

21/11/2024

**

*Project Machine Learning*

*The labeled data set consists of 50,000 IMDB movie reviews, specially selected for sentiment analysis*

*Table des matières*

*[Introduction 1](#_Toc12544)*

*[Etude sur le Dataset 2](#_Toc25062)*

*[Importation des bibliothèques requis 2](#_Toc20214)*

*[Prétraitement et normalisation de texte 3](#_Toc24944)*

*[Vectorisation du texte prétraité 4](#_Toc21039)*

*[Conception, entraînement et évaluation des performances du model. 5](#_Toc10202)*

*[Le Processus 6](#_Toc3538)*

*[Choix du modèle le plus performant 8](#_Toc19184)*

*[Conclusion 8](#_Toc6834)*

# Introduction

**C'est quoi l'analyse des sentiments ?**

L’analyse des sentiments c’est l'acte de reconnaître et de catégoriser informatiquement les opinions contenues dans un morceau de texte, notamment afin de discerner si l'écrivain a une attitude bonne, négative ou neutre envers un sujet, un produit, etc.

L'analyse des sentiments est une technique permettant d'analyser un morceau de texte afin de déterminer le sentiment qu'il contient. Pour ce faire, il combine l’apprentissage automatique et le traitement du langage naturel (NLP).

L'idée clé est d'utiliser des techniques d'analyse de texte, de PNL, d'apprentissage automatique et de linguistique pour extraire des informations ou des points de données importants d'un texte non structuré. Cela peut à son tour nous aider à obtenir des **résultats qualitatifs** comme le sentiment général sur une échelle **positive,** **neutre** ou **négative** et des **résultats quantitatifs** comme la **polarité du sentiment**, la subjectivité et les proportions d'objectivité.

**La polarité des sentiments** est généralement un score numérique attribué aux aspects positifs et négatifs d'un document texte en fonction de paramètres subjectifs tels que des mots et des phrases spécifiques exprimant des sentiments et des émotions. Un sentiment neutre a généralement une polarité de 0 car il n'exprime pas de sentiment spécifique, un sentiment positif aura une polarité > 0 et un sentiment négatif < 0.

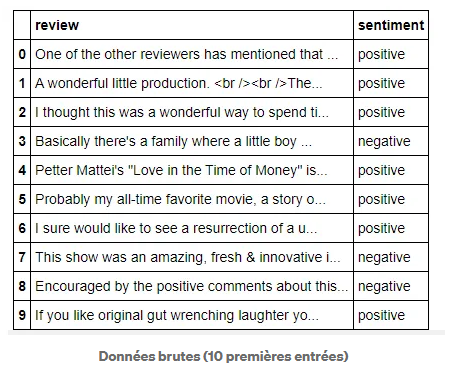
# Etude sur le Dataset

C’est un ensemble de données qui contient 50 000 critiques de films pré-étiquetées avec des étiquettes de classe de sentiments « positifs » et « négatifs » en fonction du contenu de la critique. L’ensemble de données peut être obtenu sur

*<https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>*

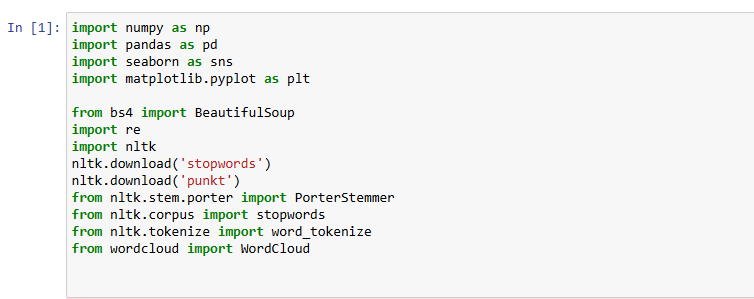
**Objectif du problème**

L'objectif principal du projet est de prédire le sentiment d'un certain nombre de critiques de films obtenues à partir de la base de données de films Internet (IMDb).

*Contenu:*

## Importation des bibliothèques requis

Tout d’abord, nous devons nous assurer que les bibliothèques requis sont installés puis importés

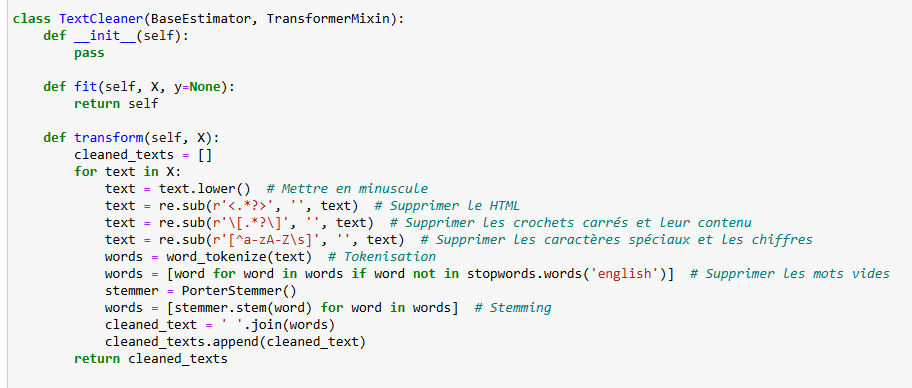


## Prétraitement et normalisation de texte

Avant de commencer l'ingénierie et la modélisation des caractéristiques, il est essentiel de nettoyer, pré-traiter et normaliser le texte. Cela permet de standardiser les éléments textuels, tels que les phrases et les mots, facilitant ainsi l'analyse des documents. Cette normalisation contribue à créer des caractéristiques pertinentes tout en réduisant le bruit, souvent causé par des éléments comme les symboles non pertinents, les caractères spéciaux, ou encore les balises HTML et XML.

Pour ce faire nous avons créer un pipeline qui permet de faire la normalisation du texte comprenant plusieurs étapes clés :

* **Nettoyage du texte** : Les balises HTML, qui n'apportent que peu de valeur lors de l'analyse, doivent être supprimées. La fonction strip\_html\_tags(…) est conçue pour cela.
* **Suppression des caractères accentués** : Étant donné que nous traitons des critiques en anglais, il est nécessaire de convertir les caractères accentués en caractères ASCII. Par exemple, le caractère é sera remplacé par e, grâce à la fonction remove\_accented\_chars(…).
* **Expansion des contractions** : Les contractions comme "don't" doivent être développées en "do not", ou "I'd" en "I would". La fonction expand\_contractions(…), utilisant le fichier contractions.py, accomplit cette tâche.
* **Suppression des caractères spéciaux** : Des expressions régulières peuvent être utilisées pour éliminer les caractères spéciaux. La fonction remove\_special\_characters(…) permet de faire ce nettoyage, avec la possibilité de conserver ou non les chiffres.
* **Élimination des mots vides** : Les mots courants tels que "a", "an", ou "the", qui n'ont que peu d'importance, doivent être supprimés. La fonction remove\_stopwords(…) est utilisée pour filtrer ces mots et conserver ceux qui portent un sens significatif.
* **Lemmatisation** : Cette technique consiste à réduire les mots à leur forme de base, en éliminant les affixes. La fonction lemmatize\_text(…) est utilisée pour garantir que les mots restent lexicographiquement corrects.
* **Création d'un normalisateur de texte** : Toutes ces fonctions sont intégrées dans la class TextCleaner, qui prend en entrée un corpus de documents et renvoie une version nettoyée et normalisée de ceux-ci.

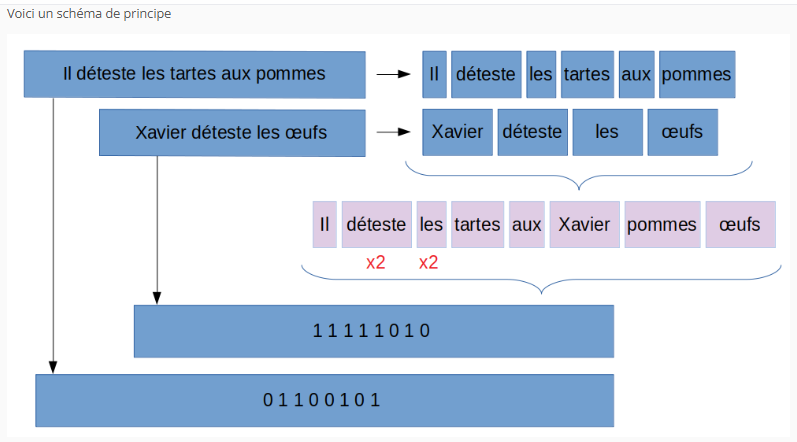


## Vectorisation du texte prétraité

Dans cette étape nous avons effectué l’incorporationde mots est une méthode qui permet de transformer le texte en vecteurs numériques. Les techniques utilisées sont les suivantes

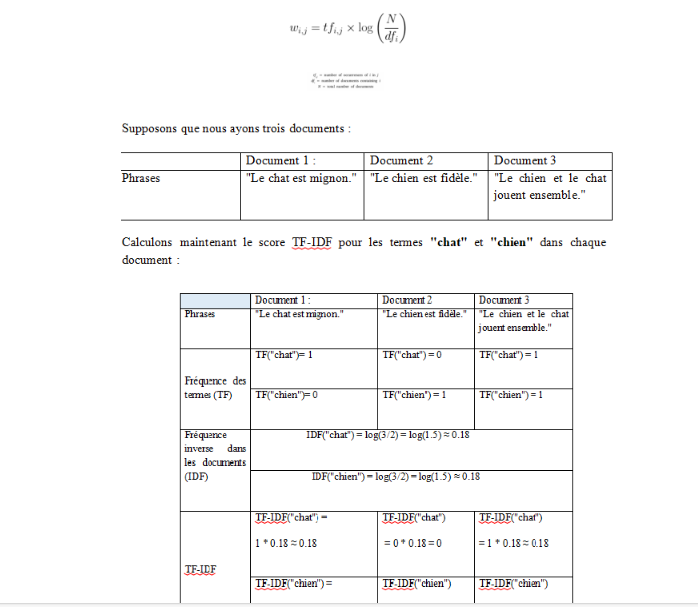
* **BoW : Sac de mots**

Le modèle Bag of Words (BoW) est l'une des méthodes les plus simples pour représenter du texte sous forme numérique. Comme son nom l'indique, il considère une phrase comme un ensemble de mots, qu'on peut représenter par un vecteur composé de nombres.



* **TF-IDF : Fréquence des termes - Fréquence inverse des documents**

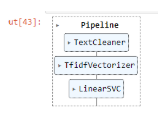
La méthode TF-IDF (Term Frequency-Inverse Document Frequency) est une mesure qui vise à évaluer l'importance d'un mot dans un document par rapport à l'ensemble d'un corpus. Elle combine la fréquence d'un mot dans un document avec son inverse de fréquence dans l'ensemble des documents, offrant ainsi une idée plus précise de sa pertinence.



## Conception, entraînement et évaluation des performances du model.

Dans cet étapes on a implémenter un pipeline qui permet de transformer et de modéliser les données de manière automatique et cohérente. Voici les composants principaux du pipeline :

* ****Prétraitement des données**** : Cette étape comprend le nettoyage des données (par exemple, la suppression des valeurs manquantes, le traitement des caractére spéciaux etc...), la transformation des variables (par exemple, la normalisation, la discrétisation, l'encodage), et la sélection des caractéristiques pertinentes.(comme on l’a ci-dessus)
* ****Modélisation**** : Une fois que les données sont préparées, le pipeline inclut l'étape où le modèle de machine learning est sélectionné et entraîné sur les données traitées. Cela peut inclure l'ajustement des hyperparamètres du modèle pour obtenir de meilleures performances.
* ****Évaluation**** : Après avoir entraîné le modèle, le pipeline évalue sa performance en utilisant des métriques appropriées pour la tâche (par exemple, précision, rappel, F1-score pour la classification).
* ****Déploiement**** : Dans certains cas, le pipeline peut également inclure des étapes pour déployer le modèle entraîné dans un environnement de production, prêt à être utilisé pour des prédictions en temps réel.

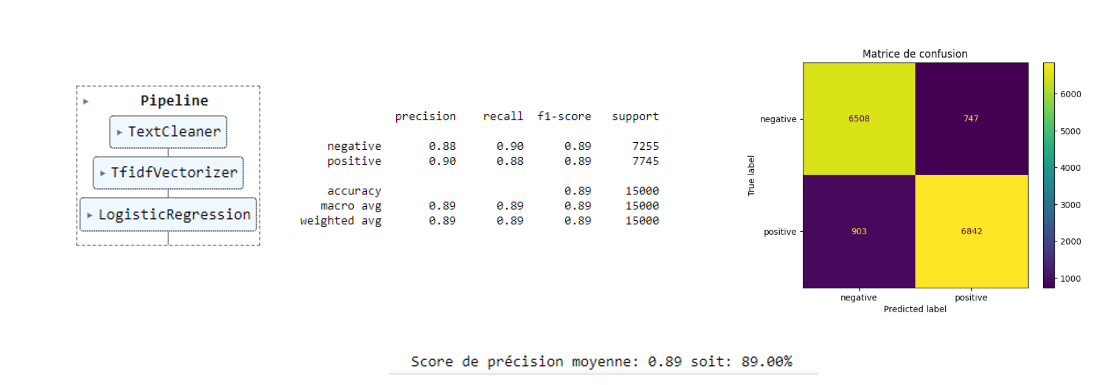


modele

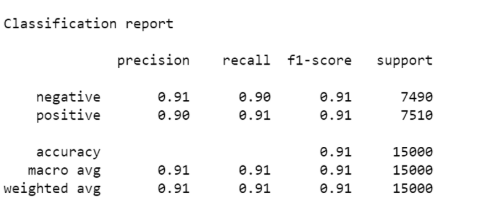
## Le Processus

1. ****La régression logistique****  est utilisée pour calculer la probabilité qu'un événement binaire se produise et pour traiter les problèmes de classification. Les images ci-dessous montrent la matrice de confusion et les mesures de performance du modèle.

****Avec utilisation du tf-idf****

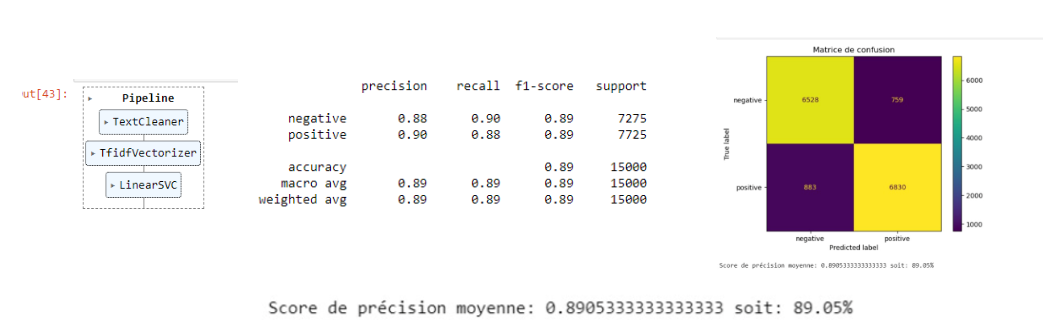


****Avec utilisation** BoW : Sac de mots**

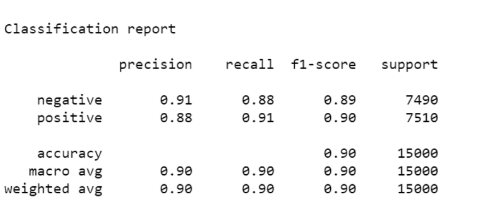


1. ****La descente du gradient stochastique (SGD)****  est un algorithme d'optimisation simple mais efficace utilisé pour trouver les valeurs des paramètres/coefficients des fonctions qui minimisent une fonction de coût. En d'autres termes, il est utilisé pour l'apprentissage discriminatif des classificateurs linéaires tels que SVM et la régression logistique. Les images ci-dessous montrent la matrice de confusion et les mesures de performance du modèle.

****Avec utilisation du tf-idf****



****Avec utilisation** BoW : Sac de mots**



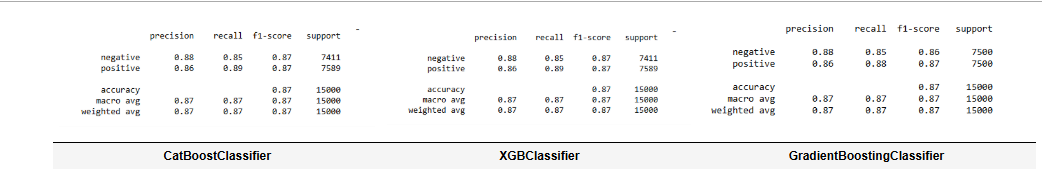
3 Autres algorithmes

****Avec utilisation du tf-idf****

**GradientBoostingClassifier** est un algorithme qui combine plusieurs arbres de décision faibles pour créer un modèle robuste. Il fonctionne en corrigeant les erreurs des arbres précédents à chaque étape et est souvent utilisé pour des tâches de classification.

**XGBClassifie**r est une version améliorée du Gradient Boosting, conçue pour être rapide et efficace. Il est particulièrement adapté aux grands ensembles de données et aux problèmes complexes.

**CatBoostClassifier** est un algorithme de classification basé sur le Gradient Boosting, optimisé pour les données contenant des variables catégoriques. Il est rapide, précis et facile à utiliser.



****Avec utilisation** BoW : Sac de mots**

## Choix du modèle le plus performant

Les images ci-dessus montrent la comparaison des mesures de performance de 5 modèles. Nous constatons que le modèle de régression logistique sur les fonctionnalités du modèle Bag of Word est le plus performant car il a une précision de 90,65 %. Mais on peut voir aussi que les performances des autres modèles sont également très proches de ce modèle.

# Conclusion

Cette analyse a montré l'efficacité des différents classificateurs dans notre étude. Les résultats obtenus avec les modèles comme CatBoost, XGBoost et SVC Linéaire indiquent qu'il existe une réelle variabilité dans les performances selon les techniques utilisées, que ce soit avec BoW ou TF-IDF.

Chaque modèle a ses forces, et il est clair que certains sont mieux adaptés à notre ensemble de données que d'autres. Par exemple, LogisticRegression a montré des résultats prometteurs, particulièrement en termes de précision et de rappel. Cela souligne l'importance de choisir le bon modèle en fonction des caractéristiques spécifiques des données.