Data Mining & Machine Learning

RAPPORT DE PROJET

Lilian ANDRES & Joris GARCIA

SOMMAIRE

INTRODUCTION

A l'ère de la digitalisation, la quantité de données que nous produisons chaque jour explose de manière exponentielle. L'exploitation intelligente de ces ressources relève alors d'une importance capitale. En effet, les entreprises ont compris que l'analyse de données était un avantage stratégique et concurrentiel majeur dans leur croissance et leur développement.

Le présent rapport détaille le projet de Data Mining et Machine Learning entrepris dans le cadre de notre parcours académique, dont l'objectif principal était de collecter des images et leurs métadonnées afin de construire un système de recommandations personnalisées pour les utilisateurs.

Ce rapport explorera en détail les différentes étapes de notre projet, depuis la collecte initiale des données jusqu'à la mise en œuvre et l'évaluation du système de recommandations. Nous discuterons des défis rencontrés, des solutions adoptées et des leçons apprises tout au long du processus. Enfin, nous conclurons sur l'auto-évaluation de notre travail ainsi que sur quelques remarques concernant le déroulement de ce module.

RÉCOLTE DES DONNÉES

Afin d'automatiser notre récolte de données (images et autres informations contextuelles), nous nous sommes appuyés sur le site Internet WikiData (https://www.wikidata.org/), une base de connaissances libre et gratuite pouvant être lue et modifiée tant par des personnes que par des dispositifs informatisés. Toutes les ressources que nous avons utilisées se trouvent donc sous la licence CC0 1.0 DEED, qui fait référence à la licence Creative Commons Zero (CC0). Par conséquent, cela signifie que lorsqu'une œuvre est publiée sous la licence CC0, elle peut être utilisée, modifiée, distribuée et reproduite par n'importe qui, pour n'importe quel usage, sans demander la permission de l'auteur ou du détenteur des droits d'auteur, et ce, dans le monde entier.

Nous avons donc mis au point une requête au format SparQL afin de récupérer un ensemble de données brutes sur le thème des oeuvres d'arts, plus particulièrement des peintures, grâce à l'API exposée par WikiData. Ainsi, nous avons récupérés un jeu de données de 1400 lignes contenant les informations suivantes :

- Nom de l'oeuvre
- Nom et prénom de l'artiste
- Pays d'origine de l'oeuvre
- Genre de l'oeuvre
- Liste de propriétés utilisées pour décrire l'oeuvre (aussi appelé "depicts")
- Lien vers l'image associée à l'oeuvre

Nous avons ensuite supprimé les lignes contenant des images dont le format ne nous intéressait pas afin de ne conserver que des formats standards (PNG et JPEG) avant de télécharger la totalité de ces images (10,3 Go). Afin de faciliter la collaboration des membres de l'équipe et de permettre la persistance de nos données, nous avons stocké ces images ainsi que l'ensemble des données relatives à ces images (CSV) dans un répertoire partagé sur l'application Google Drive.

TRAITEMENT DES MÉTADONNÉES

Une fois ces étapes terminées, nous avons extrait les métadonnées principales de ces images. Nous avons donc récupérés, pour chacune des images téléchargées, les caractéristiques suivantes :

- Le nom du fichier de l'image
- La taille de l'image (en pixels)
- Le format de l'image
- Le mode (RGB, RGBA...)

Nous avons également extrait un certain nombre de caractéristiques Exif (Exchangeable image file format) de ces images :

- DateTimeOriginal (Date et heure exactes auxquelles la photo a été prise)
- DateTimeDigitized (Date et heure exactes de numérisation)
- SOSpeedRatings (Sensibilité ISO)
- ExposureTime (Temps d'exposition)
- FNumber (Ouverture du diaphragme de l'appareil photo)
- MeteringMode (Mode de mesure)
- ExposureBiasValue (Compensation d'exposition)
- FocalLength (Longueur focale)
- Make (Fabricant de l'appareil photo)
- Model (Modèle de l'appareil photo)

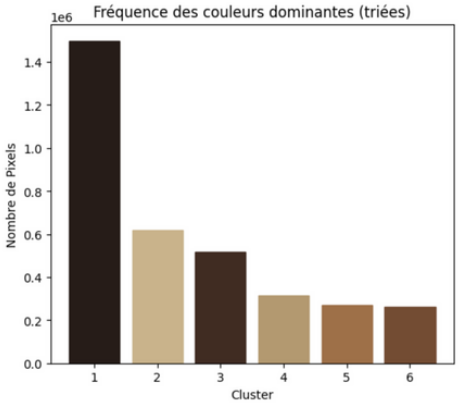
A l'instar des données précédemment citées, ces métadonnées ont été stockées dans des fichiers JSON distincts (i.e. un fichier JSON par image). Un DataFrame a ensuite été créé à partir de tout ces fichiers JSON afin de faciliter la manipulation de ces données. Parmi ces 1391 images restantes, seules 125 d'entre elles possédaient des métadonnées complètes, c'est-à-dire des métadonnées sans valeurs manquantes. Afin de conserver un jeu de données totalement homogène et pour ne pas fausser les résultats de nos recommandations, nous avons fait le choix de supprimer l'ensemble des images (et données associées) pour lesquelles certaines métadonnées étaient manquantes.

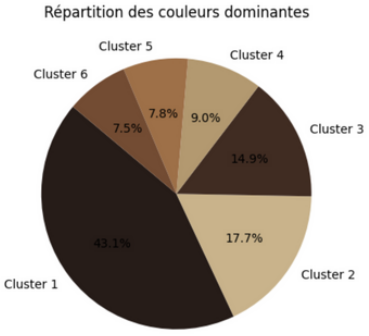
ÉTIQUETTAGE ET ANNOTATION

Dès lors que notre jeu de données était complet et propre, il ne nous restait que 125 images. L'objectif de la phase d'étiquettage et d'annotation était de compléter encore une fois les métadonnées que l'on possédait sur chacune des images en leur ajoutant des étiquettes, des tags et une liste de couleurs prédominantes.

Afin d'étiquetter nos images, nous nous sommes servis des données contextuelles que nous avions récoltées lors de la phase de récolte des données sur WikiData. Ainsi, nous avons ajouté dans les métadonnées de chaque image les champs "creator", "country" et "genre". Nous nous sommes également appuyés sur les "depicts" afin de tagger de manière descriptive les images de notre jeu de données.

Avant d'entamer la phase finale du processus d'étiquettage, à savoir l'extraction des couleurs prédominantes de chaque image, qui s'avérait plus coûteuse en termes de ressources que les précédentes tâches, nous avons opté pour la compression des images de notre collection. Cette décision visait à diminuer l'espace de stockage requis (2,3 Go) et à accélérer les futurs traitements numériques sur ces images. De ce fait, nous avons utilisé l'algorithme de clustering KMeans pour extraire arbitrairement 6 tuples de couleurs prédominantes au format (R, G, B) sur nos images avant de les intégrer dans les métadonnées de chacune d'elles. Cette opération aura duré une vingtaine de minutes. Afin de vérifier le bon fonctionnement de l'algorithme, nous avons affiché les informations d'une extraction aléatoire sélectionnée au cours du processus d'extraction global. Nous avons obtenu les résultats suivants, résultats totalement cohérents :





ANALYSE DES DONNÉES

Afin d'analyser efficacement les données et les préférences de l'utilisateur, nous avons mis en place, dans notre notebook, une interface permettant à l'utilisateur de sélectionner de manière dynamique ses images favorites parmi notre collection d'images. De même, nous avons créé un groupe de 3 utilisateurs fictifs supplémentaires auxquels nous avons affectés un nombre aléatoire d'images favorites aléatoires sélectionnées directement dans notre collection (entre 1 et 10 images par utilisateur fictif).

Nous avons fait le choix de ne stocker uniquement, pour chaque utilisateur, son identifiant ainsi que le lien des images qu'il avait préférées. Nous aurions évidemment pu y ajouter un ensemble d'informations complémentaires telles que le lien du fichier des métadonnées associés ou encore le nom de l'image mais cela n'aurait pas forcément été très utile. Dans un contexte où le stockage est une ressource précieuse, l'optimisation du stockage des données est un enjeu majeur et la présence de redondance d'informations n'est pas nécessaire, même si une telle approche accélèrerait nos développements.

Ainsi, nous avons pu identifier plusieurs caractéristiques essentielles à partir des préférences de l'utilisateur :

- Ses couleurs préférées (avec un algorithme de clustering KMeans)
- Ses tags préférés (avec un compteur d'occurrences)
- Ses orientations d'images favorites
- Ses genres préférés
- Ses créateurs favoris

VISUALISATION DES DONNÉES

La visualisation de données est le processus de représentation visuelle des informations, des données et des modèles pour en faciliter la compréhension et l'analyse. Elle représente un véritable outil de communication stratégique. Elle consiste à transformer des données brutes en graphiques, tableaux, cartes, diagrammes et autres représentations visuelles afin de mettre en évidence des tendances et des corrélations.

Pour réaliser notre visualisation de données, nous avons utilisé Matplotlib, la principale bibliothèque de visualisation de données Python. Nous avons ressorti dans notre jeu de données plusieurs informations intéressantes.

Nous avons effectué plusieurs types de visualisation différentes tout au long du projet (et pas seulement dans la partie “Visualisation de données”) :

- affichage des couleurs prédominantes (partie à l’ensemble)
- comptage de la répartition des images par année (histogramme)
- classement des 10 modèles d’appareils photos le plus utilisés (classement)
- répartition des fabricants d’appareils photos (partie à l’ensemble)
- distribution des tailles d’images (histogramme)

SYSTÈME DE RECOMMANDATION

Afin de mettre en place notre système de recommandation d'images basé sur les préférences de l'utilisateur, nous avons été contraint d'adapter le format de certaines de nos données dans notre jeu de données afin de les rendre exploitables et utilisables par un modèle d'apprentissage machine.

Par conséquent, nous avons donc transformé nos tuples de couleurs (R, G, B) en noms de couleurs textuels. De plus, nous avons également labellisé la taille des images et leur orientations respectives à partir des tuples de tailles (exprimés en pixels). Nous avons enfin supprimé les colonnes jugées inutiles pour entraîner notre modèle de recommandation d'images de notre jeu de métadonnées ('DateTimeOriginal', 'DateTimeDigitized', 'filename').

Un autre défi résidait dans le fait de traiter les tags et la liste des couleurs dominantes textuelles de manière à ce que notre modèle d'apprentissage soit en mesure d'utiliser ces informations correctement. Nous avons donc utilisé le modèle "MultiLabelBinarizer" sur la totalité des tags et la totalité des couleurs afin de convertir ces labels en encodage binaire, que nous avons concaténés à notre jeu de métadonnées.

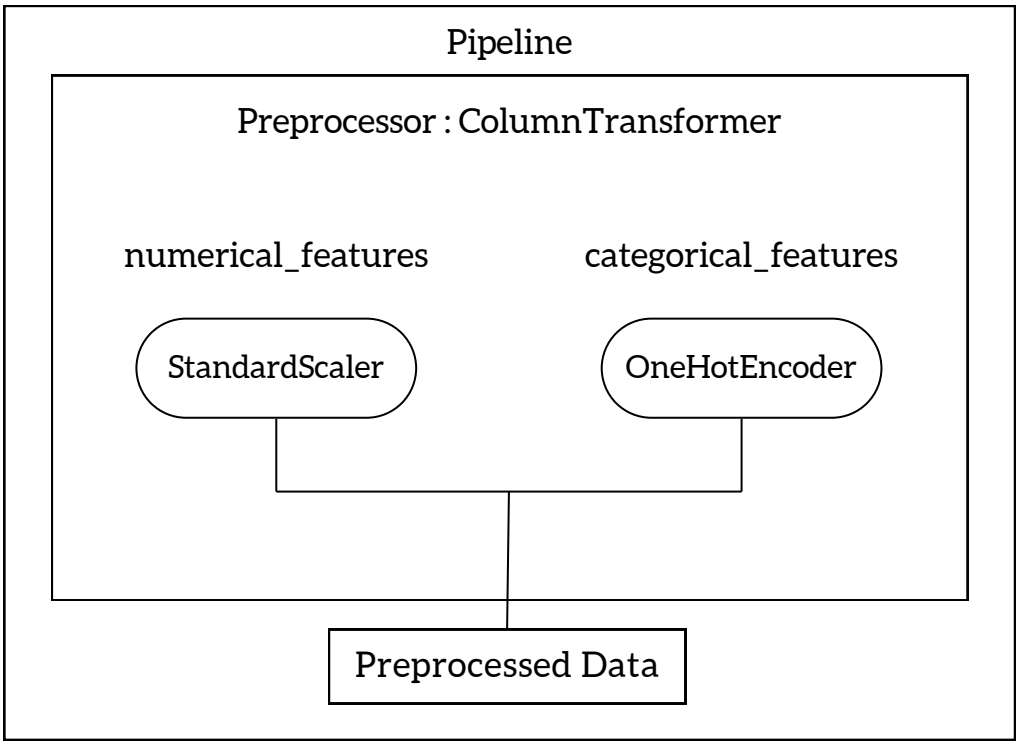
Cependant, notre jeu de métadonnées était toujours inexploitable à ce stade. En effet, il contenait des données binaires, mais également des données textuelles (catégoriques) et des données numériques. Afin de pré-traiter l'ensemble de ces données et de les rendre homogène, nous avons utilisé deux types de modèles de pré-traitement :

- Le StandardScaler pour mettre à l'échelle les caractéristiques numériques de manière à ce qu'elles aient une moyenne nulle et une variance unitaire
- Le OneHotEncoder pour convertir les caractéristiques catégorielles en représentations numériques binaires.

SYSTÈME DE RECOMMANDATION

Nous avons fait volontairement le choix de ne pas utiliser le modèle "LabelEncoder" afin de convertir nos caractéristiques catégorielles. En effet, lorsque l'on encode nos catégories à l'aide de ce modèle, une sorte de hiérarchie est créée entre les différentes étiquettes d'une même catégorie en fonction de l'ordre alphabétique (si une colonne contient 5 catégories différentes, on obtiendra 0 < 1 < 2 < 3 < 4). Ce phénomène aurait faussé les résultats de notre modèle d'apprentissage, c'est la raison pour laquelle nous avons utilisé un encodeur binaire.

Afin d'appliquer et de combiner ces modèles de pré-traitement sur nos données, nous avons utilisé un modèle de transformation nommé "ColumnTransformer".



Ainsi, notre jeu de données pré-traitées était prêt à être utilisé pour entraîner notre modèle d'apprentissage. Ce dernier contenait au total 368 features.

SYSTÈME DE RECOMMANDATION

Nous avons testé deux approches différentes concernant le mécanisme de recommandation : l'apprentissage non supervisé et l'apprentissage supervisé.

Pour le modèle d'apprentissage non supervisé, nous sommes partis sur un modèle de clustering KMeans que nous avons entraînés avec notre jeu de métadonnées soigneusement pré-traitées. Nous avons utilisé la méthode Silhouette afin de trouver le nombre de clusters optimal dans notre contexte. Nous avons également opté pour l'utilisation de la stratégie KMeans++ pour optimiser au maximum le placement initial des centroïdes.

Ainsi, il était possible de sélectionner les métadonnées d'une image favorite d'un utilisateur donné, de les pré-traiter de la même manière que les données ayant servi à entraîner le modèle (en utilisant les mêmes encodeurs, transformers et processeurs), et de les fournir à notre modèle afin qu'il puisse faire prédire le cluster auquel appartiendra l'image fourni et donc retourner un certain nombre d'images appartenant au même cluster.

Pour le modèle d'apprentissage supervisé, nous avons construit la liste des étiquettes du statut des images pour un utilisateur donné en respectant l'ordre des images de la collection globale afin de former une liste dont le format est le suivant: ['Favorite', 'NotFavorite', 'NotFavorite', 'NotFavorite', 'Favorite'].

Nous avons ensuite séparer le jeu de métadonnées en deux : un jeu d'entraînement et un jeu de test afin d'entraîner notre modèle SVC "C-Support Vector Classification" de la famille des modèles SVM "Support Vector Machines". Grâce au calcul du score de précision, nous obtenu un "accuracy_score" sur ce modèle de 0,8 soit 80%, ce qui est un très bon score.

De la même manière que pour le modèle précédent, il est possible de pré-traiter les métadonnées d'une image sélectionnée aléatoirement parmi notre collection d'images et de demander à notre modèle d'étiqueter l'image comme étant une image 'Favorite' ou 'NotFavorite'.

LIMITATIONS ACTUELLES

Cependant, notre approche n'est pas sans défaut et nous avons relevé un certain nombre de limites.

Premièrement, nous avons relevé le fait que notre système de recommandation n'est pas réellement scalable, de par la structure de données qu'il génère, gère et manipule. En effet, étant donné que nous utilisons un OneHotEncoder, si une nouvelle image n'ayant pas servi à entraîner le modèle d'apprentissage lui est fournie, et que les métadonnées de cette images contiennent de nouvelles balises (i. e. inconnues du modèle) comme par exemple de nouvelles couleurs ou de nouveaux tags, de nouvelles colonnes vont apparaître durant le pré-processing de l'interaction utilisateur. De ce fait, le modèle sera incapable de réaliser une prédiction sur ces données puisqu'il n'aura pas été entraîné avec le format de données que l'on lui fournit. Un nouvel entraînement du modèle, avec ces nouvelles colonnes, serait alors nécessaire, ce qui n'est pas forcément idéal.

Ensuite, nous avons remarqué que certaines tâches manquent d'automatisation. C'est le cas par exemple de la partie "Analyse de données". Nous aurions pu créer des méthodes plus génériques permettant d'extraire un certain nombre d'informations concernant un utilisateur donné. Ces méthodes auraient donc été réutilisables et plus simples à tester.

Enfin, nous avons remarqué que les résultats obtenus n'étaient pas forcément très pertinent. En effet, nous avons testé notre système de recommandation visuellement en affichant les images recommandées ainsi que leurs métadonnées respectives. Ces résultats n'ont pas été très "parlants" pour nous. Nous pensons que ce phénomène est lié à l'absence de données supplémentaires dans notre jeu de données. Il nous aurait fallu beaucoup plus de données pour pouvoir observer de vrais résultats, plus cohérents.

CONCLUSION

Pour conclure, nous avons réalisé la mise en place d'un système de recommandation d'images basé sur les préférences des utilisateurs, comme pourrait le faire un réseau social, au travers d'un projet riche et complet. Nous avons eu l'occasion d'étudier les axes principaux du Data Mining : de l'extraction à l'analyse, en passant par la visualisation de données et l'exploitation de modèles d'apprentissages machines supervisés et non supervisés. Ce projet aurait évidemment pu être poussé plus loin, notamment en y ajoutant de l'automatisation et plus de diversité dans le choix des modèles d'apprentissages utilisés. Nous aurions également pu pousser les tests fonctionnels et utilisateurs plus loin. Nous sommes toutefois satisfaits du projet que nous livrons ce jour.

Concernant le module et les séances pratiques, nous avons plusieurs remarques et observations. Le domaine du Data Mining et du Machine Learning sont des domaines particulièrement complexes. Ce module est suffisant pour acquérir certaines connaissances théoriques de bases mais il reste globalement assez superficiel. Les cours sont vulgarisés correctement pour nous permettre de mieux comprendre les principes et concepts essentiels de ces domaines mais certains détails manquent toutefois. Certaines formations en école d'ingénieur sont axés entièrement sur le Big Data et le Machine Learning et les concepts sont abordés dans le détail au fur et à mesure. Un module aussi court que celui-ci ne nous permet pas de comprendre en profondeur le fonctionnement des modèles de Machine Learning que nous manipulons. Il en va de même pour les réseaux de neurones. Proposer un panel d'une dizaines d'exemples de modèles d'apprentissages différents pour nous demander de les utiliser n'est pas forcément la meilleure approche puisque nous ne comprenons même pas ce que nous manipulons, nous disposons seulement de quelques connaissances génériques, ce qui est dommage. C'est également dommage que CPE ne propose pas de majeur spécialisée dans ces domaines qui sont de plus en plus dans l'ère du temps...