

# Projet Machine Learning

## Analyse sentimentale de *reviews* Amazon

Guedj Odelia  
Marcoux Pépin Thomas  
Tounsi Mohamed

26 septembre 2019

- 1 Introduction
- 2 Preprocessing
- 3 Bag of words
- 4 Cross-validation et résultats
- 5 Conclusion

- 1 Introduction
- 2 Preprocessing
- 3 Bag of words
- 4 Cross-validation et résultats
- 5 Conclusion

Les données étudiées sont composées de commentaires (*reviews*) laissés par des clients Amazon et d'une note associée à leur commentaire, leur perception du produit.

Nous nous sommes alors posés la question suivante :

## Problématique

Est-il possible de déterminer la nature "sentimentale" d'un commentaire en se basant sur le vocabulaire employé par son auteur ? Si oui, quels sont les mots les plus susceptibles d'exprimer ce sentiment ?

- 1 Introduction
- 2 Preprocessing
- 3 Bag of words
- 4 Cross-validation et résultats
- 5 Conclusion

# Première approche des données

- Forme des données : fichier texte où chaque *review* est précédée de son label.  
On distingue une *review* d'une autre par un retour à la ligne.
- Exemple : premier commentaire

```
b'__label__2 Stuning even for the non-gamer: This sound track was beautiful  
! It paints the senery in your mind so well I would recomend it even to peo  
ple who hate vid. game music! I have played the game Chrono Cross but out o  
f all of the games I have ever played it has the best music! It backs away  
from crude keyboarding and takes a fresher step with grate guitars and soul  
ful orchestras. It would impress anyone who cares to listen! ^_^\n'
```

- On a donc label (1 ou 2) + review + retour à la ligne

# Étapes de preprocessing

- Etape 1 : Séparation des paragraphes.
- Etape 2 : Séparation des labels et reviews, recodage des labels en 0/1.
- Etape 3 : Encodage UTF-8
- Etape 4 : Mettre tout le texte en minuscules
- Etape 5 : Detection et suppression d'éventuelles URL
- Etape 6 : Remplacement des caractères spéciaux par des espaces
- Etape 7 : On transforme tous les accents éventuels par des lettres sans accents

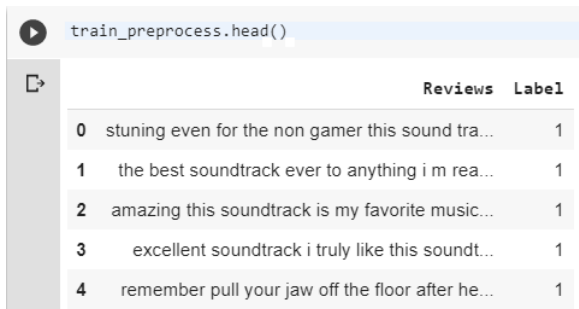
En appliquant l'ensemble des étapes de pre-processing au premier commentaire on obtient :

'stuning even for the non gamer this sound track was beautiful it paints the senery in your mind so well i would recomend it even to people who hate vid game music i have played the game chrono cross but out of all of the games i have ever played it has the best music it backs away from crude keyboarding and takes a fresher step with grate guitars and soulful orchestras it would impress anyone who cares to listen '



# Data frame final

Une idée de l'aspect du data frame propre :

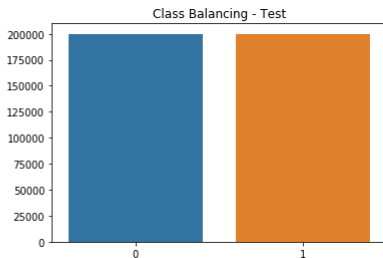
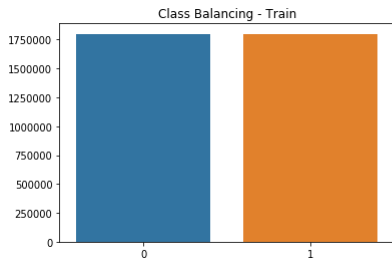


```
train_preprocess.head()
```

		Reviews	Label
0	stuning even for the non gamer this sound tra...		1
1	the best soundtrack ever to anything i m rea...		1
2	amazing this soundtrack is my favorite music...		1
3	excellent soundtrack i truly like this soundt...		1
4	remember pull your jaw off the floor after he...		1

# Répartition des différentes classes dans le jeu de données

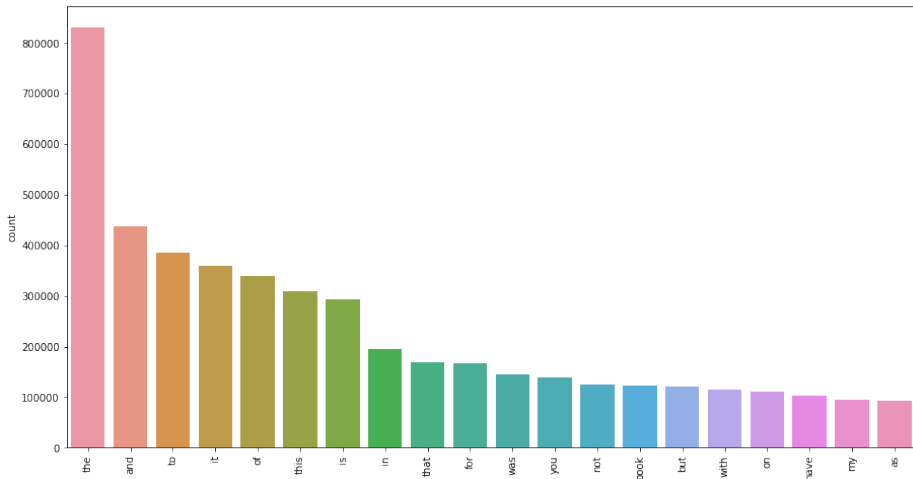
Avant de commencer la *vectorization* et l'entraînement des modèles, vérifions que les classes sont bien équilibrées dans le train et dans le test.



- 1 Introduction
- 2 Preprocessing
- 3 Bag of words**
- 4 Cross-validation et résultats
- 5 Conclusion

# Premier dictionnaire

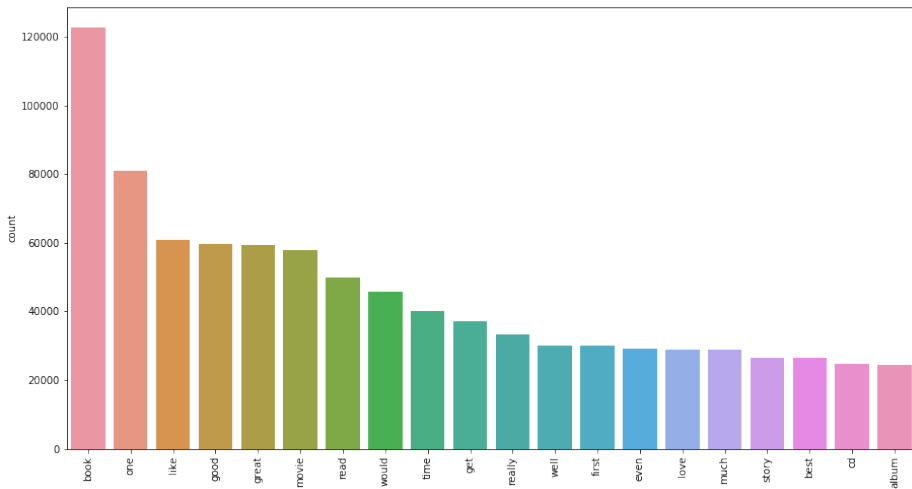
Taille du dictionnaire : 4530 mots



Mots dont la fréquence d'apparition est comprise entre 0.1% et 95%

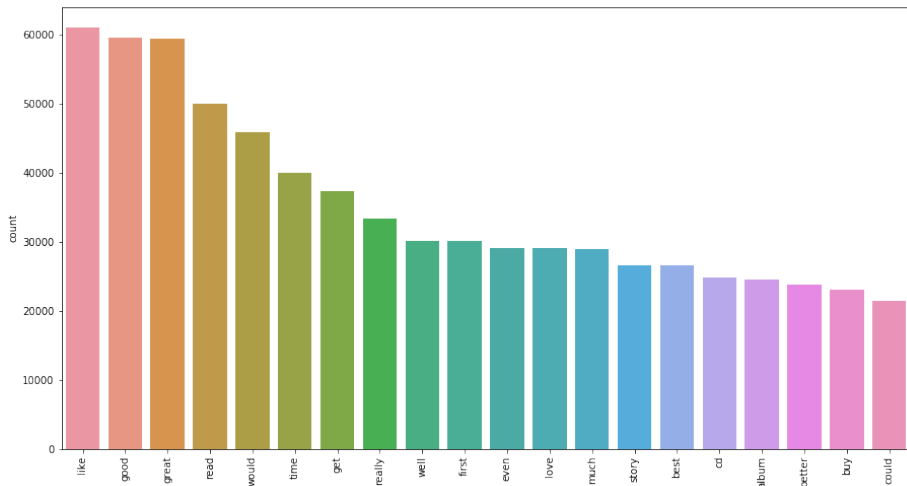
# Dictionnaire avec stopwords

Taille du dictionnaire : 4392 mots



# Dictionnaire avec stopwords personnalisés

Taille du dictionnaire : 4389 mots





- 1 Introduction
- 2 Preprocessing
- 3 Bag of words
- 4 Cross-validation et résultats**
- 5 Conclusion



# Apprentissage supervisé sur *count vectorizer*

## Première approche des algorithmes

Première approche: counts

- 1 - SVM 0.87994
- 2 - Logistic Regression 0.87871
- 3 - LDA /!\ 0.87094
- 4 - Naive Bayes 0.84340
- 5 - Random Forest 0.81944
- 6 - QDA /!\ 0.78336
- 7 - Decision Tree 0.76301

# Apprentissage supervisé sur TF-IDF

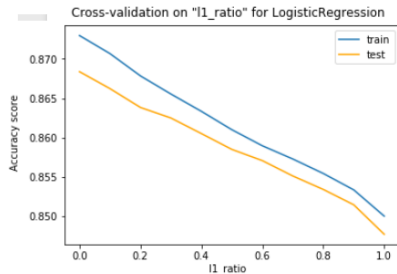
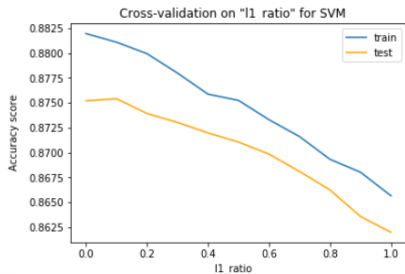
## Première approche des algorithmes

Première approche: tf-idf

- 1 - SVM 0.87559
- 2 - LDA /!\ 0.87149
- 3 - Logistic Regression 0.86783
- 4 - Random Forest 0.82024
- 5 - Naive Bayes 0.79084
- 6 - QDA /!\ 0.78531
- 7 - Decision Tree 0.75534

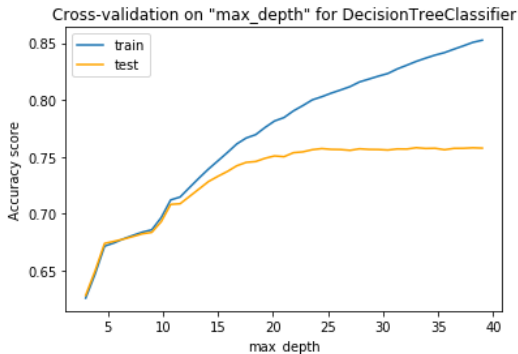
# Cross-validation

## SVM et Regression Logistique sur TF-IDF



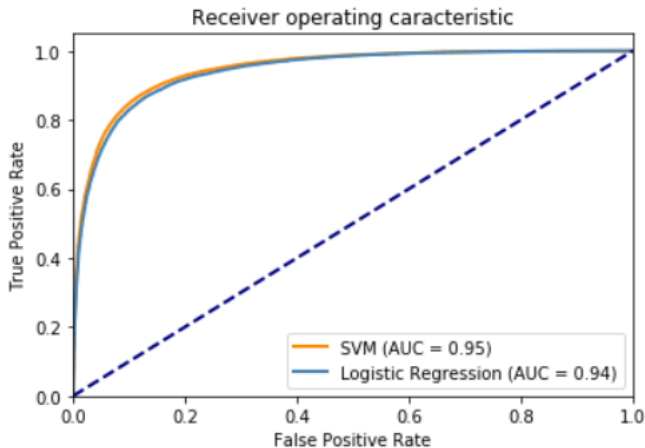
# Cross-validation

DecisionTreeClassifier sur TF-IDF



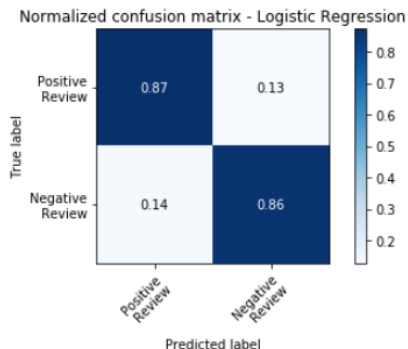
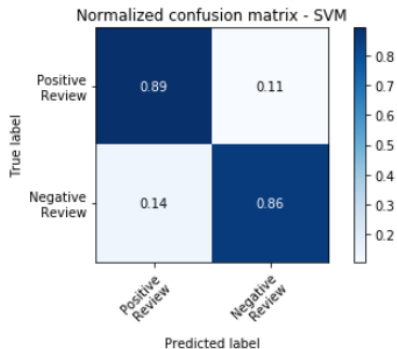
# Vérification des résultats

## Courbes ROC



# Vérification des résultats

## Matrices de confusion



- 1 Introduction
- 2 Preprocessing
- 3 Bag of words
- 4 Cross-validation et résultats
- 5 Conclusion**

### Problématique

Est-il possible de déterminer la nature "sentimentale" d'un commentaire en se basant sur le vocabulaire employé par son auteur ? Si oui, quels sont les mots les plus susceptibles d'exprimer ce sentiment ?

- Passer les noms au singulier
- Passer les verbes à l'infinitif
- Considérer les commentaires langue par langue, avec leur dictionnaire et *stopwords* associés
- Avoir la puissance nécessaire à entrainer les modèles sur l'entièreté des données



