# Compétition Kaggle 2 – IFT 6390 – Text Classification Challenge

## Auteurs : Maylis Heussner, Khadidja Yasmine Bourega & Alexis Raffier

## Matricules : \_, \_ & 20251348

Table des matières

[I. Introduction 2](#_Toc118026603)

[II. *Feature Design* 2](#_Toc118026604)

[III. Algorithmes utilisés 3](#_Toc118026605)

[1. Régression logistique 3](#_Toc118026606)

[2. Régression logistique avec données séparées 3](#_Toc118026607)

[3. SVM 4](#_Toc118026608)

[4. CNN – Resnet-18 4](#_Toc118026609)

[IV. Méthodologie 4](#_Toc118026610)

[V. Résultats 5](#_Toc118026611)

[1. Définition des hyperparamètres 5](#_Toc118026612)

[2. Comparaison des performances 7](#_Toc118026613)

[VI. Discussion 7](#_Toc118026614)

[VII. Références 8](#_Toc118026615)

# Introduction

Le but de cette compétition est de créer un classifieur de *Machine Learning* ayant la plus grande précision sur des textes courts (moins de 100 mots). Notre classifieur doit être capable de retourner l’humeur (positive, neutre ou négative) associée à chacun des textes présentés.

Pour ce problème, nous avons implémenté 4 modèles différents (indiqués ici en ordre croissant de performances) : un classifieur de Bayes (utilisant un sac de mots), un SVM (*Support Vector Machine*) avec des noyaux contenant des chaînes de caractères, un modèle de RNN (*Recurrent Neural Network*) - plus particulièrement un LSTM (*Long Short-Term Memory*) et enfin un modèle de *Transformer*.

Les entrées données étant du texte, il faut opérer quelques pré-traitements pour pouvoir ensuite les utiliser dans nos modèles de *Machine Learning*. Nous allons maintenant présenter les opérations qui ont été faites sur ces données, puis nous passerons aux différents modèles ainsi qu’à leurs résultats.

# *Feature Design*

Concernant les *features* utilisées pour cette compétition…

# Algorithmes utilisés

## Classifieur de Bayes naïf

Pour ce premier modèle…

## SVM

Pour le SVM…

## RNN (LSTM)

Le RNN…

## *Transformer*

Enfin, le modèle de Transformer…

# Méthodologie

La première chose à faire a été de séparer la base de données *train* (contenant 1 040 322 textes) en deux parties. Nous avons gardé 80% de la base pour l’entraînement de nos modèles. Les 20% restants servent à valider nos modèles, pour avoir une idée de leurs performances de généralisation sur des données qui n’ont pas encore été vues, ainsi que pour valider le choix des différents hyperparamètres.

Pour ces derniers, chaque modèle possède ses propres hyperparamètres…

# Résultats

## Définition des hyperparamètres

## Comparaison des performances

## 

Le tableau suivant présente les meilleurs résultats de précision obtenus sur l’ensemble de test (d’après les résultats donnés par Kaggle) ainsi que les hyperparamètres ayant permis d’atteindre ces valeurs :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modèle | Bayes naïf | SVM | LSTM | Transformer |
| Précision (Test) | 0.1756 | 0.1658 | 0.612 | 0.9922 |
| *Epochs* | 100 | 50 | - | 10 |
| *Learning Rate* | 0.25 | 0.5 | - | - |

On voit donc que le modèle le plus performant est le…

# Discussion

Pour cette compétition…

Il faudrait donc…

# Références

Image utilisée pour le rapport :

Provenance des codes utilisés pour la compétition :

* LSTM :
* Transformer :