Recommandation des filmes avec analyse de sentiments

Projet de fin d'année

Développeur en Intelligence Artificielle

2021 / 2022

MURO R. Daniel Alexander

Le besoin

Pour ce sujet, le point de départ était un code assez simple qui permet d'analyser un dataset de synopsis des films, et propose une recommandation de 10 titres à partir du titre d'un autre. Pour cela, l'algorithme vectorise la data à l'aide du TfidfVectorizer, et ensuite calcule les coefficients de la fonction sigmoïde entre les films, en utilisant le sigmoid_kernel de Scikit-learn.

La demande était :

1. Enrichir la data en trouvant une autre source de données
2. Améliorer l'algorithme en incorporant du Sentiment Analysis

La première étape a été, évidemment, la recherche d'une source de données. Il n'est pas compliqué de trouver sur le web, une énorme quantité de datasets orientés à l'entraînement de modèles pour l'analyse de sentiments, avec des milliers des textes classés "positive" ou "négative". Cependant, la plupart de ces jeux des données ne font pas la liaison entre ces critiques et les films, donc il n'est pas du tout possible de savoir avec certitude à quel film correspond chaque critique. Malheureusement, aucun de ces datasets ne nous est utile pour cette étude.

Ils existent également des datasets avec une information assez complète à propos de films, et notamment son synopsis, mais sans aucun commentaire ou critique. Ces jeux de données ne peuvent pas non plus être utilisés dans le cadre de notre étude.

La solution pour ce problème commence par un dataset trouvé sur Kaggle qui contient des synopsis de plus de 14k films mais aussi l'identifiant IMDB de chacun. Avec cette information, j'ai branché l'outil sur une API mise à disposition par cette plateforme (IMDB), et j'ai récupéré des informations telles que :

- Evaluation moyenne donnée par les spectateurs sur le site d'IMDB, en échelle sur 10
- Evaluation moyenne donnée sur le site Metacritic, sur 100

- Evaluation moyenne donnée sur le site TheMovieDb, sur 10
- Evaluation moyenne sur RottenTomatoes, sur 100
- Evaluation sur FilmAffinity, sur 10
- Et, le plus important, un ensemble de critiques données à chaque film en particulier, avec le texte de chaque critique et une qualification associée, toujours donnée par les utilisateurs.

Additionnellement, j'ai trouvé, toujours sur Kaggle, un autre dataset contenant 50K critiques de films, classées selon son sentiment général.

Une fois récupérées toutes les informations, l'étape suivante a été la construction et entraînement du modèle d'analyse de sentiments. Tout d'abord, j'ai procédé à nettoyer chaque texte à l'aide de deux méthodes qui vont s'occuper de :

- Enlever les étiquettes HTML présentes sur les textes récupérés à l'aide de l'API.
- Enlever les caractères non-alphabétiques, les majuscules et les signes de ponctuation.
- Diviser chaque texte en mots.
- Lemmatiser les mots, afin de convertir chaque mot à sa forme canonique
- Supprimer les mots vides, ou "stop words"
- Reconstituer une phrase avec les mots restants après le traitement.

Toutes ces étapes du prétraitement vont permettre aux modèles de se focaliser sur les mots les plus importants et sa relation entre eux.

Ensuite, et une fois faites les vérifications pour m'assurer qu'il n'y avait pas d'entrées en doublon, vides ou avec des valeurs qui ne correspondaient pas au positif et négatif (valeurs que, afin de pouvoir être traités, ont été convertis en format binaire), j'ai procédé à créer une pipeline pour enchaîner un modèle Tf-Idf pour vectoriser les textes, et un modèle MultinomialNB pour sa classification et futures prédictions. Le fait de créer une pipeline m'a permis d'exécuter une Grid Search afin d'optimiser, en essayant plusieurs paramètres pour chacun des deux modèles. Finalement, j'ai récupéré les meilleurs paramètres et j'ai exporté le modèle d'analyse de sentiments, afin de pouvoir le récupérer plus tard.

Séparément, j'ai pré-processé les synopsis des films d'une façon similaire, afin de supprimer les caractères et mots qui n'intéressent pas au modèle du NLP. Pour chaque film, j'ai pris sa liste de reviews et je les ai passés tous par le modèle d'analyse de sentiment, afin d'avoir une note binaire correspondante au sentiment de la critique. J'ai, ensuite, calculé la moyenne des notes obtenues à l'aide de ce modèle du NLP, et la moyenne des notes laissées par chaque utilisateur avec ses commentaires. Finalement, j'ai calculé la moyenne entre toutes les qualifications par film (IMDB, Metacritic, TheMovieDb, RottenTomatoes, Moyenne des notes laissées avec les commentaires et Moyenne des notes obtenues à l'aide de l'analyse de sentiment), ce qui m'a permis d'obtenir une note globale moyenne par film.

Ensuite, j'ai vectorisé l'ensemble des synopsis et des notes moyennes calculées à l'aide du TfIdfVectorizer, et j'ai passé ces deux matrices obtenues par un algorithme sigmoid_kernel de scikit-learn, afin d'obtenir les films les plus proches à celui demandé, mais ne pas uniquement en fonction de sa synopsis mais aussi en fonction de la qualification moyenne.