Rapport Projet TAL

Aymane EL MAHI

SOMMAIRE :

1- Introduction

Ce projet a pour objectif de développer un système de récupération d'informations dans une collection de descriptions de films publiées sur Allociné. L'enjeu principal est d'automatiser la classification des films par genre en nous basant sur le texte de leur synopsis et leur titre. Cette tâche peut être particulièrement complexe, car les genres de films peuvent être subtils et leur classification peut varier en fonction des critères et des contextes.
Pour atteindre notre objectif, nous allons explorer plusieurs approches de classification automatique et évaluer leur pertinence en utilisant des mesures de performance standard. Nous allons également étudier les techniques d'analyse de texte les plus adaptées à la nature des données et aux besoins du projet.

- Problem Understanding

Le projet consiste à développer un système de récupération d'informations dans une collection de descriptions de films provenant d'Allociné. Il sera divisé en deux étapes distinctes. La première étape portera sur la création d'un outil de classification automatique des genres de films basé sur le texte des synopsis et des titres. Pour ce faire, nous utiliserons deux fichiers CSV contenant des descriptions de films en français fournis sur Moodle. Différents algorithmes de classification et caractéristiques de texte seront comparés et évalués selon les meilleures pratiques telles que la validation croisée.
La deuxième étape consistera à appliquer le modèle sélectionné à un nouvel ensemble de données et à produire des données étiquetées avec les genres prédits. Nous analyserons les performances du modèle en termes de précision, de rappel et de F1-score. Nous évaluerons également la capacité du modèle à généraliser à de nouveaux ensembles de données et nous proposerons des pistes d'amélioration pour des performances optimales.

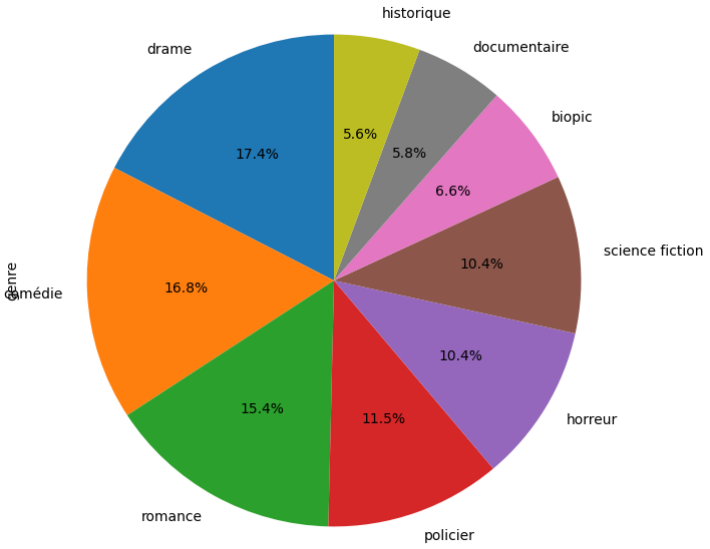
2- Preprocessing

- Data Understanding

Les données sont fournies dans deux fichiers CSV, l'un pour l'entraînement et l'autre pour les tests. Les données d'entraînement contiennent 2875 films avec les 22 champs suivants :

- acteur_1
- acteur_2
- acteur_3
- allocine_id
- annee_prod
- annee_sortie
- box_office_fr
- couleur
- duree
- langues
- nationalite
- nb_critiques_presse
- nb_critique_spectateurs
- nb_notes_spectateurs
- note_presse
- note_spectateurs
- realisateurs
- synopsis
- type_film
- genre

Ces champs contiennent des informations telles que les acteurs, les réalisateurs, les années de production et de sortie, les langues, les nationalités, les notes des critiques et des spectateurs, les résumés de films et les genres. Le champ "genre" sera utilisé comme variable cible pour la classification des films.
Les données de test contiendront des films similaires, mais sans l'étiquette de genre, et seront utilisées pour évaluer la performance du modèle de classification. Nous utiliserons également ces données pour évaluer la capacité du modèle à généraliser à de nouveaux ensembles de données.



- Prétraitement de la colonne Titre :

```python
def process_title(df):
    df['titre'] = df['titre'].str.lower()
    df['titre'] = df['titre'].str.normalize('NFKD').str.encode('ascii', errors='ignore').str.decode('utf-8')
    df['titre'] = df['titre'].str.translate(str.maketrans('', '', string.punctuation))
    df['titre'] = df['titre'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stopwords)]))
    df['titre_use'] = df['titre'].apply(lemmatize_text)
    return df
```

La fonction "process_title" est une fonction de prétraitement des données pour la colonne "titre" d'un dataset. Elle effectue plusieurs étapes pour nettoyer et normaliser les données.

Tout d'abord, elle convertit tous les titres en minuscules pour éviter les erreurs de casse et faciliter la manipulation des données. Ensuite, elle utilise la méthode "normalize" pour supprimer les accents et convertir les caractères spéciaux en caractères ASCII. Elle supprime également la ponctuation en utilisant la méthode "translate" avec une table de correspondance vide.

Ensuite, la fonction applique un filtre "stopwords" pour enlever les mots courants qui n'ont pas beaucoup de signification pour l'analyse de texte. La fonction de lemmatisation "lemmatize_text" est ensuite appliquée pour normaliser les mots en les ramenant à leur forme de base.

Finalement, la fonction renvoie le dataframe avec une colonne supplémentaire "titre_use" contenant les titres prétraités. Cette fonction est utile pour améliorer la qualité des données et faciliter l'analyse ultérieure du dataset.

- Prétraitement de la colonne Synopsis :

```python
def process_synopsis(df):
    remove_NER(df, 'synopsis')
    df['synopsis'] = df['synopsis'].str.lower()
    df['synopsis'] = df['synopsis'].str.normalize('NFKD').str.encode('ascii', errors='ignore').str.decode('utf-8')
    df['synopsis'] = df['synopsis'].str.translate(str.maketrans('', '', string.punctuation))
    df['synopsis'] = df['synopsis'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stopwords)]))
    df['synopsis_use'] = df['synopsis'].apply(lemmatize_text)
    return df
```

La fonction "process_synopsis" est une autre fonction de prétraitement de données pour la colonne "synopsis" d'un dataset. Cette fonction effectue les mêmes étapes de nettoyage et de normalisation que la fonction "process_title".
Cependant, cette fonction comporte une étape supplémentaire qui consiste à supprimer les "named entities" (NER) du synopsis en utilisant la fonction "remove_NER". Les named entities sont des entités nommées comme les noms de personnes, de lieux ou d'organisations, qui peuvent perturber l'analyse de texte si elles sont incluses dans le corpus. En supprimant les NER, la fonction permet de mieux se concentrer sur les mots clés pertinents pour la classification des genres de films.
Comme pour la fonction "process_title", la fonction renvoie le dataframe avec une colonne supplémentaire "synopsis_use" contenant les synopses prétraités. Ces deux fonctions de prétraitement sont essentielles pour améliorer la qualité des données avant l'analyse et la classification des genres de films.

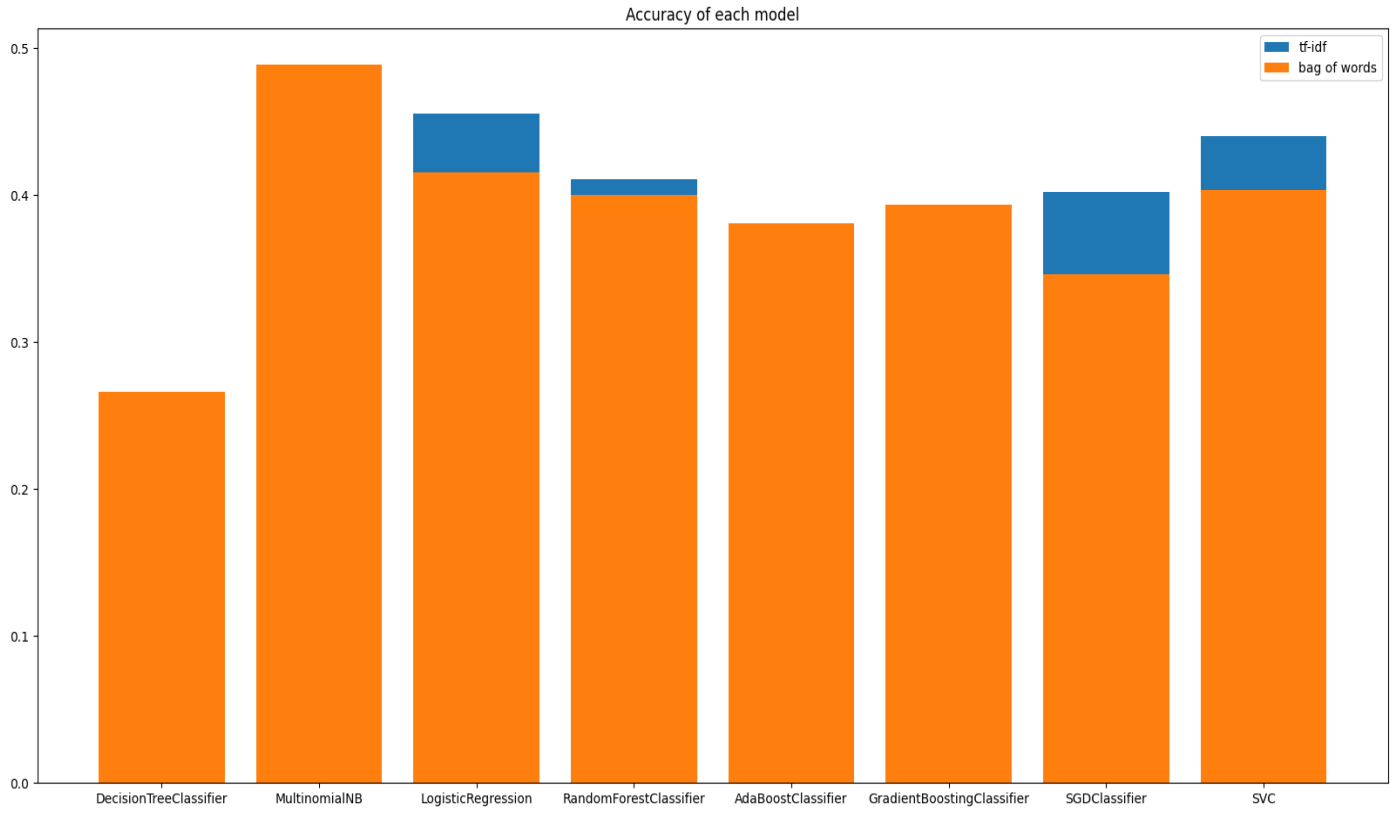
3- Différentes approches:

- B-o-W et TF-IDF:

Dans le cadre de ce projet, j'ai utilisé différentes techniques pour prétraiter les données textuelles avant de les utiliser pour entraîner un modèle de classification de genre de films. Après avoir nettoyé les données et supprimé les stopwords et la ponctuation, j'ai d'abord utilisé une approche de bag-of-words pour représenter les textes sous forme de vecteurs numériques. Cette approche consiste à considérer chaque document comme un sac de mots et à créer un vocabulaire unique pour l'ensemble des documents. Les vecteurs de chaque document sont ensuite construits en comptant le nombre d'occurrences de chaque mot du vocabulaire dans le document.

Ensuite, j'ai utilisé une approche plus avancée de la représentation de texte, appelée TF-IDF (term frequency-inverse document frequency). Cette approche prend en compte la fréquence de chaque mot dans le document, ainsi que sa fréquence inverse dans l'ensemble des documents. Cette approche permet de mieux représenter la pertinence d'un mot pour un document donné en considérant également sa fréquence dans l'ensemble des documents.

En testant plusieurs algorithmes tels que LogisticRegression, DecisionTreeClassifier, MultinomialNB, RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier …,
j'ai eu une précision de 50%, et je ne pense pas qu'il est possible de faire mieux avec de tels modèles.



- Camembert-base:

Enfin, j'ai utilisé une approche plus récente et plus avancée de représentation de texte, en utilisant les modèles de langage pré-entraînés tels que les Transformers. Les modèles de langage basés sur les Transformers sont capables de prendre en compte le contexte et la sémantique des mots pour produire des représentations de texte plus riches et plus précises. J'ai utilisé un modèle pré-entraîné “camembert-base” pour encoder les descriptions de films en vecteurs numériques, qui ont ensuite été utilisés pour entraîner un modèle de classification de genre de film. Cette approche a permis d'obtenir des résultats un peu meilleurs que les approches précédentes, ceci est principalement dû à la lemmatisation, sans celle-là, les transformers ont pû monter jusqu'à 60% de précision, tandis qu’avec la lemmatisation, ça ne dépasse pas les 50% . J’ai aussi laissé les NER car je me suis dit qu’ils peuvent aider à classer certain films contenant des personnages ou des lieux pertinents (Sherlock Holmes, Hercule Poirot…). Il faut dire que je me suis beaucoup plus concentré sur le fait d’améliorer les modèles vus dans la première approche.

J’ai voulu tenter d’utiliser le modèle “cmarkea/distilcamembert-base”, mais j’ai eu des problèmes de mémoire GPU sur mon pc personnel mais aussi sur Google Collab, cependant le code est dans la fin du model.ipynb pour ce modèle ci.

4- Conclusion :

En arrivant à la fin de ce projet, j’ai réalisé que différentes approches nécessitent différents prétraitements, et ce qui est optimal pour une approche peut être très néfaste pour une autre, c’est le cas de la lemmatisation ici.

Bibliographie:

- https://huggingface.co/cmarkea/distilcamembert-base
- https://github.com/Wonuabimbola/movie-genre-prediction/blob/main/predict_movie_genre_from_plot.ipynb
- https://maelfabien.github.io/machinelearning/NLPfr/#9-transformers-
- ChatGPT