**1-Informations que nous avons décidé de stocker pour chaque image.**

Nous avons récupéré les métadonnées des images et nous les avons stockés dans un fichier JSON.

Nous avons téléchargé un fichier HTML avec les URL des images de 4 thèmes différents . le thème des 30 premières est montagne , les 30 suivantes ont pour thème Ville , les 20 suivantes sont des rivières et les 20 dernières sont des fruits .Dans un premier temps nous avons attribué à chaque image un tag selon son thème avec une boucle et nous avons stocké ces tags dans un dictionnaire qui a pour clé le nom de l'image et les tags comme valeur. Ensuite nous avons décidé de stocker la couleur dominante des images , nous avons calculé la valeur RGB de la couleur dominante avec la méthode K-means puis nous l'avons converti en couleur python en utilisant la bibliothèque Webcolors.Les couleurs dominantes des images sont aussi stockés dans une liste data . Ensuite on demande à l'utilisateur de donner un #hashtag pour les images qu'il a choisi , nous avons stocké les hashtag donnés par l'utilisateur .Nous avons stockés également la taille des images . Ainsi nous avons mis ces données qu'on a récupéré dans une liste data qui contient : ( couleur dominante, tag,hashtag,taille)

**2-Informations concernant les préférences de l'utilisateur**

Nous avons créé des une liste des utilisateurs,on demande à chaque utilisateur de choisir aléatoirement des images et ensuite on récupères les données des images choisies par l'utilisateur ( couleur prédominante , tag , taille, hashtag) .Ensuite, on affiche chaque image et on demande à l'utilisateur d'indique s'il aime l'image proposée ou pas, on lui demande de saisir oui ou non et on stocke par la suite ses réponses dans une liste result qui peut contenir : Favorite ou NotFavorite. Si la valeur de la liste result est Favorite on stocke les données de l'image en question .Ensuite , on mets en place un classificateur d'arbre de décision pour prédire les images que l'utilisateur peut aimer et lui proposer ces images .

**3-Les modèles d'exploration de données et/ou d'apprentissage machine que vous avez utilisés avec les métriques obtenues.**

Notre procédure utilisée dans l'exploration des données est la suivante:

**Extraction des données:**

le but de cette partie est d'extraire des informations comme la taille de l'image, le format de l'image (.jpeg, .png, etc.), l'orientation de l'image (paysage, portrait, carré, etc.), date de création, modèle d'appareil photo, etc. dans plusieurs fichiers JSON. Dans cette étape nous nous occupons aussi de la phase de nettoyage des données car nous n'avons pas toutes les données pour chacune des images.

**Regroupement des données:**

Nous avons utilisé le regroupement avec l'algorithme de regroupement K-means afin de trouver la couleur prédominante des images sélectionnées

**Filtrage:**

Pour chaque utilisateur nous avons stocker dans une liste les données des images que l'utilisateur a aimé ( c'est à dire choix=='Favorite) pour construire le profil personnel des utilisateurs.

**Classification:**

Pour la classification nous avons utilisé l'arbre de décision .Cet outil d’aide à la décision ou d’exploration de données permet de représenter un ensemble de choix sous la forme graphique d’un arbre. C’est une des méthodes d’apprentissage supervisé les plus populaires pour les problèmes de classification de données . Dans notre contexte nous avons réalisé une méthode pour prédire des images que l'utilisateur peut aimer et lui recommander ces images par la suite.

**4-l'auto-évaluation de votre travail.**

liste Data est la liste qui contient les données des images sélectionnées par l'utilisateur et on stocke dans une autre liste result[] son avis ( Favorite ou NotFavorite)

liste Data=[['large', 'montagne', 'slategray', 'portrait'], ['large', 'montagne', 'darkgray', 'landscape'], ['large', 'ville', 'darkgray', 'landscape'], ['large', 'fruit', 'palegoldenrod', 'landscape'], ['medium', 'chat', 'dimgray', 'landscape'], ['medium', 'chat', 'rosybrown', 'landscape']]

Cet utilisateur a aimé que des images avec des chats :

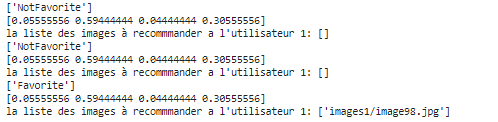


On envoie une liste de 4 images à notre système de prédiction pour prédire le choix de l'utilisateur, on récupère les données de ces images et on les stocke dans une liste :

[['large', 'montagne', 'dimgray', 'landscape'], ['medium', 'chat', 'dimgray', 'portrait'],

['large', 'montagne', 'gray', 'landscape'],['medium', 'chat', 'dimgray', 'landscape']]

Ensuite on envoie la liste des images à recommander dans une liste:



Dans ce test la liste renvoyée contient une seule image : images1/image98.jpg, et il s'agit bien d'une image d'un chat:

