CHARRAT Philippe 30/03/2021

CORNU Clément

**Data Mining**

**Rapport de projet**

# But du projet

Dans le cadre du module de Data Mining, nous nous sommes vu attribuer un projet dont le but était la mise en place d’un système de recommandation d’images en Python.

L’objectif était de nous faire manipuler les différents concepts étudiés en cours, en nous mettant dans une situation donnée. Nous devions donc faire en sorte que les recommandations d’images soit faites en fonction de l’utilisateur, en manipulant les données et en utilisant des modèles de traitement afin d’obtenir le résultat final.

# Source des données des images et de leur licence

Nous avons choisi pour ce projet d’utiliser des images issues de Wikidata. Pour cette raison, les images que nous utilisons sont libres de droit.

Ce choix d’utilisation implique cependant plusieurs contraintes : nous devons utiliser la bibliothèque Wikidata *qwikidata.sparql* et plus spécifiquement la *fonction return\_sparql\_query\_results* afin de pouvoir exécuter des requêtes vers l’API de Wikidata afin de récupérer les images. Par ailleurs, les informations (Exif Tags) des images étant hétérogènes, cela nous obligera à un post-traitement après requête Wikidata afin ne garder les images pertinentes dans la réalisation de notre projet.

Nous avons par ailleurs choisi de cibler plusieurs grands thèmes Wikidata afin de garantir l’existence d’images. Ces thèmes sont : chat, montagne, plante, peinture (**AJOUTER LES AUTRES)**.

# Taille des données

Les données téléchargées depuis Wikidata nous permettent d’avoir des images d’environ 2Mo. Cela nous permet un temps de téléchargement d’environ 10 minutes pour une centaine d’images et leurs informations. Les images sont alors de bonne qualité, ce qui rend possible leur exploitation pour déterminer des couleurs prédominentes par exemple.

Par ailleurs, les différents scripts Python dévéloppés ont des tailles n’excédant pas les 5 Ko : leur taille n’a donc rien à voir avec celles des images.

# Informations stockées pour chaque image

Nous avons choisi tout d’abord de nous concentrer sur les informations contenues directement dans l’image : les Exif Tags. Cependant, comme mentionné dans la deuxième partie, tous les Exif Tags ne sont pas présents sur toutes les images, et leur répartition est assez hétérogène : certaines images en comporteront énormément, et d’autres presque aucun.

Pour notre part, nous avons donc choisi de nous focaliser sur certains, qui nous paraissaient pertinents pour notre projet :

* 270 : la description de l’image (et donc son titre, entres autres)
* 305 : le logiciel de traitement de l’image
* 306 : date et heure de la prise de vue
* 40962 : la largeur en pixel de l’image
* 40963 : la hauteur en pixels de l’image

Par ailleurs, nous avons choisi de rajouter « à la main » certains tags sur l’image. Nous avons développé un script nous permettant de déterminer la couleur prédominante de l’image, et rajouté ce tag sur l’image. Nous avons développé un second script dont le but était de déterminer le type d’images : une icone, une grande ou une très grande image. De plus, l’orientation, le nom et le thème de l’image ont été ajoutés pour chaque image.

# Informations conernant les préférences utilisateurs

Les informations stockées à propos de l’utilisateur ont vocation à être complémentaires à celles stockées pour les images.

Ainsi, afin d’obtenir tout d’abord les préférences de l’utilisateur, nous avons choisi d’adopter une première phase de création de profil, qui permet d’obtenir les premières images likées par l’utilisateur et donc une base de travail pour les propositions d’images futures.

Les noms de ces différentes images sont stockées dans un fichier profil\_*username*.json, qui contient donc les préférences de l’utilisateur.