Compte-Rendu: Projet Data Mining

Majeure: CLBD, 4ETI
Étudiants: OUENADIO Alexandre, GUZELIAN Raphaël

But du projet

L'objectif de ce projet est de recommander des posters de film en fonction des préférences de l'utilisateur.

Sources des données collectées et leur licences

Nos données proviennent de la base de données de The Movie Database (TMDB).

La licence de TMDB nous permet d'avoir un usage des données, issues de contributions utilisateurs, uniquement personnel et non-commercial (cf. "Terms of Use")

Taille des données

100 films provenant de cette liste TMDB: https://www.themoviedb.org/list/1235

Modèles d'exploration de données et/ou d'apprentissage machine utilisés avec les métriques obtenues

- Clustering : KMeans pour trouver la couleur dominante de chaque poster
- Classification: Decision Tree

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

Les notions de machine learning sont assez théoriques et un peu dur à assimiler, cela aurait été mieux d'avoir quelques exemples pratiques, concrets, précis et expliqués dans le cours. Le cours actuel est rempli de concepts sommairement expliqués ce qui est bien mais aussi perturbant car l'on ne sait pas trop ce qui est important de mémoriser dans le cadre du module.

I. Collecte des données

Les données sont collectées de manière automatique par des requêtes HTTP à l'API de TMDB, stockées dans un fichier JSON, puis transformées en dataframe avec la librairie python "pandas".

Ces dernières représentent des informations sur le top 100 films créé par un utilisateur de TMDB.

Dans le cadre de notre travail, nous récupérons pour chaque film les informations suivante:

- le titre du film
- sa popularité TMDB
- son genre (Adventure, Mystery, Drama, …)
- sa période de sortie (40s, 80s, 2000s,...)
- son poster (au format jpg et de 500px de largeur)

II. Etiquetage et annotation

Pour chaque film, nous analysons le poster (après l'avoir téléchargé dans le dossier "images") et déterminons la couleur dominante du poster grâce à l'algorithme de regroupement de données Kmeans et la librairie python "webcolors" puis ajoutons cette nouvelle information dans le fichier JSON des films.

III. Analyse des données

Rappelons que notre objectif est d'analyser les préférences ("Favorite" ou "Not Favorite") d'un utilisateur et de lui recommander des films en fonction de ces dernières.

L'analyse des préférences d'un utilisateur est réalisée par l'utilisation d'un algorithme de classification de la librairie "scikit-learn": l'arbre de décision (Decision Tree)

Il repose sur de l'apprentissage supervisé qui consiste à donner à l'algorithme dans un premier temps des données et des étiquettes de reconnaissance des données: c'est la phase d'entraînement. Dans un second temps, l'algorithme doit prédire les étiquettes de nouvelles données similaires fournies: c'est la phase de prédiction.

Ici, nos données représentent les informations des films collectées (genre, période de sortie et couleur dominante) et les étiquettes la préférence de l'utilisateur pour un film donné.

Ensuite vient la phase de prédiction. L'algorithme, après avoir mémorisé approximativement le profil de l'utilisateur à la phase d'entraînement, doit pouvoir prédire si en lui donnant d'autres films, l'utilisateur aimera ces films.

Nous aurions pu faire notre analyse sur d'autres algorithmes de classifications tels que Perceptron, Perceptron multicouche (MLP) ou bien la machine à vecteurs de support (SVM), mais par manque de temps nous nous sommes limités à l'algorithme d'arbre de décision.

IV. Visualisation des données

Nous pouvons exploiter les données sur chaque film pour en faire des études statistiques.

Nous avons ainsi pu réaliser des graphiques à l'aide des librairies "pandas" et "matplotlib" pour tracer:

- Le nombre de films disponibles par période de sortie
- Le nombre de films disponibles par genre
- La proportion des couleurs dominantes dans le lot de posters téléchargés
- Les informations relatives au profil d'un utilisateur (genre préférés, périodes de sortie, couleurs et films préférés)

Remarques concernant les graphiques:

Pour le diagramme en bâtons de la proportion des couleurs dominantes, toutes les couleurs dominantes sont affichées sur les barres mais certaines ont été décalées et ne correspondent pas au label indiqué en abscisse, on suspecte un bug de la part de la librairie pandas…

V. Système de recommandation

Pour réaliser notre système de recommandation, nous avons tout d'abord suggéré des listes de 3 tags générés aléatoirement à notre algorithme de prédiction. L'algorithme prédit alors si ces listes de tags sont étiquetées "Favorite" ou "NotFavorite".

Si l'étiquette "Favorite" est associée à une liste de tags, on récupère les tags qui sont contenus dans cette liste. On construit alors une liste contenant tous les tags qui ont été classés "Favorite". Nous comptons ensuite l'occurrence de ces tags pour extraire les tags favoris, c'est-à-dire ceux qui apparaissent le plus de fois. Nous stockons ces tags favoris dans un fichier JSON pour construire un profil d'utilisateur.

Nous recherchons ensuite dans nos informations collectées sur les films quels films possèdent les tags appréciés par l'utilisateur. Si un film est susceptible d'être aimé par l'utilisateur alors nous lui recommandons le film en lui montrant son poster. L'utilisateur peut alors choisir de classer le film en tant que favoris ou non. Nous avons décidé de stocker uniquement les films classés en tant que favoris dans son profil. Nous pouvons alors rajouter au fichier JSON une liste des films classés comme favoris pour chaque utilisateur.

Pour améliorer notre approche, nous pouvons également stocker les tags des films classés favoris, puis proposer uniquement d'autres films ayant ces tags. Cela limitera le nombre de nouvelles propositions de films mais on saura que le prochain film proposé aura plus de chance d'être en favoris car il correspondra plus à son profil. Nous pouvons aussi raisonner de manière inverse et ne plus proposer de films ayant des tags qui appartenaient à un film que l'utilisateur a classé en non-favoris.

Conclusion

Au terme de ce projet, nous avons pu recommander des posters de film en fonction des préférences de l'utilisateur. Nous avons trouvé ce projet très intéressant car il permet d'allier programmation et manipulation de données. Le projet nous a laissé une certaine forme de liberté dans nos choix de conception ce qui est une bonne chose mais cela ne permet pas de se rendre compte si notre projet a la forme attendue ou non. Néanmoins, il nous a permis d'avoir un premier aperçu positif sur le data mining.