rassembler les donnees d'une base de données et les utiliser pour connaitre les gouts de nos utilisateurs et de nouvelles information sur le long terme .

Les graphiques obtenues sont a étoffé sur le long terme mais donne des informations utiles sur les gouts des utilisateurs.

Pour conclure nous avons besoin de beaucoup plus de données utilisateur pour que notre projet soient réellement aboutit.