Note méthodologique

I. Démarche de travail

A. Prétraitement du Test

Nous nettoyons le texte et nous le transformons :

Suppression:

- URL
- Balises HTML
- Références de personnages
- Caractères non imprimables
- Valeurs numériques

Traitement

- Lemmatisons le texte
- Conversion en minuscules.
- Suppression des caractères répétés dans les mots allongés,
- Suppression des mots vides
- Conservation des hashtags car ils peuvent fournir des informations précieuses sur ce projet particulier.

➢ Feature Engineering

Nous créons 10 colonnes qui sont :

- Nombre de phrases
- Nombre de mots
- Nombre de caractères
- Nombre de hashtags
- Nombre de mentions
- Nombre de mots tout en majuscules
- Longueur moyenne des mots
- Nombre de noms propres (PROPN)
- Nombre de noms non propres (NOM)
- Pourcentage de caractères qui sont de la ponctuation

B. Features extractions

Cette étape nous permet de convertir les documents en vecteur pour etre utilisé par les modèles pour la prédiction
Nous utilisons 2 méthodes principales :

- **TfidfVectorizer**
- **Doc2vec**

C. Modélisation : Résultats

1. Logistic Régression :

a. With TFIDFVectorizer

- Nous recherchons les meilleurs hyper paramètres avec GridSearchCV
- Accuracy : 0.7809139784946236

b. With Doc2vec

- Nous recherchons les meilleurs hyper paramètres avec GridSearchCV
- Accuracy : 0.6503311258278146

2. Transformers

a. **With TFIDFVectorizer**
Accuracy: 0.569

b. With Doc2vec
Accuracy: 0.558

- **Algorithme d'optimisation : Sparse catégorical Crossentropy**

Cette fonction basée sur la formule mathématique

$$J(w)=-1N\sum i=1N[yi\ log(y\hat{}i)+(1-yi)log(1-y\hat{}i)]$$

de catégorical_crossentropy.

Le sparse signifie juste que cette fonction attend des entiers en entrée et non des vecteurs (cas du catégorical_crossentropy).

Toutefois, comme notre sortie est binaire, la fonction binary crossentropy peux être utilisé mais dans ce cas avec la fonction d'activation « sigmoid »

- **Metric : Accuracy**

Calculé en divisant le nombre total de prédictions correct par le nombre de prédiction total

3. Pre entrained model nnlm-en-dim50

Accuracy: 0.7775

II. Limites et amélioration

Le meilleur model jusqu'ici est la régression logistique utilisé avec le TFIDFVectorizer. Mais ce model ne peux pas obtenir de bonnes précisions que les Transformers car n'est pas assez profond et ne pourra donc pas apprendre parfaitement sur de très grands ensembles de données.

Aussi, les modèles prés entrainés, bien qu'ils peuvent avoir de bonnes performances, auront du mal à comprendre et à avoir une bonne performance sur les bases de données un peu différentes : en exemple en lui donnant des tweets dans une autre langue.

En général, les performances peuvent être améliorés en augmentant le nombre d'epochs, ou en ayant encore plus de tweets donc une large base de données.