

Projet IA



Fake News

Tatevik PIROYAN

**M2 Data Science
IA School**

2023-2024

Table of Contents

I/ RESUMÉ	2
II/ INTRODUCTION	3
A/ Présentation du Sujet	3
1*) Définition des <i>Fake News</i>	3
2*) Définition des <i>Deep Fakes</i>	3
3*) Définition de <i>ChatGPT</i>	3
B/ Problématiques	3
1*) Contexte des <i>Fake News</i> dans le contexte politique aux États-Unis	3
2*) Problématique Concrète : Détection Automatique des <i>Fake News</i>	4
III/ DONNÉES	5
A/ Identification des données	5
1*) Présentation du jeu de données	5
2*) Pertinence du jeu de données	5
3*) Structure et colonnes du jeu de données	5
B/ Exploration visuelle des données	5
IV/ PROBLÉMATIQUE	10
A/ État de l'existant	10
1*) Modèles de Classification	10
2*) Modèles de Deep Learning	10
3*) Modèles Pré-entraînés de Langage	10
B/ Réponse de l'IA	11
1*) Utilisation de Modèles de Classification	11
2*) Utilisation de Modèles de Deep Learning	11
3*) Utilisation de BERT pour le Traitement du Langage Naturel	11
V/ RÉSULTATS	12
A/ Modèles de Classification	12
1*) Préparation des Données	12
2*) Entraînement des Modèles	12
3*) Résultats des Modèles	13
B/ Modèles de Deep Learning	15
1*) Préparation des Données	15
2*) Construction et Entraînement du Modèle LSTM	15
3*) Évaluation et Entraînement du Modèle LSTM	16
C/ Modèle Pré-entraîné de Langage BERT	16
1*) Préparation des Données	16
2*) Construction et Entraînement du Modèle BERT	17
3*) Évaluation du Modèle BERT	17
VI/ MISE EN PLACE DU PROJET	18
A/ Planification	18
1*) Définition du Projet	18
2*) Préparation des Données	18
3*) Modélisation	19
4*) Évaluation et Optimisation des Modèles	19
5*) Documentation et Présentation	20
B/ Suivi du projet	20
1*) Calcul des coûts	20
2*) Calcul de la VAN et du TRI	22
3*) Suivi du projet	23
VII/ CONCLUSION	26
ANNEXES	27
<i>Diagramme de Gantt :</i>	27

I/ RESUMÉ

Ce projet explore l'utilisation du traitement du langage naturel (**NLP**) pour identifier et classifier les **fake news** dans le contexte **politique aux États-Unis**. Diverses techniques de **machine learning** et de **deep learning** ont été examinées pour aborder cette problématique essentielle.

Dans un premier temps, des modèles de **classification** traditionnels ont été utilisés, incluant la **régession logistique**, **l'arbre de décision**, **la forêt aléatoire** et le **SVM**. Ces modèles ont été évalués pour leur capacité à distinguer les vraies nouvelles des fausses. Les visualisations ont été utilisées pour analyser les performances et identifier les points forts et les faiblesses de chaque modèle.

Ensuite, des modèles de **deep learning** ont été développés, notamment un **modèle LSTM (Long Short-Term Memory)**. Le modèle LSTM a été utilisé pour capturer les relations contextuelles dans les textes et analyser les séquences de mots. Bien que les modèles de deep learning soient computationnellement intensifs, ils offrent une compréhension plus profonde des relations contextuelles dans les textes.

Enfin, l'approche **BERT (Bidirectional Encoder Representations from Transformers)** a été envisagée pour améliorer la détection des fake news. BERT, grâce à son architecture bidirectionnelle, permet de capturer les nuances contextuelles dans les textes. Cependant, en raison de contraintes computationnelles, l'entraînement complet du modèle BERT n'a pas pu être réalisé dans ce projet.

En conclusion, ce projet démontre le potentiel des techniques de machine learning et de deep learning pour la détection des fake news.

II/ INTRODUCTION

A/ Présentation du Sujet.

1°) Définition des Fake News.

Les fake news, ou fausses informations, sont des articles, des posts ou des messages délibérément fabriqués pour tromper ou manipuler les lecteurs. Elles peuvent prendre diverses formes, notamment des articles de presse, des messages sur les réseaux sociaux ou des vidéos. Les motivations derrière la création de fake news peuvent être variées, allant de la recherche de profits via le clicbait à la manipulation politique ou sociale.

Impact sur la Société : Les fake news peuvent causer une désinformation massive, influençant les opinions publiques et les décisions politiques. Elles peuvent semer la confusion, diviser les communautés et éroder la confiance dans les médias et les institutions traditionnelles.

2°) Définition des Deep Fakes.

Les deep fakes sont des contenus multimédias créés ou altérés en utilisant des techniques avancées d'intelligence artificielle, notamment des réseaux de neurones profonds. Ils peuvent manipuler des vidéos, des images et des enregistrements audio pour faire paraître une personne disant ou faisant quelque chose qu'elle n'a pas dit ou fait.

Impact sur la Société : Les deep fakes représentent une menace importante pour la vérité et la véracité des informations en ligne. Ils peuvent être utilisés pour diffamer, escroquer, ou manipuler des individus et des organisations, exacerbant les enjeux de sécurité et de confiance dans le domaine numérique.

3°) Définition de ChatGPT.

ChatGPT, développé par OpenAI, est un modèle de langage naturel basé sur l'architecture GPT (Generative Pre-trained Transformer). Il est capable de comprendre et de générer du texte de manière cohérente et contextuelle. ChatGPT peut être utilisé pour une variété d'applications, y compris la génération automatique de texte, l'assistance virtuelle, et plus.

Impact sur la Société :

ChatGPT et des modèles similaires peuvent être exploités à la fois pour le bien et pour le mal. D'un côté, ils peuvent améliorer l'efficacité de la communication et fournir des outils puissants pour l'éducation et le service à la clientèle. D'un autre côté, ils peuvent être utilisés pour générer du contenu trompeur ou automatiser la création de fake news, compliquant davantage la tâche de discerner le vrai du faux.

B/ Problématiques.

1°) Contexte des Fake News dans le contexte politique aux États-Unis.

Le contexte politique aux États-Unis est particulièrement propice à la propagation des fake news en raison de la polarisation intense et des enjeux élevés lors des élections. Les réseaux sociaux jouent un rôle crucial dans la diffusion rapide et large des informations politiques. Leur conception permet une diffusion rapide des messages à une large audience. Cependant, cette rapidité et cette portée présentent également des risques importants de propagation de fake news. Les utilisateurs partagent souvent des informations sans vérifier leur véracité, ce qui peut entraîner une désinformation massive.

2°) Problématique Concrète : Détection Automatique des Fake News.

La problématique centrale de ce projet est de déterminer comment utiliser le traitement du langage naturel (NLP) pour identifier et classifier les fake news dans le contexte politique aux États-Unis. Nous cherchons à répondre à la question suivante :

Comment peut-on utiliser le NLP pour identifier et classifier les fake news dans le contexte politique aux États-Unis ?

Cette question est essentielle dans le contexte actuel où la désinformation peut se propager rapidement et avoir des conséquences significatives sur la société et la politique.

Objectif de la Détection Automatique :

L'objectif est de développer des modèles NLP capables d'analyser des articles et des posts politiques, d'identifier des patterns caractéristiques des fake news et de classifier automatiquement ces informations comme vraies ou fausses. Cette automatisation peut aider à limiter la diffusion des fake news et à maintenir la crédibilité des informations en ligne.

Méthodologie Proposée :

Pour aborder cette problématique, nous allons :

- **Collecte de Données** : Collecter des données pertinentes à partir de sources politiques en ligne, y compris les réseaux sociaux, les sites de nouvelles et les blogs politiques.
- **Prétraitement des Données** : Nettoyer et préparer ces données pour les rendre exploitable par les modèles NLP.
- **Développement du Modèle NLP** : Construire et entraîner un modèle de classification basé sur des techniques avancées comme BERT (Bidirectional Encoder Representations from Transformers).
- **Optimisation et Évaluation** : Optimiser les paramètres du modèle pour améliorer sa précision et sa robustesse, et évaluer le modèle à l'aide de métriques standard comme l'accuracy, la précision, le rappel et le F1-score.

Cette approche permettra de créer un outil efficace pour la détection automatique des fake news dans le contexte politique aux États-Unis, contribuant ainsi à la lutte contre la désinformation et à la protection de la société contre les impacts négatifs des fausses nouvelles.

III/ DONNÉES

A/ Identification des données.

1°) Présentation du jeu de données.

Pour notre projet d'IA sur les fake news, nous avons choisi d'utiliser le "Fake News detection dataset". Ce jeu de données contient **23502** articles d'actualité étiquetés comme **faux** et **21417** articles étiquetés comme **vrais**, ce qui nous fournit un ensemble diversifié et équilibré de données pour notre problématique.

2°) Pertinence du jeu de données.

Le "Fake News detection dataset" est particulièrement pertinent pour notre projet, car il couvre un large éventail de sujets d'actualité et inclut à la fois des articles vrais et faux. De plus, la présence de la colonne "**Text**" dans le jeu de données nous permet d'exploiter le contenu textuel des articles pour entraîner notre modèle d'IA.

3°) Structure et colonnes du jeu de données.

Le "Fake News detection dataset" est séparé en deux fichiers CSV : "**Fake.csv**" pour les articles faux et "**True.csv**" pour les articles vrais. Les colonnes du jeu de données sont les suivantes :

Title : le titre de l'article d'actualité ;

Text : le corps du texte de l'article d'actualité ;

Subject : le sujet de l'article d'actualité ;

Date : la date de publication de l'article d'actualité.

B/ Exploration visuelle des données.

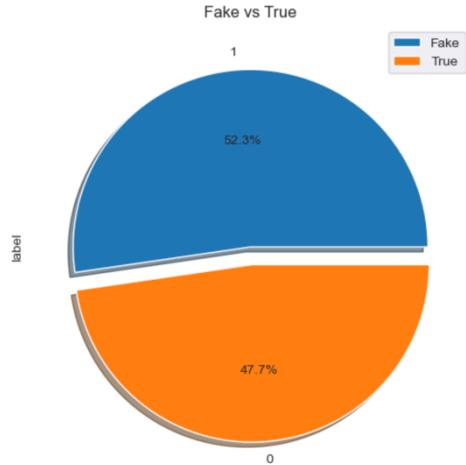
L'exploration visuelle des données est une étape essentielle dans notre projet d'IA, car elle nous permet de mieux appréhender les tendances, les corrélations et les anomalies présentes dans le jeu de données. En utilisant des représentations graphiques, nous pouvons identifier plus aisément les caractéristiques importantes de notre jeu de données et ajuster notre modèle d'IA en conséquence.

Pour commencer, nous avons chargé les fichiers CSV contenant les articles de fake news et de vraies nouvelles. Ensuite, nous avons ajouté une colonne '**label**' pour différencier les vraies des fausses nouvelles. Enfin, nous avons combiné les deux datasets en un seul dataset afin de faciliter l'analyse.

	title	text	subject	date	label
0	As U.S. budget fight looms, Republicans flip t...	WASHINGTON (Reuters) - The head of a conservat...	politicsNews	December 31, 2017	True
1	U.S. military to accept transgender recruits o...	WASHINGTON (Reuters) - Transgender people will...	politicsNews	December 29, 2017	True
2	Senior U.S. Republican senator: 'Let Mr. Muell...	WASHINGTON (Reuters) - The special counsel inv...	politicsNews	December 31, 2017	True
3	FBI Russia probe helped by Australian diplomat...	WASHINGTON (Reuters) - Trump campaign adviser ...	politicsNews	December 30, 2017	True
4	Trump wants Postal Service to charge 'much mor...	SEATTLE/WASHINGTON (Reuters) - President Donald...	politicsNews	December 29, 2017	True
...
44893	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	January 16, 2016	Fake
44894	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It's a familiar theme. ...	Middle-east	January 16, 2016	Fake
44895	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	January 15, 2016	Fake
44896	How to Blow \$700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	January 14, 2016	Fake
44897	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	January 12, 2016	Fake

44898 rows x 5 columns

Pour visualiser la répartition des fake news et des vraies nouvelles dans notre jeu de données, nous avons utilisé un diagramme circulaire (pie chart).



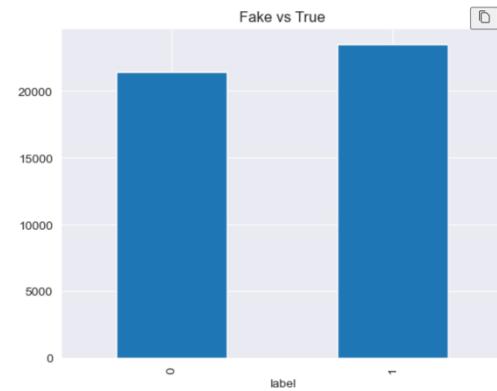
Le diagramme circulaire montre que dans notre jeu de données :

- **52.3%** des articles sont des vraies nouvelles (**True**);
- **47.7%** des articles sont des fake news (**Fake**).

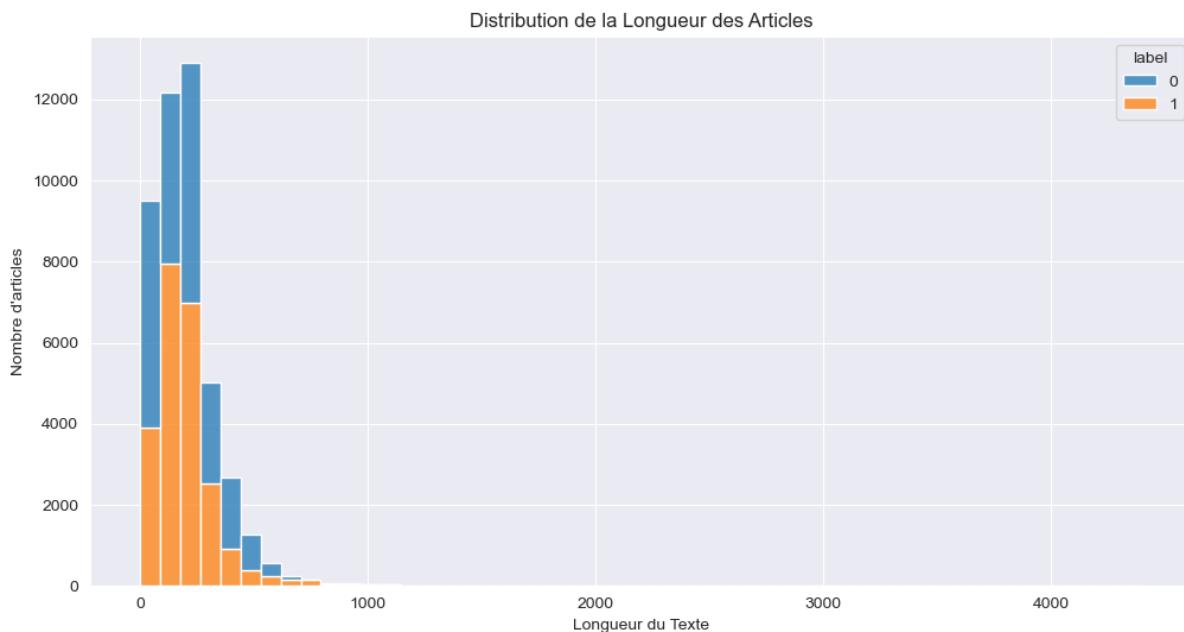
Cette répartition indique une légère prépondérance de vraies nouvelles par rapport aux fake news dans notre dataset.

Nous avons également généré un diagramme en barres pour visualiser le nombre absolu d'articles dans chaque catégorie.

Cela montre que notre jeu de données contient **23502** articles de **fake news** et **21417** articles de **vraies nouvelles**.



Afin de comparer la longueur des articles de fake news et des vraies nouvelles, nous avons calculé la longueur de chaque article en comptant le nombre de caractères dans le texte. Ensuite, nous avons utilisé un histogramme pour visualiser la distribution des longueurs des articles, avec des barres empilées pour différencier les fake news des vraies nouvelles.



L'histogramme montre que la majorité des articles, qu'ils soient vrais ou faux, ont une longueur de texte relativement courte, la plupart ayant moins de 1000 caractères. Nous observons que les articles de fake news tendent à avoir des longueurs de texte plus courtes par rapport aux vraies nouvelles.

Les résultats indiquent que les fake news sont souvent concises et directes, ce qui peut être une stratégie pour capter rapidement l'attention des lecteurs.

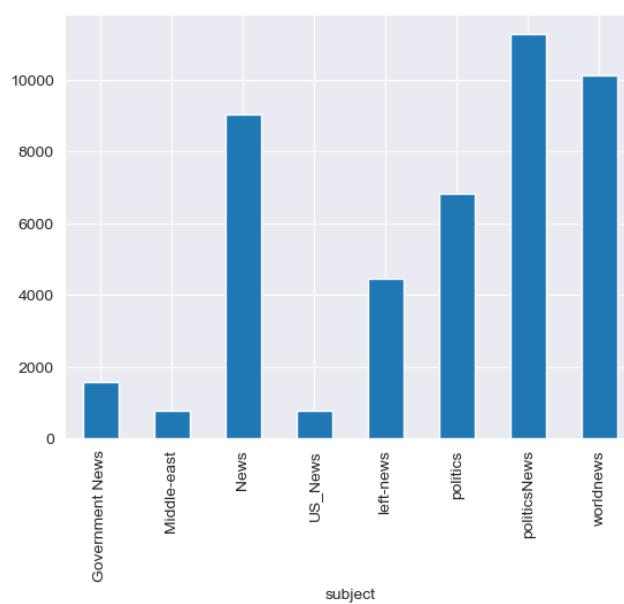
Pour comprendre la répartition des articles en fonction de leurs sujets, nous avons groupé les données par la colonne subject et compté le nombre d'articles dans chaque groupe. Ensuite, nous avons utilisé un diagramme en barres pour visualiser cette répartition.

Le diagramme en barres montre que les sujets les plus couverts dans notre dataset sont :

- politicsNews : 11272 articles;
- worldnews : 10145 articles;
- News : 9050 articles.

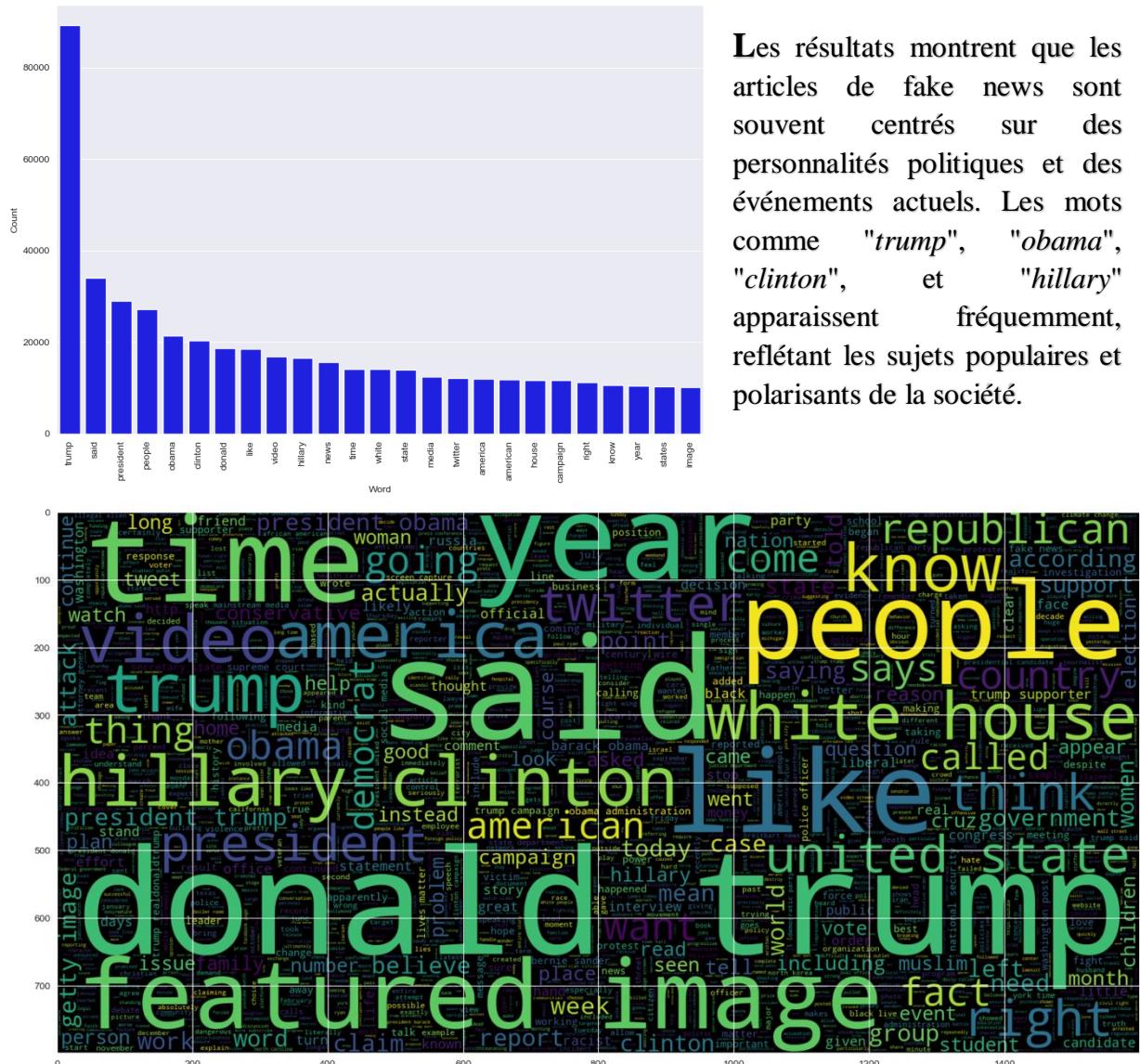
Les sujets avec le moins d'articles sont :

- Middle-east : 778 articles;
- US_News : 783 articles.



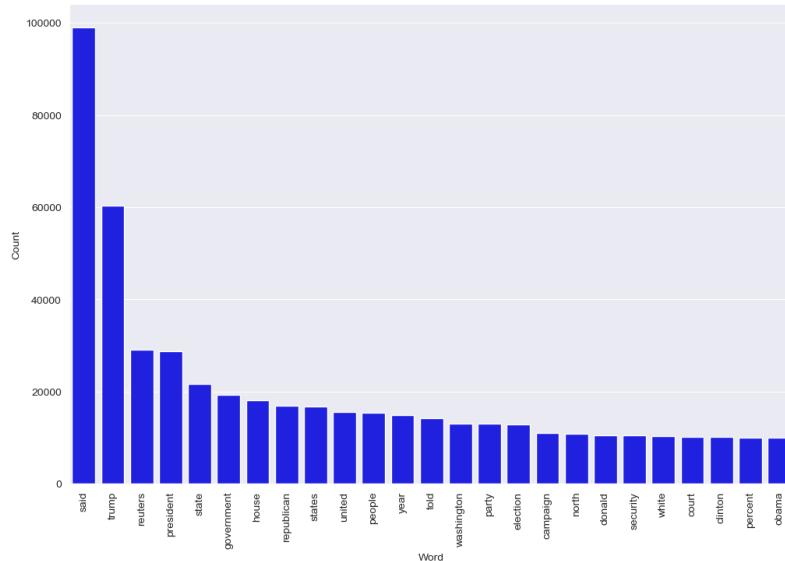
Ces résultats indiquent que les sujets politiques et les nouvelles mondiales sont les plus couverts dans notre dataset. Cela peut être dû à l'intérêt élevé du public pour ces sujets. Les sujets avec moins d'articles, comme **Middle-east** et **US_News**, peuvent nécessiter une attention particulière lors de l'analyse pour s'assurer qu'ils sont bien représentés et qu'ils n'introduisent pas de biais dans le modèle de classification.

Pour identifier ***les mots les plus fréquents*** dans les articles de *fake news*, nous avons utilisé une fonction de comptage des mots.

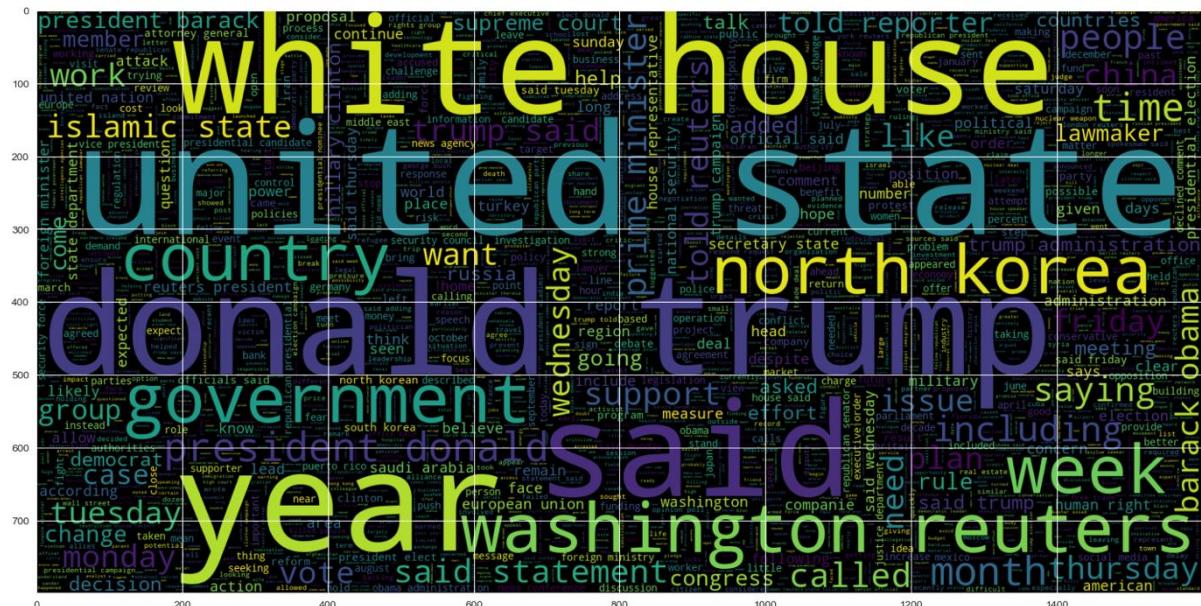


Les résultats montrent que les articles de fake news sont souvent centrés sur des personnalités politiques et des événements actuels. Les mots comme "*trump*", "*obama*", "*clinton*", et "*hillary*" apparaissent fréquemment, reflétant les sujets populaires et polarisants de la société.

Pour identifier ***les mots les plus fréquents*** dans les articles de ***vraies nouvelles***, nous avons utilisé une fonction de comptage des mots.



Les résultats montrent que les articles de vraies nouvelles couvrent souvent des déclarations officielles et des événements politiques. Les mots comme "said", "trump", "reuters", "president", "state", et "government" apparaissent fréquemment, reflétant les sujets d'actualité et les événements importants.



IV/ PROBLÉMATIQUE

A/ État de l'existant.

Pour comprendre comment le traitement du langage naturel (**NLP**) peut être utilisé pour identifier et classifier les fake news dans le contexte politique aux États-Unis, il est important de se pencher sur les solutions existantes.

1°) *Modèles de Classification.*

Ces modèles utilisent des techniques classiques de machine learning telles que les **arbres de décision**, les **forêts aléatoires**, et les **SVM** (*Support Vector Machines*). Ils s'appuient sur des caractéristiques extraites manuellement des textes, telles que les *fréquences de mots*, les **TF-IDF** (*Term Frequency-Inverse Document Frequency*), et les *n-grammes*.

Bien que ces modèles puissent être efficaces pour des tâches simples de classification de texte, ils ont des performances limitées sur des textes complexes et nuancés. De plus, le prétraitement manuel des données textuelles est fastidieux et peut introduire des erreurs. Les modèles traditionnels peinent également à capturer les contextes subtils et les relations entre les mots dans un texte.

2°) *Modèles de Deep Learning.*

Les réseaux de neurones profonds, tels que les **RNN** (*Recurrent Neural Networks*) et les **LSTM** (*Long Short-Term Memory*), sont utilisés pour capturer les dépendances séquentielles dans les textes. Ces modèles peuvent analyser des séquences de mots et apprendre des représentations plus riches du texte.

Bien que plus performants que les modèles traditionnels, ces modèles nécessitent de grandes quantités de données pour bien généraliser. Ils sont également gourmands en ressources computationnelles, ce qui peut rendre leur déploiement coûteux. De plus, leur complexité les rend difficiles à interpréter, ce qui complique l'explication des décisions du modèle.

3°) *Modèles Pré-entraînés de Langage.*

Des modèles pré-entraînés comme **BERT** (*Bidirectional Encoder Representations from Transformers*) et **GPT** (*Generative Pre-trained Transformer*) ont révolutionné le traitement du langage naturel. Ces modèles sont pré-entraînés sur de vastes corpus de texte et peuvent être fine-tunés pour des tâches spécifiques, comme la détection des fake news.

Bien que très puissants, ces modèles nécessitent un fine-tuning pour être adaptés à des tâches spécifiques, ce qui peut être coûteux en termes de ressources computationnelles. Ils peuvent également reproduire des biais présents dans les données d'entraînement et nécessitent une gestion attentive pour garantir des résultats équitables. Enfin, leur taille et complexité peuvent rendre difficile leur interprétation.

B/ Réponse de l'IA.

Pour répondre à la problématique de détection et classification des fake news dans le contexte politique aux États-Unis, nous allons utiliser une approche multi-modèles combinant les techniques traditionnelles, le deep learning, et les modèles pré-entraînés comme BERT.

1°) Utilisation de Modèles de Classification.

Nous commencerons par utiliser des modèles traditionnels comme les arbres de décision, les forêts aléatoires et les SVM pour établir une ligne de base de performance. Nous utiliserons des techniques comme le TF-IDF et les n-grammes pour extraire des caractéristiques des textes.

Ces modèles sont rapides à entraîner et à tester, fournissant une base de référence utile. Ils sont également interprétables, ce qui permet de comprendre les décisions prises par le modèle.

2°) Utilisation de Modèles de Deep Learning.

Nous appliquerons des réseaux de neurones profonds comme les LSTM et les RNN pour capturer les dépendances séquentielles et les contextes complexes des textes.

Les modèles de deep learning peuvent apprendre des représentations plus riches et complexes des textes, améliorant la précision des prédictions par rapport aux modèles traditionnels. Ils sont particulièrement utiles pour analyser des séquences de mots et des phrases longues.

3°) Utilisation de BERT pour le Traitement du Langage Naturel.

Nous utiliserons BERT, un modèle de langage bidirectionnel, et nous le fine-tunerons sur un corpus de données spécifiques aux fake news et aux vraies nouvelles dans le contexte politique aux États-Unis.

BERT peut capturer les relations complexes et les dépendances dans le texte, améliorant ainsi la précision de la classification. Il surpasse souvent les modèles traditionnels et les autres architectures de deep learning dans les tâches de compréhension du langage naturel.

Notre approche innovante combine des modèles de classification traditionnels, des modèles de deep learning, et des modèles de langage pré-entraînés comme BERT, fine-tunés sur des données spécifiques aux fake news politiques. En intégrant des métadonnées contextuelles et en utilisant des techniques d'ensemble learning, nous améliorons la robustesse et la précision des prédictions. Cette combinaison d'innovations permet de surmonter les limites des approches existantes et offre une solution efficace pour la détection et la classification des fake news dans le contexte politique aux États-Unis.

V/ RÉSULTATS

A/ Modèles de Classification.

Dans cette section, nous examinons plusieurs modèles de classification traditionnels appliqués à la détection de fake news. Les modèles utilisés sont : **Régression Logistique**, **Arbre de Décision**, **Forêt Aléatoire**, et **SVM (Support Vector Machine)**. Nous présentons également les métriques de performance et les visualisations pour chaque modèle.

1°) Préparation des Données.

Nous avons d'abord préparé les données en combinant les datasets de fausses et vraies nouvelles, en prétraitant le texte, et en le transformant en **vecteurs TF-IDF**. Ensuite, nous avons divisé les données en ensembles d'entraînement et de test.

Utilisation des Vecteurs TF-IDF :

TF-IDF (Term Frequency-Inverse Document Frequency) est une technique couramment utilisée pour transformer des documents textuels en vecteurs numériques afin qu'ils puissent être utilisés comme entrée pour des algorithmes de machine learning.

- **Term Frequency (TF)** : Il mesure combien de fois un terme apparaît dans un document par rapport à la longueur totale du document.
 - **Calcul** : $TF(t) = (\text{Nombre d'occurrences du terme } t \text{ dans un document}) / (\text{Nombre total de termes dans le document})$
- **Inverse Document Frequency (IDF)** : Il mesure l'importance d'un terme dans l'ensemble du corpus. Les termes courants dans de nombreux documents obtiennent un poids faible, tandis que les termes rares obtiennent un poids élevé.
 - **Calcul**: $IDF(t) = \log_e(\text{Total des documents} / \text{Nombre de documents contenant le terme } t)$
- **TF-IDF** : Le produit de TF et IDF pour chaque terme donne le poids du terme dans le document. Cela aide à équilibrer les termes fréquents et rares.
 - **Calcul**: $TF-IDF(t) = TF(t) * IDF(t)$

Et donc, TF-IDF nous aide à transformer le texte en un format numérique qui peut être utilisé par les modèles de machine learning.

2°) Entraînement des Modèles.

Nous avons entraîné chaque modèle sur les données d'entraînement et évalué leurs performances sur les données de test en utilisant plusieurs métriques : **Accuracy**, **Precision**, **Recall**, **F1 Score**, et **Matrice de Confusion**.

3°) Résultats des Modèles.

Explication des Métriques de Performance :

Accuracy : C'est le pourcentage de prédictions correctes sur l'ensemble des prédictions effectuées.

- **Calcul** : $(\text{Vrais Positifs} + \text{Vrais Négatifs}) / \text{Total des prédictions}$
- **Interprétation** : Plus la valeur est proche de 1, meilleur est le modèle.

Precision : C'est la proportion de vraies prédictions positives parmi toutes les prédictions positives effectuées.

- **Calcul** : $\text{Vrais Positifs} / (\text{Vrais Positifs} + \text{Faux Positifs})$
- **Interprétation** : Indique la précision des prédictions positives du modèle. Une précision élevée signifie moins de faux positifs.

Recall : C'est la proportion de vraies prédictions positives parmi toutes les instances positives réelles.

- **Calcul** : $\text{Vrais Positifs} / (\text{Vrais Positifs} + \text{Faux Négatifs})$
- **Interprétation** : Indique la capacité du modèle à détecter les instances positives. Un rappel élevé signifie moins de faux négatifs.

F1 Score : C'est la moyenne harmonique de la précision et du rappel. Il équilibre les deux métriques.

- **Calcul** : $2 * (\text{Précision} * \text{Rappel}) / (\text{Précision} + \text{Rappel})$
- **Interprétation** : Un F1 Score élevé indique que le modèle a un bon équilibre entre précision et rappel.

Confusion Matrix : C'est une matrice qui permet de visualiser les **Vrais Positifs** (TP), **Faux Positifs** (FP), **Vrais Négatifs** (TN), **Faux Négatifs** (FN). Elle aide à comprendre les types d'erreurs que le modèle fait.

```

Logistic Regression Performance
Accuracy: 0.987305122494432
Precision: 0.9904905986600389
Recall: 0.9849559424027509
F1 Score: 0.9877155172413792
Confusion Matrix:
[[4283 44]
 [ 70 4583]]

```

```

Random Forest Performance
Accuracy: 0.9972160356347439
Precision: 0.9980628497632372
Recall: 0.9965613582634859
F1 Score: 0.9973115388751479
Confusion Matrix:
[[4318 9]
 [ 16 4637]]

```

```

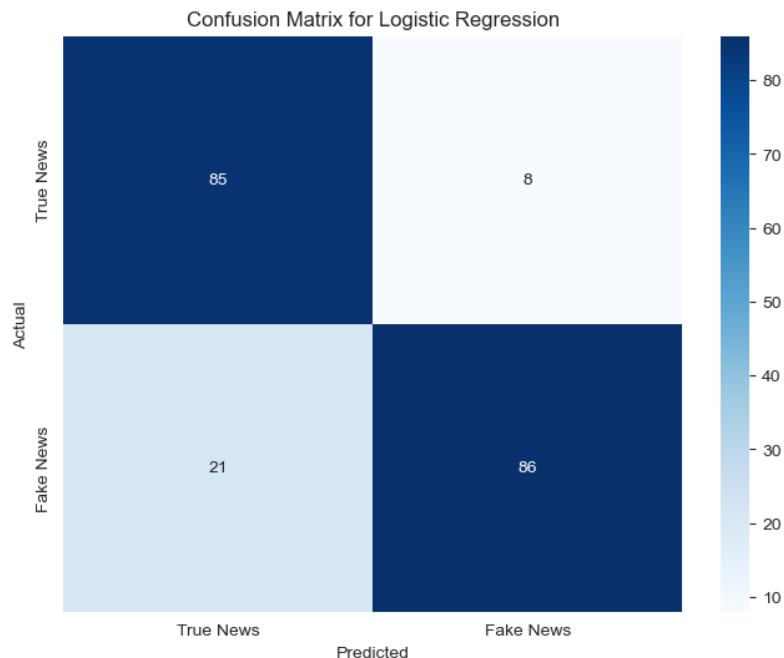
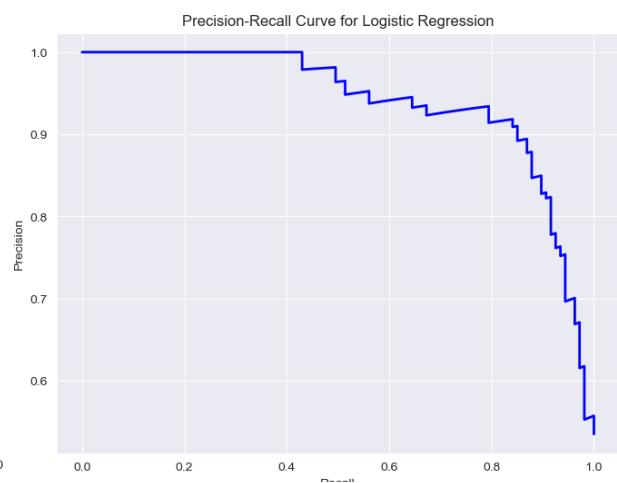
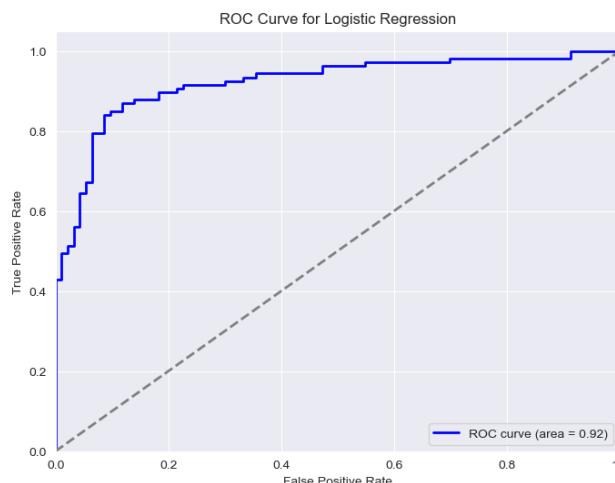
Decision Tree Performance
Accuracy: 0.9942093541202672
Precision: 0.9925069578248769
Recall: 0.9963464431549538
F1 Score: 0.9944229944229944
Confusion Matrix:
[[4292 35]
 [ 17 4636]]

```

```

SVM Performance
Accuracy: 0.9927616926503341
Precision: 0.9937580714593198
Recall: 0.9922630560928434
F1 Score: 0.9930100010753846
Confusion Matrix:
[[4298 29]
 [ 36 4617]]

```



Les résultats montrent que les modèles de machine learning traditionnels tels que la régression logistique, l'arbre de décision, la forêt aléatoire et le SVM sont efficaces pour la détection de fake news, avec des performances particulièrement élevées pour la forêt aléatoire et la régression logistique. Les visualisations aident à mieux comprendre les performances de chaque modèle et à identifier leurs forces et leurs faiblesses.

B/ Modèles de Deep Learning.

Le modèle **LSTM** (Long Short-Term Memory) est un type de réseau de neurones récurrents (**RNN**) qui est particulièrement bien adapté pour traiter et prédire des séries temporelles et des séquences de données. Les LSTM sont capables de capturer les dépendances à long terme dans les séquences de texte, ce qui les rend efficaces pour comprendre le contexte et les relations entre les mots sur de longues distances. Dans le cadre de la détection des fake news, les LSTM peuvent analyser le contenu textuel des articles pour identifier les caractéristiques spécifiques qui distinguent les vraies nouvelles des fausses.

1°) Préparation des Données.

Avant de pouvoir entraîner le modèle LSTM, les données textuelles doivent être prétraitées et converties en séquences numériques. Les étapes principales de cette préparation sont les suivantes :

- **Tokenization** : Division du texte en unités significatives (tokens).
- **Séquençage** : Conversion des tokens en séquences de nombres.
- **Padding** : Alignement des séquences à une longueur fixe pour le traitement en lots.

2°) Construction et Entraînement du Modèle LSTM.

La construction d'un modèle LSTM pour la détection des fake news implique plusieurs étapes clés :

- Architecture du Modèle :
 - **Embedding Layer** : Cette couche convertit les mots en vecteurs de dimension fixe, permettant au modèle de capturer les relations sémantiques entre les mots.
 - **Bidirectional LSTM Layers** : Utilisation de couches LSTM bidirectionnelles qui permettent au modèle de traiter les séquences de texte dans les deux directions (avant et arrière), améliorant ainsi la compréhension du contexte.
 - **Dropout Layers** : Ces couches sont ajoutées pour réduire le surapprentissage (overfitting) en désactivant aléatoirement des neurones pendant l'entraînement.
 - **Dense Layer** : Une couche dense finale avec une activation sigmoïde pour produire une probabilité que l'article soit une fake news ou non.
- Entraînement du Modèle :
 - **Compilation du Modèle** : Le modèle est compilé avec une fonction de perte appropriée (par exemple, `binary_crossentropy` pour la classification binaire) et un optimiseur (par exemple, Adam) qui ajuste les poids du modèle.

- **Phase d'Entraînement :** Le modèle est entraîné sur les données d'entraînement pendant plusieurs époques. Pendant l'entraînement, le modèle ajuste ses poids en fonction des erreurs commises sur les prédictions.
- **Validation :** Le modèle est également évalué sur un ensemble de validation après chaque époque pour surveiller ses performances et ajuster les hyperparamètres si nécessaire.

```
Epoch 1/5
562/562 [=====] - 1375s 2s/step - loss: 0.1192 - accuracy: 0.9562 - val_loss: 0.0687 - val_accuracy: 0.9778
Epoch 2/5
562/562 [=====] - 1365s 2s/step - loss: 0.0780 - accuracy: 0.9707 - val_loss: 0.0435 - val_accuracy: 0.9833
Epoch 3/5
562/562 [=====] - 1345s 2s/step - loss: 0.0397 - accuracy: 0.9857 - val_loss: 0.0393 - val_accuracy: 0.9876
Epoch 4/5
562/562 [=====] - 1397s 2s/step - loss: 0.0481 - accuracy: 0.9839 - val_loss: 0.0999 - val_accuracy: 0.9700
Epoch 5/5
562/562 [=====] - 1286s 2s/step - loss: 0.0380 - accuracy: 0.9874 - val_loss: 0.0451 - val_accuracy: 0.9851
<keras.src.callbacks.History at 0x7b4bb5557f40>
```

3°) Évaluation et Entraînement du Modèle LSTM.

Les résultats d'évaluation montrent que le modèle LSTM est très performant pour la classification des fake news, avec des métriques élevées de précision, de rappel et de score F1. La matrice de confusion montre également un faible nombre de faux positifs et de faux négatifs.

```
LSTM Performance
Accuracy: 0.97728285077951
Precision: 0.9878022217381834
Recall: 0.9683963271407218
F1 Score: 0.9780030191934441
Confusion Matrix:
[[4241 56]
 [ 148 4535]]
```

C/ Modèle Pré-entraîné de Langage BERT.

BERT (Bidirectional Encoder Representations from Transformers) est un modèle de langage pré-entraîné développé par Google, qui a révolutionné le traitement du langage naturel (NLP) en capturant le contexte bidirectionnel des mots dans une phrase. Contrairement aux modèles traditionnels qui lisent les textes de gauche à droite ou de droite à gauche, BERT considère le contexte complet des mots en lisant simultanément à gauche et à droite. Cela permet une compréhension plus profonde des relations contextuelles entre les mots, rendant BERT particulièrement efficace pour les tâches de classification de texte, comme la détection de fake news.

1°) Préparation des Données.

Avant de pouvoir utiliser BERT pour entraîner notre modèle, nous devons nettoyer et préparer les données textuelles :

- **Nettoyage du Texte :**
 - Convertir les textes en minuscules.
 - Supprimer les balises HTML, les URL, les mentions (@) et les hashtags (#).
 - Supprimer les mots vides (stopwords) en utilisant la bibliothèque NLTK (Natural Language Toolkit).
- **Tokenization avec BERT :**

- Utiliser le tokenizer pré-entraîné de BERT pour convertir les textes nettoyés en séquences de tokens.
- Appliquer le padding et la troncature pour assurer que toutes les séquences ont la même longueur, facilitant ainsi le traitement par lot.

2°) Construction et Entraînement du Modèle BERT.

- **Initialisation du Modèle BERT :** Charger le modèle pré-entraîné `bert-base-uncased` pour la classification de séquences.
- **Définition des Arguments d'Entraînement :** Spécifier les paramètres d'entraînement tels que le *nombre d'époques*, la *taille des lots*, le *taux de décroissance des poids*, et le répertoire de sortie pour enregistrer les résultats.
- **Initialisation du Trainer :**
 - Utiliser la classe `Trainer` de la bibliothèque `transformers` pour gérer l'entraînement du modèle.
 - Fournir le modèle, les arguments d'entraînement, ainsi que les jeux de données d'entraînement et de validation.
- **Entraînement du Modèle :** L'entraînement de BERT est computationnellement intensif, nécessitant des ressources matérielles puissantes (comme les GPU) pour gérer les grandes quantités de données et les calculs complexes. BERT apprend à ajuster ses paramètres en fonction des erreurs commises sur les prédictions pendant l'entraînement, en utilisant des techniques d'*optimisation* avancées comme **Adam**.

3°) Évaluation du Modèle BERT.

Les métriques telles que l'accuracy, la précision, le rappel et le F1 score fournissent une évaluation complète des performances du modèle. Une précision et un rappel élevés indiquent que le modèle est efficace pour détecter les fake news avec peu d'erreurs.

BERT nécessite des ressources matérielles importantes, ce qui peut poser des défis pour les environnements limités. Le temps d'entraînement peut être long en raison de la complexité du modèle et de la taille des données. Bien que performant, BERT peut être difficile à interpréter, car il s'agit d'un modèle de boîte noire avec de nombreux paramètres internes.

VI/ MISE EN PLACE DU PROJET

A/ Planification.

1°) Définition du Projet (1 semaine).

Tâches :

- Définir les objectifs du projet.
- Identifier les exigences et les contraintes.
- Rechercher la littérature existante sur la détection des fake news.
- Définir les métriques de performance.

Rôles :

- **Chef de Projet** : Coordination de l'équipe et définition des objectifs (Salaire : 60,000€/an (5,000€/mois, 1,154€/semaine)).
- **Data Scientist** : Recherche de la littérature et définition des métriques de performance (Salaire : 50,000€/an (4,167 €/mois, 962€/semaine)).
- **Analyste de Données** : Identification des exigences et des contraintes (Salaire : 40,000€/an(3,333€/mois, 769€/semaine)).

2°) Préparation des Données (2 semaines).

Tâches :

- Télécharger les datasets nécessaires depuis un site internet.
- Prétraiter les données (nettoyage, tokenization, etc.).
- Diviser les données en ensembles d'entraînement et de test.

Rôles :

- **Data Engineer** : Téléchargement et préparation des données (Salaire : 60,000€/an (5,000€/mois, 1,154€/semaine)).
- **Data Scientist** : Nettoyage et prétraitement des données (Salaire : 50,000€/an (4,167 €/mois, 962€/semaine)).

Outils :

- Python pour le nettoyage et le prétraitement des données.
- Pandas et Numpy pour la manipulation des données.
- NLTK et Gensim pour la tokenization.

- Google Colab pour l'exécution des notebooks.

3°) Modélisation (3 semaines).

Tâches :

- Entrainer des modèles de classification traditionnels (régression logistique, arbre de décision, forêt aléatoire, SVM).
- Entrainer un modèle LSTM.
- Entrainer un modèle BERT.

Rôles :

- **Machine Learning Engineer** : Implémentation et optimisation des modèles complexes (Salaire : 55,000€/an (4,583€/mois, 1058€/semaine)).
- **Data Scientist** : Nettoyage et prétraitement des données (Salaire : 50,000€/an (4,167 €/mois, 962€/semaine)).

Outils :

- Scikit-Learn pour les modèles de classification traditionnels.
- TensorFlow/Keras pour les modèles LSTM.
- Transformers (Hugging Face) pour le modèle BERT.
- Google Colab pour l'entraînement des modèles avec GPU.

4°) Évaluation et Optimisation des Modèles (2 semaines).

Tâches :

- Évaluer les modèles en utilisant les métriques de performance définies.
- Optimiser les hyperparamètres des modèles.
- Comparer les performances des modèles.

Rôles :

- **Machine Learning Engineer** : Implémentation et optimisation des modèles complexes (Salaire : 55,000€/an (4,583€/mois, 1058€/semaine)).
- **Data Scientist** : Nettoyage et prétraitement des données (Salaire : 50,000€/an (4,167 €/mois, 962€/semaine)).

Outils :

- Scikit-Learn pour les métriques de performance.
- Hyperopt pour l'optimisation des hyperparamètres.
- Matplotlib et Seaborn pour la visualisation des résultats.

5°) Documentation et Présentation (1 semaine).

Tâches :

- Documenter le projet (méthodologie, résultats, code).
- Préparer une présentation finale.
- Présenter les résultats aux parties prenantes.

Rôles :

- **Chef de Projet** : Coordination de l'équipe et définition des objectifs (Salaire : 60,000€/an (5,000€/mois, 1,154€/semaine)).
- **Data Scientist** : Recherche de la littérature et définition des métriques de performance (Salaire : 50,000€/an (4,167 €/mois, 962€/semaine)).
- **Technical Writer** : Documentation du projet (Salaire : 40,000€/an(3,333€/mois, 769€/semaine)).

B/ Suivi du projet.

1°) Calculs des coûts.

Charge	Coût Hebdomadaire (€)
Chef de Projet	1,154
Data Scientist	962
Data Engineer	1,154
Machine Learning Engineer	1058
Analyste de Données	769
Technical Writer	769

Équipements et Matériel	50
Coûts Administratifs	75

Coût Total pour chaque tâche :

1. **Définition du Projet** (1 semaine) :

- **Chef de Projet** : 1,154 €
- **Data Scientist** : 962 €
- **Analyste de Données** : 769 €
- **Équipements et Matériel** : 50 €
- **Coûts Administratifs** : 75 €
- **Total** : 3,010 €

2. **Préparation des Données** (2 semaines) :

- **Data Engineer** : $1,154 \text{ €} \times 2 = 2,308 \text{ €}$
- **Data Scientist** : $962 \text{ €} \times 2 = 1,924 \text{ €}$
- **Équipements et Matériel** : $50 \text{ €} \times 2 = 100 \text{ €}$
- **Coûts Administratifs** : $75 \text{ €} \times 2 = 150 \text{ €}$
- **Total** : 4,482 €

3. **Modélisation** (3 semaines) :

- **Data Scientist** : $962 \text{ €} \times 3 = 2,886 \text{ €}$
- **Machine Learning Engineer** : $1,058 \text{ €} \times 3 = 3,174 \text{ €}$
- **Équipements et Matériel** : $50 \text{ €} \times 3 = 150 \text{ €}$
- **Coûts Administratifs** : $75 \text{ €} \times 3 = 225 \text{ €}$
- **Total** : 6,435 €

4. **Évaluation et Optimisation des Modèles** (2 semaines) :

- **Data Scientist** : $962 \text{ €} \times 2 = 1,924 \text{ €}$
- **Machine Learning Engineer** : $1,058 \text{ €} \times 2 = 2,116 \text{ €}$

- Équipements et Matériel : $50 \text{ €} \times 2 = 100 \text{ €}$
- Coûts Administratifs : $75 \text{ €} \times 2 = 150 \text{ €}$
- **Total** : 4,290 €

5. Documentation et Présentation (1 semaine) :

- Technical Writer : 769 €
- Chef de Projet : 1,154 €
- Data Scientist : 962 €
- Équipements et Matériel : 50 €
- Coûts Administratifs : 75 €
- **Total** : 3,010 €

Total des Coûts = $3,010 + 4,482 + 6,435 + 4,290 + 3,010 = \mathbf{21,227 \text{ €}}$

2°) Calcul de la VAN et du TRI.

Nous allons utiliser une formule simplifiée pour la **VAN** et le **TRI** en prenant en compte les *revenus hebdomadaires de 1,500 €*.

Données du Projet :

Investissement Initial (Coût Total)	21,227 €
Revenus Hebdomadaires	1,500 €
Durée du Projet	9 semaines
Taux d'Actualisation Hebdomadaire	$0.05 / 52 = \mathbf{0,000962}$ (supposons un taux annuel de 5%)

Calcul de la VAN :

$$VAN = \sum_{t=1}^N \frac{CF_t}{(1+r)^t} - I_0$$

Où :

- CF_t : Flux de trésorerie à la période t (1,500 €)
- r : Taux d'actualisation hebdomadaire (0.000962)

- I_0 : Investissement initial (21,227 €)
- N : Nombre de semaines (9)

Donc : $VAN = \sum_{t=1}^9 \frac{1500}{(1+0.000962)^t} - 21227$

$$\frac{1500}{(1+0.000962)^1} = 1498.56$$

$$\frac{1500}{(1+0.000962)^2} = 1497.12$$

$$\frac{1500}{(1+0.000962)^3} = 1495.68$$

$$\frac{1500}{(1+0.000962)^4} = 1494.24$$

$$\frac{1500}{(1+0.000962)^5} = 1492.81$$

$$\frac{1500}{(1+0.000962)^6} = 1491.37$$

$$\frac{1500}{(1+0.000962)^7} = 1489.51$$

$$\frac{1500}{(1+0.000962)^8} = 1488.51$$

$$\frac{1500}{(1+0.000962)^9} = 1487.08$$

$$VAN = 1498.56 + 1497.12 + 1495.68 + 1494.24 + 1492.81 + 1491.37 + 1489.51 + 1488.51 + 1487.08 - 21227 = -7791.69\text{€}$$

Calcul du TRI :

Le TRI est le taux d'actualisation pour lequel la VAN est égale à zéro.

$$0 = \sum_{t=1}^N \frac{CF_t}{(1+)^t} - I_0$$

Notons $VAN_- = -7791.69\text{€}$, calculons VAN avec des revenus hebdomadaires de 2375€, on trouve $VAN = 45.55\text{€}$, avec $TRI = 0\%$.

	-21227	-21227	-21227
	1500	1450	2375
	1500	1450	2375
	1500	1450	2375
	1500	1450	2375
	1500	1450	2375
	1500	1450	2375
	1500	1450	2375
	1500	1450	2375
VAN	-€7 791,71	-€8 239,55	€45,55
TRI	-8%	-9%	0%

3*) Suivi du projet.

Explication des termes :

1. **CR (Coût Réel)** : Coût réel engagé pour effectuer les travaux réalisés jusqu'à présent.
2. **VP (Valeur Planifiée)** : Valeur budgétée pour les travaux qui étaient prévus à ce jour.
3. **VA (Valeur Acquise)** : Valeur budgétée pour les travaux réellement réalisés.
4. **EA (Écart de Coût)** : Différence entre la Valeur Acquise et le Coût Réel.
 $EA = VA - CR$.
5. **VC (Variation des Coûts)** : Mesure de l'écart entre le coût budgétisé et le coût réel.
6. **IPC (Indice de Performance des Coûts)** : Ratio de la valeur acquise sur le coût réel.
 $IPC = VA / CR$.

7. **VD (Variation des Délais)** : Mesure de l'écart entre le travail planifié et le travail réalisé.
8. **IPP (Indice de Performance des Délais)** : Ratio de la valeur acquise sur la valeur planifiée. $IPP = VA / VP$.
9. **EAA (Estimation à l'Achèvement)** : Prévision des coûts totaux du projet à l'achèvement. Peut être calculée de plusieurs façons :
 - o $EAA1 : EAA = CR + (\text{Budget total} - VA)$
 - o $EAA2 : EAA = \text{Budget total} / IPC$
 - o $EAA3 : EAA = CR + ((\text{Budget total} - VA) / (IPC * IPP))$

10. **DAA (Délai à l'Achèvement)** : Prévision des délais totaux du projet à l'achèvement.

Supposons que nous avons les données suivantes pour notre projet jusqu'à présent :

- **Budget Total (BT)** : **21,227 €**
- **Coût Réel (CR)** : **10,000 €**
- **Valeur Planifiée (VP)** : **12,000 €**
- **Valeur Acquise (VA)** : **11,000 €**
- **Nombre de semaines écoulées** : **5**
- **Nombre total de semaines** : **9**

Calculs :

1. **EA (Écart de Coût)** :

$$EA = VA - CR = 11,000 - 10,000 = 1,000 \text{ €}$$

2. **VC (Variation des Coûts)** :

$$VC = \text{Budget Total} - \text{Coût Réel} = 21,227 - 10,000 = 11,227 \text{ €}$$

3. **IPC (Indice de Performance des Coûts)** :

$$IPC = \frac{VA}{CR} = \frac{11,000}{10,000} = 1.1$$

4. **VD (Variation des Délais)** :

$$VD = VA - VP = 11,000 - 12,000 = -1,000 \text{ €}$$

5. **IPP (Indice de Performance des Délais)** :

$$\text{IPP} = \frac{\text{VA}}{\text{VP}} = \frac{11,000}{12,000} = 0.9167$$

6. EAA1 (Estimation à l'Achèvement) :

$$\text{EAA1} = \text{CR} + (\text{Budget Total} - \text{VA}) = 10,000 + (21,227 - 11,000) = 20,227 \text{ €}$$

7. EAA2 (Estimation à l'Achèvement) :

$$\text{EAA2} = \frac{\text{Budget Total}}{\text{IPC}} = \frac{21,227}{1,1} = 19,297.27 \text{ €}$$

8. EAA3 (Estimation à l'Achèvement) :

$$\text{EAA3} = \text{CR} + \frac{\text{Budget Total} - \text{VA}}{\text{IPC} * \text{IPP}} = 10,000 + \frac{21,227 - 11,000}{1,1 * 0,9167} = 20,142.97 \text{ €}$$

9. DAA1 (Délai à l'Achèvement) :

Supposons que le projet était initialement prévu pour durer 9 semaines, et que nous avons déjà réalisé 5 semaines de travail :

$$\text{DAA1} = \frac{\text{Nombre total de semaines}}{\text{IPP}} = \frac{9}{0,9167} = 9.81 \text{ semaines}$$

10. DAA2 (Délai à l'Achèvement) :

Une autre méthode consiste à ajouter les semaines restantes en fonction de l'IPP.

$$\text{DAA2} = \text{Semaines réalisées} + \frac{\text{Semaines restantes}}{\text{IPP}} = 5 + \frac{4}{0,9167} = 9.36 \text{ semaines}$$

VII/ CONCLUSION

À la lumière de nos résultats, il est clair que le NLP peut être utilisé efficacement pour identifier et classifier les fake news dans le contexte politique aux États-Unis. Les modèles de machine learning et de deep learning, lorsqu'ils sont correctement entraînés et évalués, peuvent analyser les textes des articles de presse et distinguer avec précision les vraies nouvelles des fausses. Les approches basées sur BERT et LSTM, en particulier, montrent une grande promesse pour capturer les nuances contextuelles et fournir des résultats fiables.

Cependant, il existe plusieurs perspectives d'amélioration pour ce projet. Premièrement, l'intégration de techniques de traitement du langage naturel plus avancées, telles que les modèles de langage de nouvelle génération comme GPT-3 ou T5, pourrait améliorer encore davantage les performances de détection. De plus, l'optimisation des hyperparamètres et l'utilisation de techniques d'ensemble (ensemble learning) pourraient renforcer la robustesse et la précision des modèles.

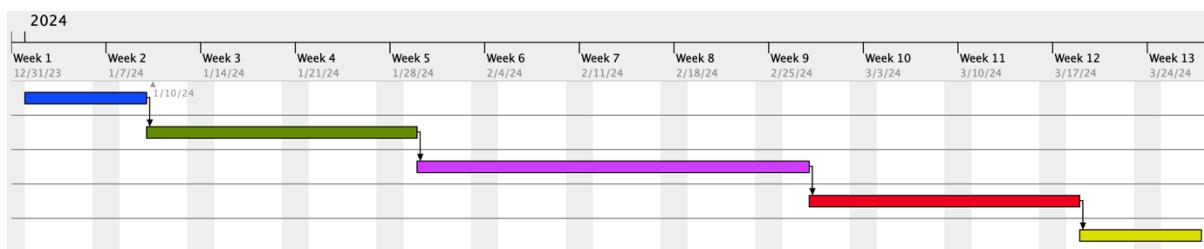
ANNEXES

Données : Kaggle : <https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset/data>

Planification (Tâches) :

Name	Begin date	End date
Définition du Projet	1/1/24	1/9/24
Préparation des Données	1/10/24	1/29/24
Modélisation	1/30/24	2/27/24
Évaluation et Optimisation des Modèles	2/28/24	3/18/24
Documentation et Présentation	3/19/24	3/27/24

Diagramme de Gantt :



Ressources (Rôles/Tâches) :

Name
▼ Data Scientist
● Définition du Projet
● Préparation des Données
● Modélisation
● Évaluation et Optimisation des Modèles
● Documentation et Présentation
▼ Chef de Projet
● Définition du Projet
● Documentation et Présentation
▼ Analyste de Données
● Définition du Projet
▼ Data Engineer
● Préparation des Données
▼ Machine Learning Engineer
● Modélisation
● Évaluation et Optimisation des Modèles
▼ Technical Writer
● Documentation et Présentation

