# NLP TD n°1

## Partie 1 :

**Les a priori :** Tout d’abord j’ai déjà travaillé sur un sujet de Natural Langage Processing, c’était une compétition Kaggle sur un corpus de reviews provenant du site Goodreads.

Je sais donc à peu près, ce que je dois faire en pré-traitement de mes données et quels modèles utilisés.

Pré-traitements :

* Supprimer les stopwords (« le », « les », « ce », …) tous les mots qui ne contribuent pas au sens d’une phrase.
* Faire du Stemming changer tous les mots qui sont dans un même lexique par la racine de ceux-ci.
* Suppression de la ponctuation

Modèles :

* Residual Neural Network : réseau de neurone sont beaucoup utilisés dans des tâches de NLP, donc je pense l’utiliser
* Transformers : pareil que pour les RNN, ce sont des modèles qui sont les plus utilisés dans les tâches de traitements du langage.

**Partie train :**Ensuite pour la partie train, j’ai changé le retour de la fonction, j’utilise la library joblib pour créer un dump de mon model que je passe en paramètre dans ma commande.

Dans « make\_model » j’utilise le model avec ce même paramètre pour que ça soit plus dynamique.

Les modèles que nous avons utilisés et leurs résultats sont les suivants avec le pré-traitement que nous avons évoqué dans la partie a priori :

Logistic Regression : 91.19497487437187%

Random Forest Classifier : 91.29346733668342%

SVM : 91.49547738693468%

Multinomial Naive Bayes : 91.49396984924623%

XGBoost : 93.39597989949749%

Nous pouvons remarquer que le model XGBoost est le modèle avec la meilleure accuracy.

J’ai utilisé que des modèles faciles a implémenter avec la library sklearn c’est pour cela que je n’ai pas utilisé le modèle transformer.