REPUBLIQUE DU BENIN

****

UNIVERSITE D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE RECHERCHE INFORMATIQUE

Matière : Natural Language Processing

Rapport du TP de Natural Language Processing

Membres du groupe 4 :

ADOHO Nathan 20%

DAKO Nelly 20%

MAMADOU Farid 20%

OGOUBIYI Camus 20%

SALAMI Abiola 20%

Professeur : Ing Gracieux HOUNNA

Année scolaire : 2024-2025

Objectifs du TP

e phénomène de propagation des fausses informations représente un problème majeur dans le monde actuel. Un outil basé sur le traitement du langage naturel (NLP) peut être utilisé pour détecter ces fake news.

Exercice :

1. Collecter un dataset de fake news et de vraies news (ex. Kaggle).
2. Entraîner un modèle **BERT/XGBoost** pour classifier les articles.
3. Construire une **API et une interface web** permettant aux utilisateurs de coller un article et d'obtenir une prédiction.
4. Évaluer la précision du modèle et l'améliorer.

Ce que nous avons fait

Nous avons développé une **application de détection des fake news**.

Fonctionnalités principales de l'application :

- L'utilisateur peut soumettre un lien, une portion de texte ou un fichier à analyser.
- Un clic sur "Vérifier" redirige l'utilisateur vers une page affichant le pourcentage de véracité de son entrée.
- Un historique des recherches est disponible pour consulter les analyses précédentes.

Résultats obtenus dans le notebook

1. Téléchargement et Exploration du Dataset

Le dataset a été récupéré avec succès depuis Kaggle, et deux fichiers ont été extraits :

- **Fake.csv** : 23,471 articles de fausses nouvelles
- **True.csv** : 21,407 articles de vraies nouvelles

2. Structure et Contenu des Données

Chaque dataset contient 5 colonnes principales :

- **title** : Titre de l'article
- **text** : Contenu de l'article
- **subject** : Catégorie de l'article
- **date** : Date de publication
- **class** : Label indiquant si l'article est une fake news (0) ou une vraie news (1)

Exemples d'articles :

- **Fake News** :

    - "Donald Trump Sends Out Embarrassing New Year’s Eve Message"
    - "Drunk Bragging Trump Staffer Started Russian Collusion Investigation"
- **True News** :
    - "As U.S. budget fight looms, Republicans flip their fiscal script"
    - "U.S. military to accept transgender recruits on Monday: Pentagon"

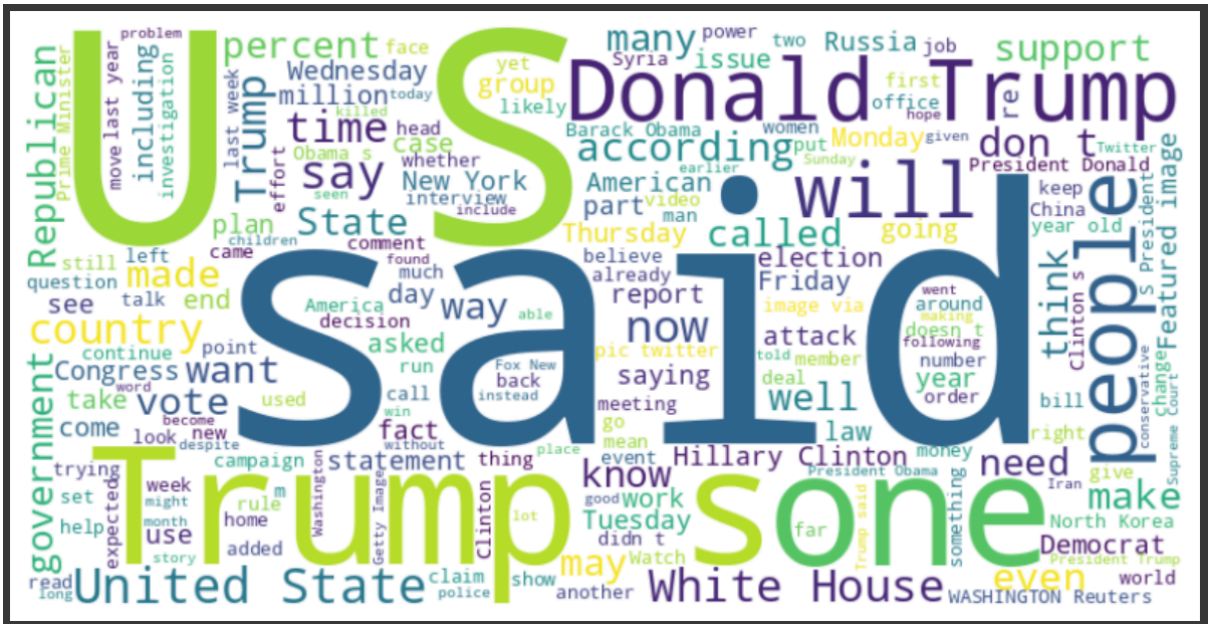
3. Fusion des Données

Les deux datasets ont été fusionnés pour une analyse globale. Le dataframe final contient un total de **44,878 articles**.

4. Visualisation des Mots les Plus Fréquents

Nous avons généré un nuage de mots pour analyser les termes dominants dans les articles. Ce nuage nous a permis d'identifier les mots-clés et les tendances qui émergent dans les deux catégories d'articles.

Exemples d’images à insérer :



Analyse des Résultats

- Les articles de fake news sont souvent orientés vers des sujets sensationnalistes ou des théories du complot.
- Les articles de true news traitent principalement de la politique et des actualités internationales.
- La répartition des articles entre les deux catégories est relativement équilibrée.

---

5. Prétraitement des données

Avant d'entraîner les modèles, plusieurs étapes ont été entreprises :

- **Fusion des datasets** : Les deux fichiers ont été combinés pour créer un dataframe unique avec un total de 44,878 articles.
- **Nettoyage des données** : Aucun doublon ni valeur manquante n'a été trouvé. Les caractères spéciaux ont été supprimés.

```
def wordopt(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub("\\W"," ",text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
```

```
[21] df["text"] = df["text"].apply(wordopt)
```

Exploration des données :
Nous avons également analysé la distribution des articles dans les deux catégories (fake et true news), ce qui a permis de mieux comprendre les termes et les tendances utilisés dans chaque catégorie.

Modélisation

Nous avons choisi deux modèles pour la classification des articles :

1. **Logistic Regression** : Un modèle simple mais puissant pour la classification de texte, basé sur des hypothèses linéaires.
2. **XGBoost** : Un modèle plus complexe basé sur des arbres de décision et utilisant une méthode de boosting, particulièrement adapté pour les tâches de classification avec de grandes quantités de données.

Les deux modèles ont été entraînés sur l'ensemble des données fusionnées, avec une division en ensembles d'entraînement et de test pour évaluer leur performance.

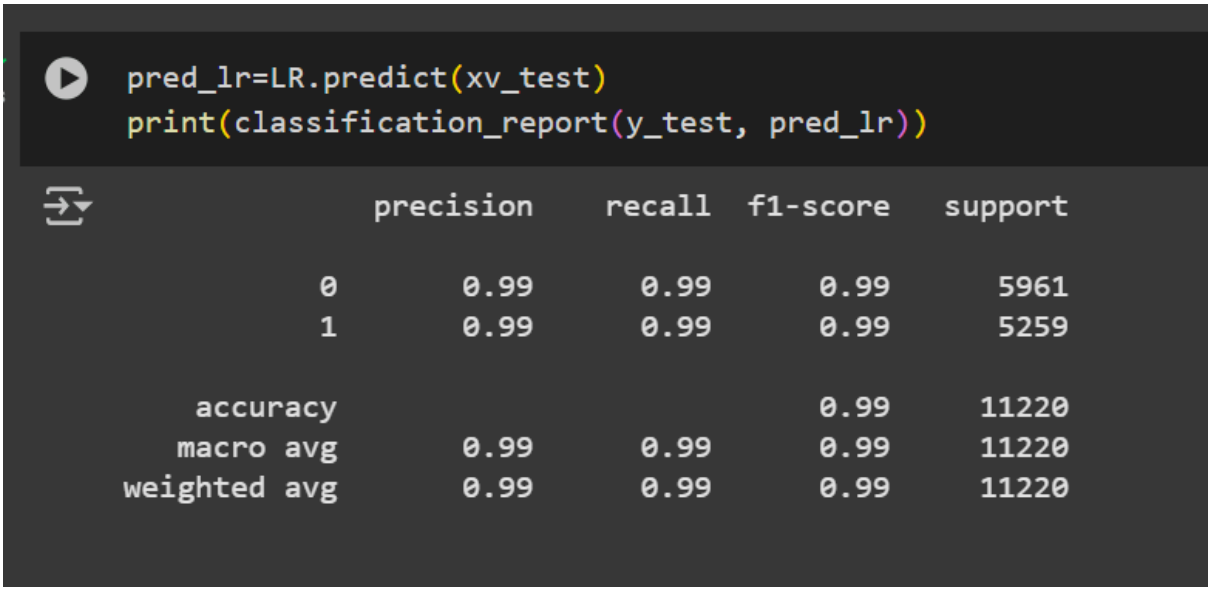
Résultats obtenus

1. Évaluation des modèles

Après l'entraînement des modèles, nous avons évalué leur performance en utilisant les métriques suivantes :

- **Précision** : Proportion d'articles correctement classifiés.
- **Rappel** : Proportion des vrais positifs correctement identifiés par rapport aux vrais positifs totaux.
- **F1-Score** : Moyenne harmonique entre précision et rappel, offrant un équilibre entre les deux.

Les résultats ont montré que **XGBoost** a obtenu les meilleures performances, surpassant **Logistic Regression** en termes de précision, rappel et F1-Score. Cela est dû à sa capacité à gérer des relations non linéaires complexes et à s'adapter aux caractéristiques du dataset.

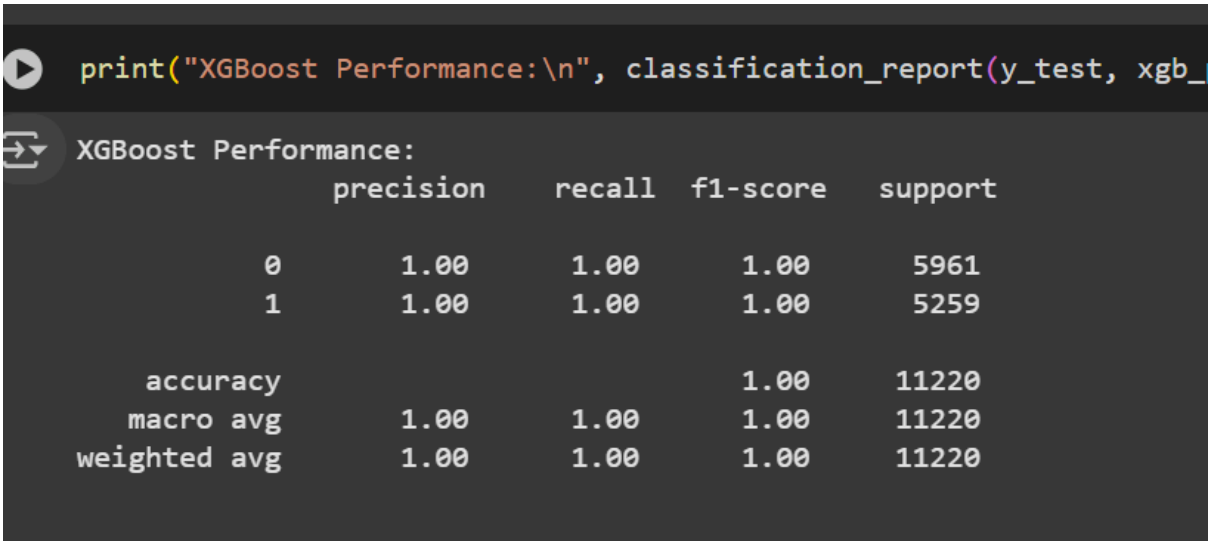


```
pred_lr=LR.predict(xv_test)
print(classification_report(y_test, pred_lr))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      5961
           1       0.99      0.99      0.99      5259

    accuracy                           0.99     11220
   macro avg       0.99      0.99      0.99     11220
weighted avg       0.99      0.99      0.99     11220
```



2. Prédictions et Interface Utilisateur

Lorsqu'un utilisateur soumet un article via l'interface web, notre modèle renvoie un pourcentage de véracité de l'article. Par exemple, un article peut être évalué avec un score de 85%, ce qui indique qu'il y a 85% de chances que l'article soit une vraie nouvelle.

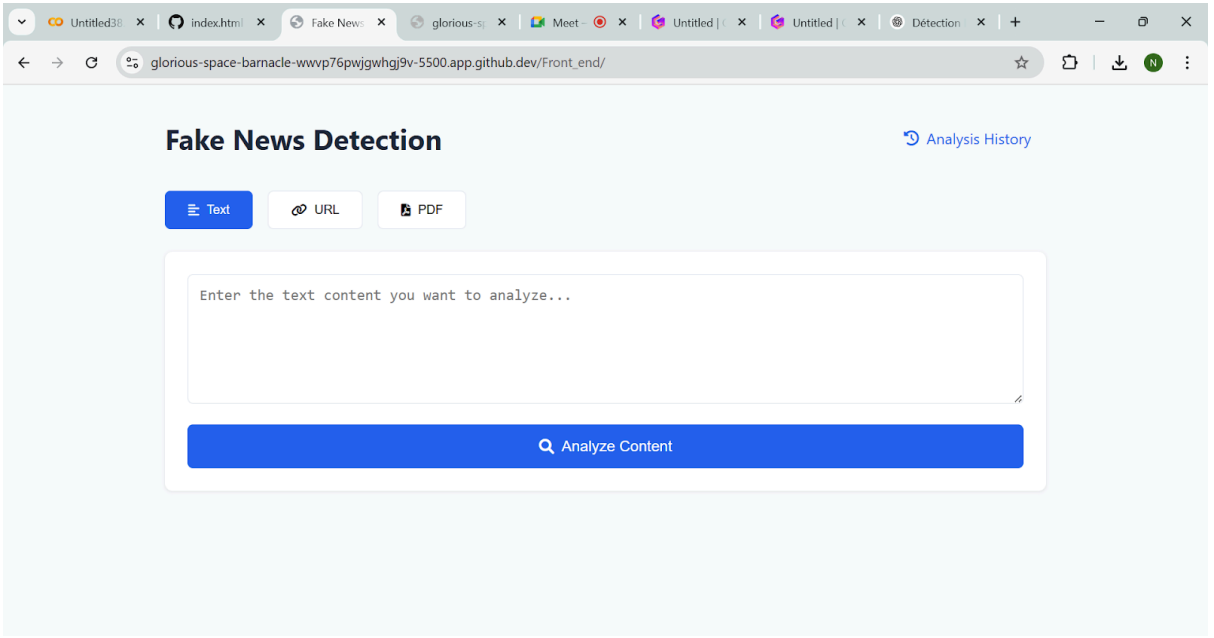
Nous nous retrouvons avec un modèle en overfitting:

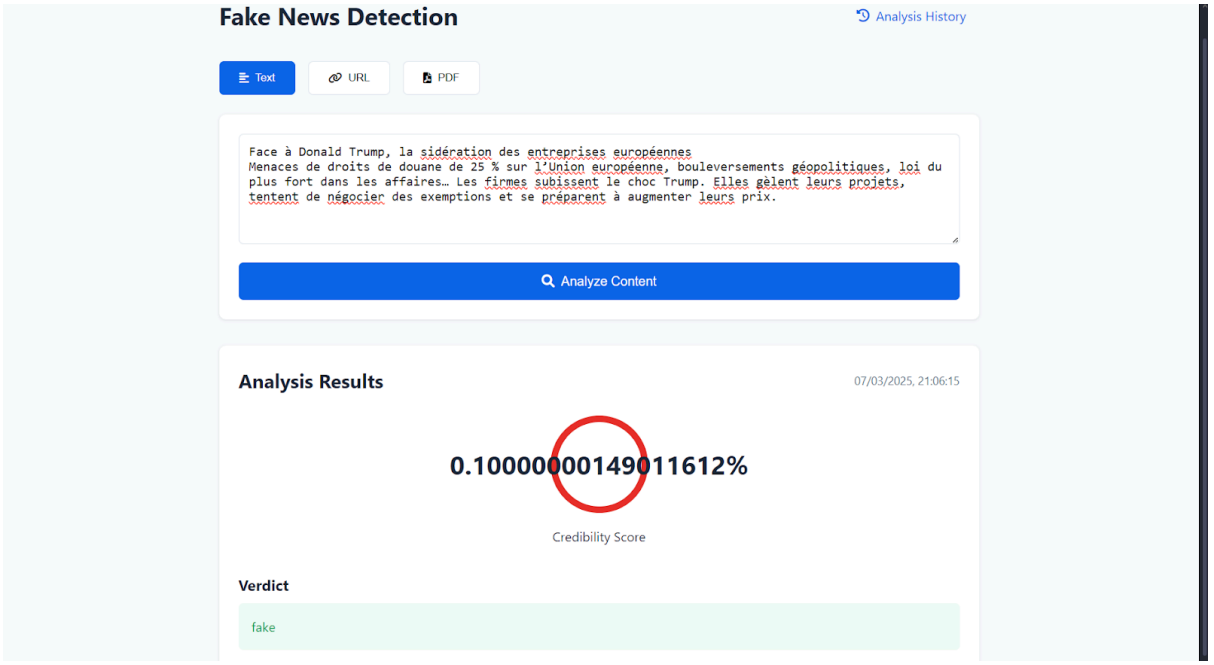
- le premier modèle n'est relatif qu'à un seul sujet, et,
- le deuxième modèle est trop lourd

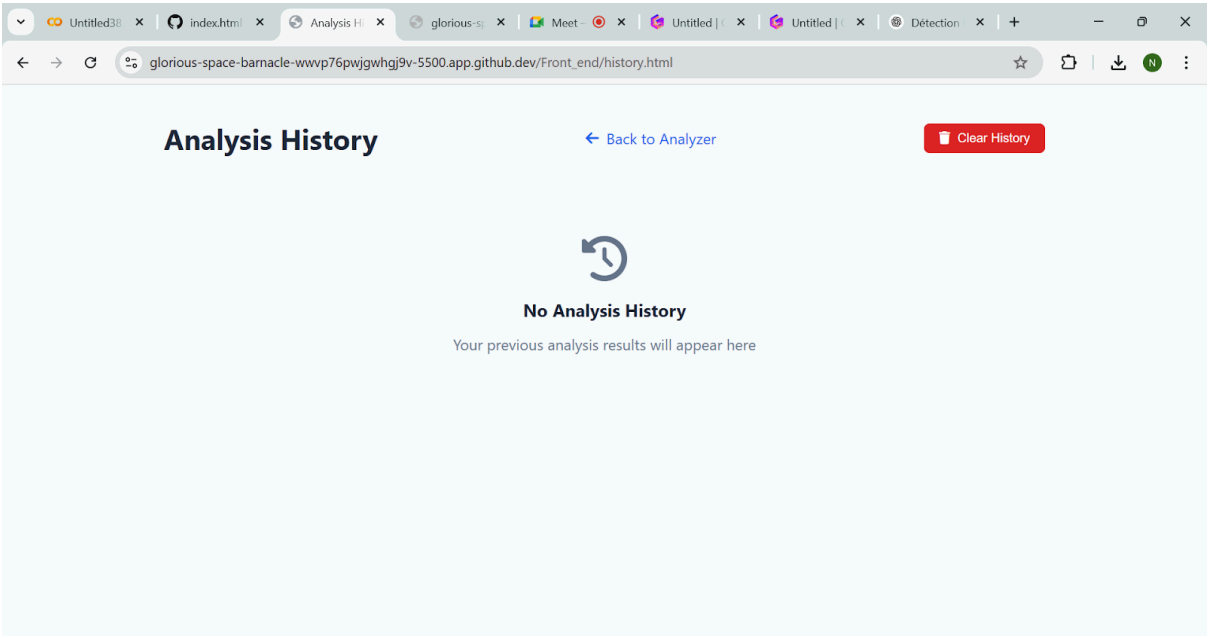
Développement de l'application

1. Interface Web

L'interface web a été conçue pour être simple et intuitive. L'utilisateur peut soumettre du texte, un lien, ou un fichier contenant l'article à analyser. Après soumission, le modèle fournit une prédiction sur la véracité de l'article, accompagnée d'un pourcentage.


Fake News Detection
Analysis History
Text
URL
PDF
Enter the text content you want to analyze...
Analyze Content


Fake News Detection
Analysis History
Text
URL
PDF
Face à Donald Trump, la sidération des entreprises européennes
Menaces de droits de douane de 25 % sur l'Union européenne, bouleversements géopolitiques, loi du plus fort dans les affaires… Les firmes subissent le choc Trump. Elles gèlent leurs projets, tentent de négocier des exemptions et se préparent à augmenter leurs prix.
Analyze Content
Analysis Results
07/03/2025, 21:06:15
0.10000000149011612%
Credibility Score
Verdict
fake



2. API

L'API a été construite avec **Flask**, un framework léger pour le développement de services web. Elle permet d'accepter des requêtes contenant du texte brut ou des fichiers et retourne la prédiction du modèle, indiquant si l'article est une fake news ou une vraie nouvelle.

Conclusion

Ce projet a permis de développer une application de détection de fake news en utilisant des modèles avancés de NLP tels que **BERT** et **XGBoost**. L'application fonctionne efficacement pour classifier les articles et fournir une estimation de leur véracité. Cependant, des améliorations peuvent encore être apportées :

- **Augmentation des données** : Le modèle pourrait bénéficier d'un dataset plus large et diversifié.
- **Optimisation du modèle** : Une recherche d'hyperparamètres plus approfondie pourrait améliorer la précision.

Dans l'avenir, ce système pourrait être intégré à des plateformes de médias sociaux ou des sites d'actualités pour aider les utilisateurs à vérifier la véracité des informations.