BOISDON Hugues

DERVILLE Thomas

Rapport Projet Datamining

**But du projet** :

L’objectif du projet était de créer un système de recommandation de monuments historiques. A partir d’une base de données composés d’une centaine de monuments et à l’aide d’une application récupérant des images « likées » par l’utilisateur, le système a pour but de recommander d’autres monuments qui correspondent à ses préférences.

**Source des données et licences:**

Les données des monuments proviennent toutes de Wikidata et ont été obtenues à l’aide du service de requête intégré. Toutes les images sont licenciées sous le CC BY-SA 4.0.

Les données concernant le style architectural et le pays ont été obtenues via l’API WikiBaseIntegrator qui permet de récupérer les données liées à un identifiant WikiData sans avoir à repasser par une requête SPARQL.

**Taille des données :**

Les données sont stockées sous deux formats différents : un fichier data au format JSON faisant environ 20 Ko et un dossier images contenant les images liées aux monuments de la base de données, dossier faisant environ 60 Mo.

**Informations stockées pour chaque image :**

label : Contient le nom (en anglais afin d’avoir le moins de valeurs inconnues) du monument.

link : Contient le lien wikimedia vers l’image principale du monument (image qui apparaît sur la page wikipédia du monument en général)

entityID : Contient l’identifiant WikiData du monument. On l’utilise pour les requêtes WikiBaseIntegrator.

format : Contient le format de l’image principale (jpeg, png,…)

path : Contient le chemin relatif vers l’image contenu dans le dossier image stocké localement.

size : Contient la taille de l’image sous forme d’une liste à deux membres sous le format [ X, Y ].

sizeCategory :

tags : Contient une liste possédant les différents tags liés au monument entre autres le siècle, le pays et le/les styles architecturaux liées au monument. Principalement utilisé dans le système de recommandation afin de différencier les différents monuments.

date : Contient la date de construction du monument. On la récupère directement depuis la requête SPARQL afin de réaliser une première filtration en ne récupérant que des monuments possédant une date.

style : Contient le/les qualifier(s) WikiData des styles architecturaux possédés par le monument. Pour la même raison que pour date on récupère ces informations directement depuis la requête.

orientation :

artist : Contient le nom de l’auteur de la photo.

dateTaken : Contient la date à laquelle la photo a été prise.

appareil : Contient le nom de l’appareil avec lequel la photo a été prise.

dColors :

dProportions :

**Informations concernant les préférences de l’utilisateur :**

**Modèles d’exploration de données :**

**Apprentissage machine :**

**Auto-évaluation :**

Annotation des données : Du fait de la requête SPARQL initiale, on a déjà un premier filtre qui permet de « sauter » quelques vérifications qui seraient normalement nécessaires comme le type des données reçues (int, str, …) avant de les utiliser dans les diverses fonctions du système. Un exemple notable est un monument n’ayant pas de date de construction précise donc il possède bien la propriété P571 mais la valeur liée est inconnue. Le monument passe alors le filtre indirect de la requête SPARQL ce qui induit des erreurs dans le programme sans les vérifications sur la présence d’une clé « year » dans le JSON contenant les données récupérées.

De plus on se permet de