**[Partie 1: Text classification: prédire si la vidéo est une chronique comique](https://github.com/afouchet/nlp_esgi/tree/main" \l "partie-1-text-classification-prédire-si-la-vidéo-est-une-chronique-comique)**

Premièrement nous avons run le pipeline pour voir si cela marche. Pour cela nous avons fait un feature simple qui permet de retourner le fait si des données sont présentes dans les colonnes is\_name,is\_comic et comic

Nous avons ensuite optimisé les features faites dans "make\_features ". Nous avons utilisé des features pour prétraiter les titres de vidéos en supprimant les caractères indésirables tels que les symboles. Pour cela on a converti les mots en leur forme de base et en supprimant les mots courants sans signification significative. En résumé nous avons préparé des données textuelles avant de les utiliser pour entraîner des modèles.

Nous utiliserons plusieurs modèles :

* Le modèle est un pipeline avec un CountVectorizer suivi d'un RandomForestClassifier.
* Le modèle est un pipeline avec un CountVectorizer suivi d'un choix de classificateurs le MultinomialNB.
* Le modèle est un pipeline avec un CountVectorizer suivi d'un choix de classificateurs le LogisticRegressionest .
* Le modèle est un pipeline avec un CountVectorizer suivi d'un choix de classificateurs le SVC.

Voici leur accuracy :

|  |  |  |  |
| --- | --- | --- | --- |
| RandomForest | Logistic Regression | Support Vector Machine (SVM) | Multinomial Naive Bayes |
| 89 ~ 91% | 91% | 92% | 89% |

Le meilleur accuracy obtenue est celle du modèle SVM

La classe DumpableModel fournit des méthodes pour sauvegarder et charger le modèle, en plus des méthodes standard d'entraînement et de prédiction.  
  
  
Pour finir nous avons adapté les fichiers train et predict pour qu’ils marchent.