Rapport projet Machine Learning

Josse DE OLIVERA & Léo BARBIER

L'objectif de ce projet est de recommander des images en fonction des préférences de l'utilisateur grâce à un système de recommandation en python. Les tâches liées à l'acquisition, l'annotation, l'analyse et la visualisation des données sont automatisées.

Nous avons décidé de réaliser ce projet avec des images de casinos.

Le projet se déroule avec plusieurs étapes :

1. Collecte de données
2. Étiquetage et annotation
3. Analyses de données
4. Visualisation des données
5. Système de recommandation
6. Tests

1. **Collecte de données :**

Nous récupérons les images de casinos sur Wikidata, nous avons choisi de récupérer les 128 premières images sur les 424 disponibles pour réduire la taille de l'échantillon et réduire la taille de stockage requise pour les enregistrer.

Cela représente environ 420 Mo de données pour les photos des 128 casinos.

Ces images sont censées être libres de droits car sur les plateformes de Wikimedia dont Wikidata fait partie, les utilisateurs de ces services doivent ne mettre en ligne que des images libres de droits. Elles sont donc normalement sous la Creative Commons CC0 License.

Nous récupérons ensuite les métadonnées de chaque image en utilisant les informations Exif. Elles sont introduites dans un fichier JSON global par la suite dans la partie *Étiquetage et annotation*.

2. **Étiquetage et annotation :**

En plus des informations Exif des images nous avons décidé d'enregistrer les 2 couleurs prédominantes automatiquement en analysant l'image. Nous prenons aussi des tags qui sont entrés par nous via un script annexe (en dehors du Jupyter) qui affiche l'image et attend l'entrée de l'utilisateur et stocke dans un fichier par image les tags (se référer à l'image ci-dessous).

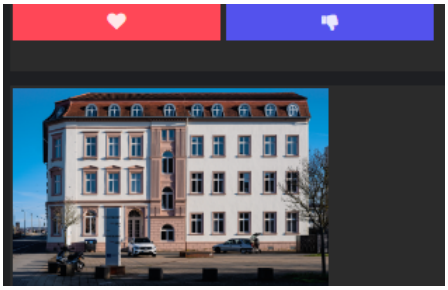
Les informations Exif, les couleurs prédominantes et les tags forment ensemble un fichier JSON qui rassemble toutes les images sous la forme:

```
"nom_image.png": { "exif": {...} , "color": [ ... ], "tags": [ ... ] },
```

Pour les 128 images nous avons donc un fichier JSON de 170Ko contenant les informations qui vont nous permettre d'analyser les données et qui vont nous permettre de recommander des images aux utilisateurs en fonction de leurs préférences.

3. **Analyses de données :**

Pour analyser les préférences des utilisateurs, nous affichons une image avec un bouton like et un bouton dislike.



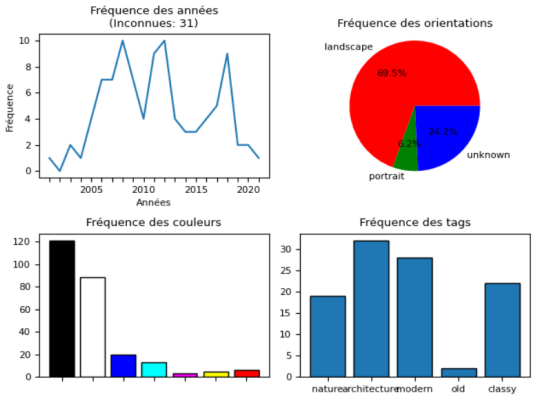
Nous demandons à l'utilisateur son avis sur 10 images pour avoir assez d'informations pour pouvoir lui proposer d'autres images de casinos qui logiquement devrait aimer.

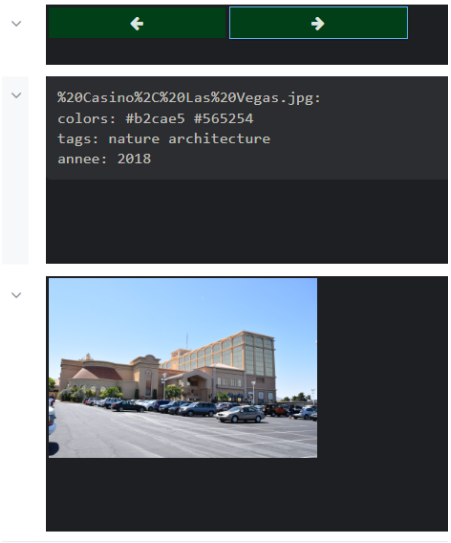
On transforme les données qui sont stockées dans le JSON pour qu'elle soit traitable plus facilement, on les met donc dans un vecteur de données.

Grâce au choix de l'utilisateur on va connaître sa couleur, ses tags, et ses Exif (orientation) favoris

4. **Visualisation des données**

On visualise certaines données sur nos images, comme les images par année, par orientation, par couleur ou par tag (capture à gauche). On peut aussi voir les données par images (capture à droite)



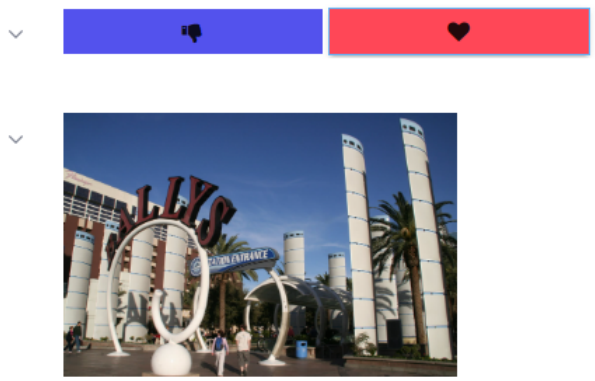


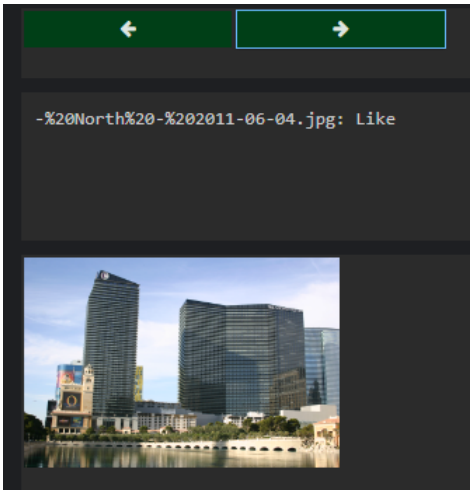
5. **Système de recommandation**

Pour le système de recommandation nous sommes partis sur un système en utilisant la classification.

Nous configurons notre encoder dans un premier temps avec les 128 images pour qu'il ait toutes les possibilités en matière de tags couleurs et orientation, puis nous séparons les images en un subset d'entrainement et un subset de test. L'utilisateur choisit ensuite ses images préférées en utilisant le même système que dans "Analyse de données" parmi un dixième des images pour récupérer les préférences de l'utilisateur (capture à gauche).

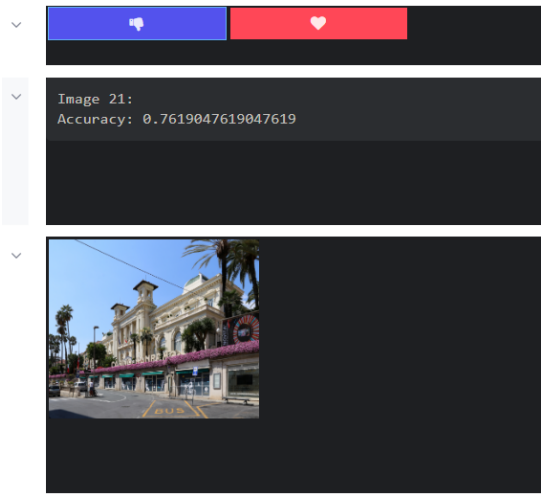
Un second système parcourt les images et informe l'utilisateur si d'après les images likées et dislikées en amont, les images proposées devrait être likées ou non (capture à droite).





6. Tests

On teste notre système de prédiction, en demandant à l'utilisateur de liker ou disliker des images supplémentaires si notre système à le même avis que l'utilisateur cela veut dire que notre système a bien compris les préférences et donc la précision augmente, dans le cas contraire elle descend.



L'auto-évaluation de notre travail :

1. Collecte de données
    a. Approches automatisées de la collecte de données → ✓ (depuis wikidata)
    b. Utilisation d'images sous licence libre → ✓ (Creative Commons)
    c. Stockage et gestion des images et des métadonnées associées → ✓
2. Étiquetage et annotation
    a. Approches automatisées de l'étiquetage → ✓/✗ (entrées utilisateur dans des fichiers, pourrait être amélioré)
    b. Stockage et gestion des étiquettes et des annotations des images → ✓
    c. Utilisation d'algorithmes de classification et de regroupement → ✓ (KMeans pour les couleurs)
3. Analyses de données
    a. Types d'analyses utilisées → ✓/✗ Les couleurs sont analysées, mais peut être amélioré: on pourrait mieux utiliser les tailles d'image, avoir + de clusters de couleur, etc...

       BONUS: Utilisation de *NearestNeighbors* pour faire des "tags de couleur"
    b. Utilisation de Pandas et Scikit-learn → ✓
    c. Utilisation d'algorithmes d'exploration de données → ✓ Demande à l'utilisateur effectuée
4. Visualisation des données
    a. Types de techniques de visualisation utilisées → ✓ Graphes et pywidgets
    b. Utilisation de matplotlib → ✓ Pour les graphes de fréquences (années, orientation, couleurs, tags)
5. Système de recommandation
    a. Stockage et gestion des préférences et du profil de l'utilisateur → ✓ (stocké en mémoire vive)
    b. Utilisation d'algorithmes de recommandation → ✓ Classification
6. Tests
    a. Présence de tests fonctionnels → ✓ Tests de chaque fonctionnalité dans le Jupyter (par exemple: le système de like, ou de création des vecteurs)
    b. Présence de tests utilisateurs → ✓ On teste la précision du modèle en comparant avec de nouveaux likes

7. Rapport
    a. Clarté de la présentation → ✔️ En français
    b. Présence d'une introduction et d'une conclusion claires, architecture des diagrammes, un résumé des différentes tâches réalisées et des limites → ✔️ Captures commentées
    c. Bibliographie → ✔️

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

On a apprécié la liberté laissée pour le choix des types images. Cependant, on a eu certaines difficultés à comprendre des instructions

Conclusion :

Grâce à ce projet, on a appris à analyser de la donnée pour prédire des choix utilisateurs. Mais on prend conscience que ça pourrait être utilisé dans d'autre cas : par exemple, pour continuer un début de phrase (comme ChatGPT), de musique, ou étendre une image existante (comme Photoshop Generative Fill).

Bibliographie :

https://www.wikidata.org/

https://fr.wikipedia.org/

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

https://ipywidgets.readthedocs.io/en/latest/

Cours de Data Mining (Plateforme CPE) - par John Samuel