

Rapport de projet DATA MINING

Joseph Pouradier-duteil & Pierre-Louis TELEP

Sommaire

- [Rapport de projet DATA MINING](#)
 - [Sommaire](#)
 - [1. Introduction](#)
 - [2. Présentation du projet](#)
 - [3. Présentation des données](#)
 - [4. Analyse des données](#)
 - [5. Prédiction](#)
 - [6. Auto-évaluation](#)
 - [7. Remarques](#)
 - [8. Conclusion](#)

1. Introduction

Ce rapport a pour but de présenter le projet de Data Mining. Le projet a été réalisé par **Joseph POURADIER-DUTEIL** et **Pierre-Louis TELEP**.

Le but de ce projet est de réaliser un système de recommandation. Un système de recommandation est un algorithme qui va analyser les données des images proposées à l'utilisateur. En fonction de si l'utilisateur aime ou pas les images l'algorithme va pouvoir proposer de nouvelles images et prédire si l'utilisateur va aimer ou non cette image. Pour cela nous avons utilisés des images de **Motos**, **Voitures**, **Pokemons** et **Exoplanet**.

2. Présentation du projet

Ce projet a pour but de faire un système de recommandation. Pour cela nous avons utilisés des images de **motos**, **voitures**, **Pokémons** et **planètes**. Nous avons utilisés des images trouvées en ligne sur WikiData que nous avons ensuite analyser pour récupérer leurs **métadonnées**.

Pour la partie de recommandation nous avons utilisé **SKLearn** et les **DecisionTree** pour analyser les choix de l'utilisateur en fonction des images proposées et prédire des images qu'il aime ou pas. Nous avons fait un deuxieme modèle de recommandation mais basé sur des préférences utilisateur aléatoire.

3. Présentation des données

Nous sommes allés sur wikidata pour récupérer les images de [motos](#), [voitures](#), [Pokémons](#) et [planètes](#). Pour télécharger les images nous avons utilisé un csv contenant les Urls des images.

Pour ensuite télécharger les images nous avons utilisé un script [python](#) qui va parcourir le csv et générer un dictionnaire contenant le nom de l'image, le dossier où enregistrer l'image, le lien pour télécharger l'image ainsi que certaines métadonnées. Ce même [script](#) va ensuite télécharger chaque image en utilisant [requests](#) avec des [get](#) et les enregistrer dans un dossier.

Une fois que toutes les images étaient téléchargées nous les avons utilisées avec un [script](#) python pour les mettre toutes au même format. Pour cela nous avons utilisé [PIL](#) et [numpy](#) pour les redimensionner au format 16:9, les normaliser et les convertir en [RGB](#).

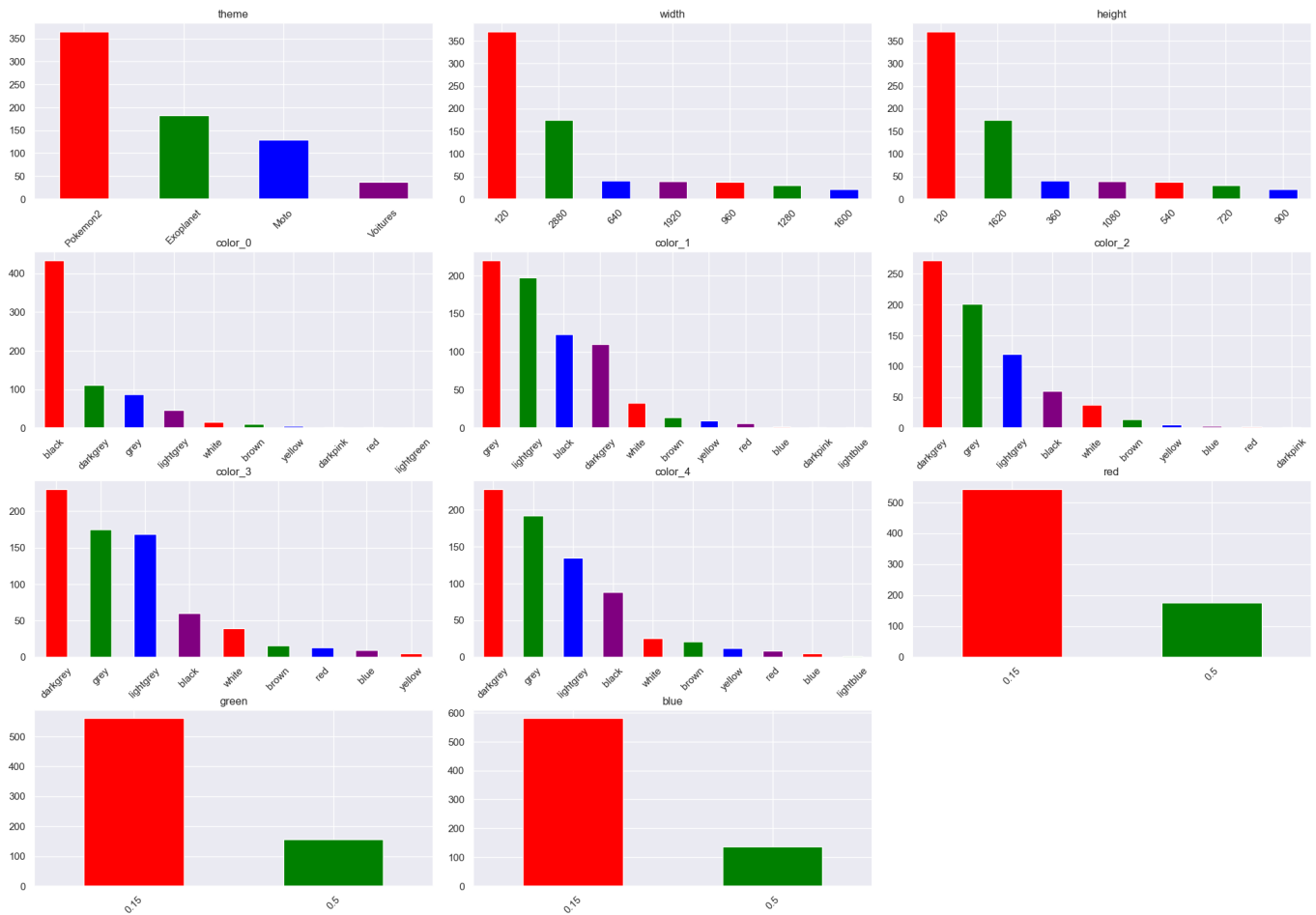
4. Analyse des données

Après avoir récupéré toutes les images, nous avons récupéré leurs métadonnées.

Nous avons utilisé les données exif des images telles que son thème, sa taille, son orientation quand elle était disponible et le format. Ensuite nous avons relevé les 5 couleurs prédominantes dans les images grâce à l'algorithme de [K-means](#). Par souci de performance nous avons utilisé les K-means MiniBatch qui est une version plus rapide des K-means. Toutefois le résultat des K-means était en hexadécimal et nous donnait une trop grande diffusion des couleurs (très peu probable d'avoir deux fois la même couleur même parmi nos 720 images). Nous avons donc utilisé la fonction [color_hex_to_rgb](#) pour convertir les couleurs en hexadécimal. Puis nous avons pris la couleur la plus proche dans un lot de 17 couleurs ([closest_color](#)). Nous avons ensuite utilisé l'histogramme des pixels rgb de chaque image pour en prendre une valeur normalisée entre 0 et 1: on fait une moyenne sur l'histogramme puis on prend la valeur normalisée de cette moyenne ([getDiagram](#)).

Enfin nous avons écrit toutes ces données dans un fichier json : [ExifDatatest2.json](#)

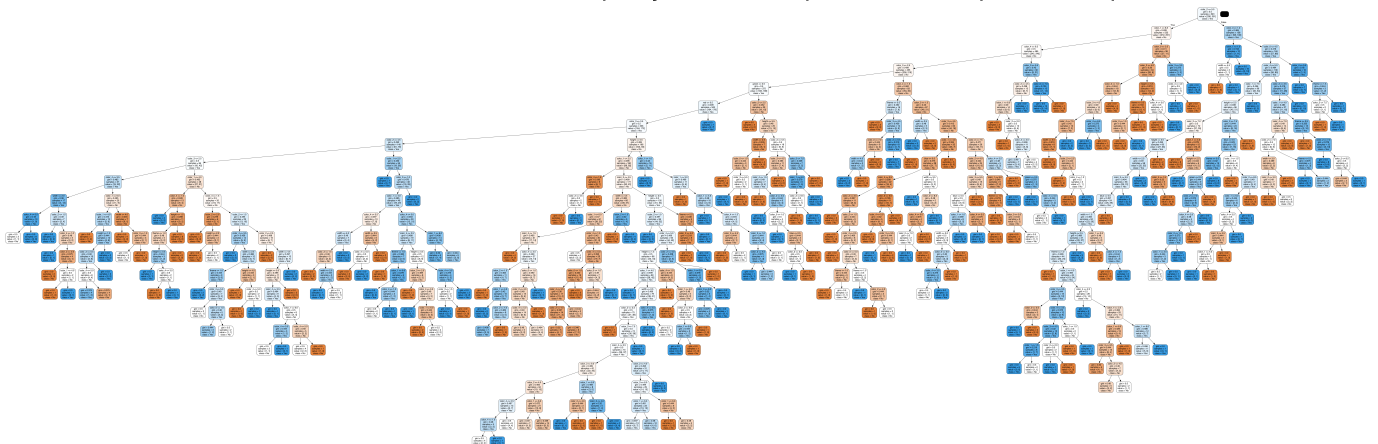
On peut maintenant en visualiser la répartition des données sur un graphique:
Number of element per value of each columns



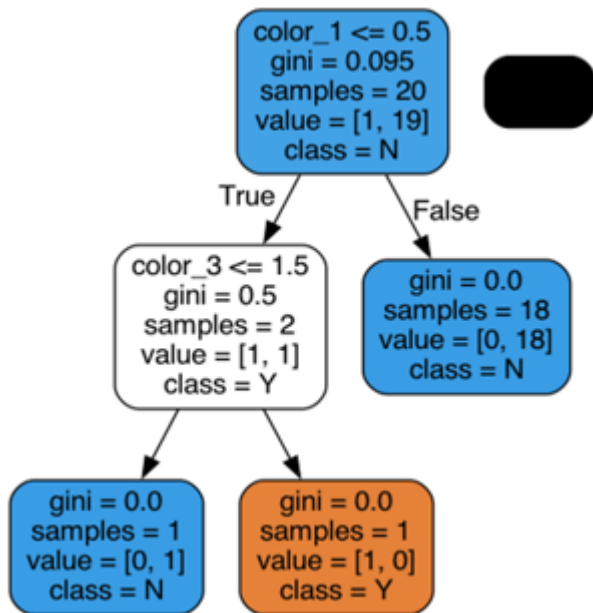
5. Prédiction

Pour la partie prédiction nous avons utilisé 2 méthodes différentes. La première est basé sur des préférences utilisateur aléatoires. La deuxième est basé sur les préférences d'un vrai utilisateur.

Pour la première méthode nous avons rempli la colonne préférence pour chaque image de façon aléatoire. Pour ensuite générer une prédiction en fonction de ces préférences, nous avons utilisé **SKLearn** et les **DecisionTree** pour analyser les choix de l'utilisateur en fonction des images proposées et prédire des images qu'il aime ou pas. Evidemment cette méthode n'est pas très fiable car les préférences sont aléatoires. On voit dans le modèle de décision qu'il y a beaucoup de noeuds qui ne font pas forcément sens.



Pour la deuxième méthode nous proposons à l'utilisateur 20 images et il doit choisir celles qu'il aime. Nous avons alors analysé les préférences de l'utilisateur. Avec ces analyses nous avons pu proposer 5 images à l'utilisateur dont le modèle avait prédit si la personne allait aimé ou non. Dans l'arbre de décision on voit que les noeuds sont plus cohérents et que le modèle est plus fiable.



6. Auto-évaluation

Dans notre test où les choix sont élatiores, nous entrainont le modele sur 70% de nos données et on test sur une image qui n'existe pas dans le dataset. On aurait pu aussi tester sur seulement 20 images aléatoires comme pour un vrai utilisateur pour pouvoir mieux comparer les résultats.

Ensuite au niveau de notre dataset, nous n'avons une répartition trop inégale des données:

- pas assez de voitures ni de moto.
- mauvais choix du set de couleur, trop de couleurs qui ne sont pas utilisées.
- pas assez de teintes sombres lors de la conversion des couleurs.

7. Remarques

Au niveau du cours on aurait préféré deux heures sur le data mining et wikidata et deux heures sur le filtrages des données. Et enfin 4h sur le machine learning pur, c'est à dire les différents modèles et leurs fonctionnement, les différentes approches en fonction de nos données, pourquoi utilisé tel ou tel modèles, comment vérifié la performance de notre modèles...

Et le cours devrait etre renommé en **Data mining et Machine Learning**.

8. Conclusion

Grace a la récolte d'image open source sur internet et grace a la librairie **SKLearn** nous avons pu faire un système de recommandation. Nous avons pu utiliser des images de **motos,voitures,Pokémons** et **Exoplanet**. Nous avons utilisés l'algorithme de **DecisionTree** pour prédire les préférences de l'utilisateur. Nous avons enfin pu comparer un modele de prédiction aléatoire et un modele de prédiction basé sur les préférences d'un utilisateur.