**Conception d'un détecteur de fake news avec création d'un ensemble de données associé**

Date://///

Salimata FAYE

2024

Table de matières

# **Partie1 Introduction**

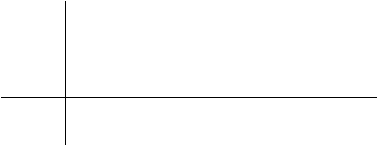
1. Introduction au système
2. Définition du problème
3. Objectif
4. But
5. Besoin du système

# **Partie2 Les étapes du projet**

1. Étape 1 : Collecte et Annotation des données.
2. Étape 2 : Analyse exploratoire des données
3. Étape 3 : Prétraitement des données
4. Étape 4 : Déploiement et évaluation du modèle

# **Partie3 Conception de l’application**

1. Étape 1 : Création d’une interface utilisateur avec Flask
2. Étape 2 : Ajoutez des fonctionnalités.



**Partie# 1**

**Introduction**

*Contenus:*

* Introduction au système
* Définition du problème
* Objectif
* But
* Besoin du système

**Introduction au projet**

Le système de détection de fausses informations est devenu indispensable dans le paysage actuel de l'information numérique, où la propagation de fausses nouvelles peut avoir des conséquences néfastes sur la société. Notre projet vise à concevoir un système robuste et efficace pour détecter les fausses informations, en utilisant des techniques avancées de traitement du langage naturel et d'apprentissage automatique.

Ce système repose sur la création d'un ensemble de données exhaustif et diversifié, qui servira de base à notre modèle de détection de fausses informations. L'ensemble de données comprendra un large éventail d'articles de presse provenant de sources variées, tant réputées que moins fiables, afin de garantir une représentation réaliste des différents types de contenus médiatiques disponibles en ligne.

En utilisant cet ensemble de données, nous mettrons en œuvre une variété de techniques d'apprentissage automatique, notamment des algorithmes de classification tels que la Régression Logistique, pour entraîner notre modèle de détection de fausses informations.

**Définition du problème**

Le problème abordé par le projet est la prolifération de fausses informations sur Internet, qui peut entraîner des conséquences négatives telles que la panique publique, la méfiance envers les sources crédibles et l'instabilité sociale. Le projet vise à développer un système automatisé capable de distinguer les vraies informations des fausses informations en utilisant des techniques de traitement du langage naturel et d'apprentissage automatique. Cela aidera à prévenir la propagation de la désinformation et à protéger le public contre la manipulation.

**Objectif**

L'objectif principal de ce projet de conception d'un détecteur de fausses informations avec création d'un ensemble de données associé est de développer un système automatisé capable de détecter et de classifier de manière précise les fausses informations présentes dans les articles de presse en ligne.

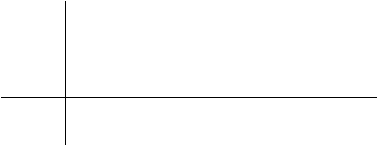
**But**

Plus spécifiquement, les buts du projet sont les suivants :

* Construire un ensemble de données complet et diversifié comprenant une variété d'articles de presse provenant de différentes sources et couvrant différents sujets.
* Annoter l'ensemble de données pour indiquer la véracité de chaque article, fournissant ainsi une référence pour l'entraînement du modèle de détection de fausses informations.
* Développer et mettre en œuvre un algorithme de détection de fausses informations basé sur des techniques avancées de traitement du langage naturel et d'apprentissage automatique.
* Optimiser le modèle de détection de fausses informations pour garantir des performances élevées en termes de précision et de rappel.
* Créer une interface utilisateur conviviale permettant aux utilisateurs de soumettre des articles pour analyse et de visualiser les résultats de la détection de fausses informations.
* Sensibiliser les utilisateurs à la propagation de la désinformation en ligne et les aider à développer des compétences critiques pour évaluer la fiabilité des sources d'information.

**Besoin du projet**

Le besoin du projet de détection de fausses informations découle de la propagation croissante de fausses nouvelles et de désinformation à travers les médias sociaux et d'autres plateformes en ligne. Le système vise à résoudre le problème des fausses informations en détectant automatiquement et en filtrant les informations fausses des vraies. Avec l'aide d'algorithmes d'apprentissage automatique le système peut identifier efficacement les fausses nouvelles, ce qui peut empêcher les gens de prendre des décisions basées sur des informations trompeuses. Ce système est crucial pour maintenir l'authenticité des informations et peut être utilisé par diverses industries telles que les agences de presse, les plateformes de médias sociaux et les institutions éducatives.



**Les étapes du projet**

**Partie# 2**

*Contenus:*

* Étape 1 : Collecte et annotation des données.
* Étape 2 : Prétraitement des données
* Étape 3 : Analyse exploratoire des données
* Étape 4 : Formation et déploiement du modèle

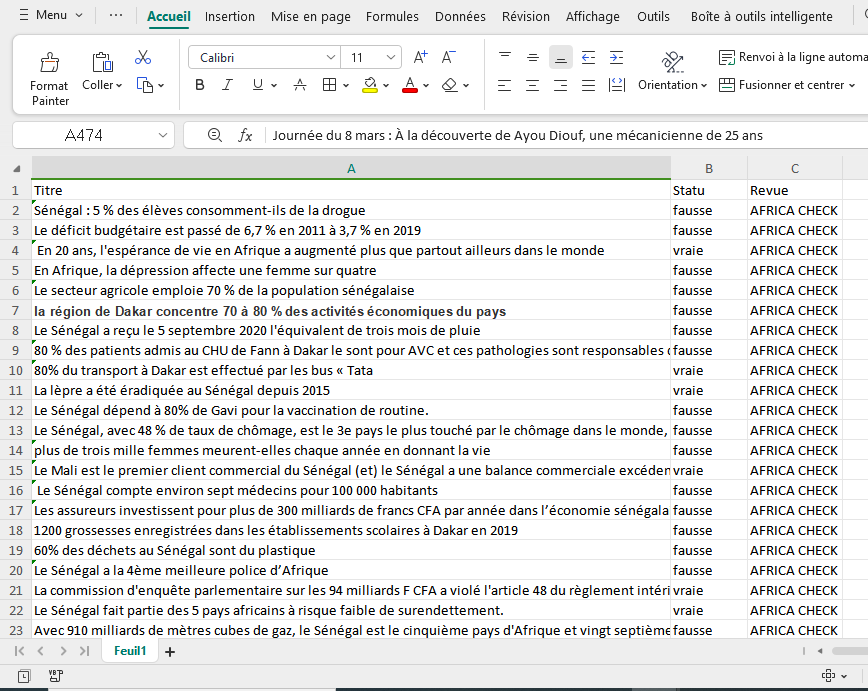
***Étape 1 : Collecte et étiquetages des données***

Les données utilisées pour la réalisation du projet sont collectées à partir de sites d'information bien connus tels que Senego, Seneweb, Chech Afrika, et PressAfrik. Le processus de collecte et de préparation des données a été effectuer comme suite :

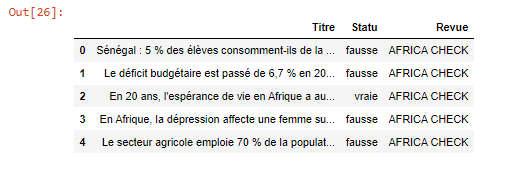
* Récupération des données à partir de sites d'information en ligne, notamment Senego, Seneweb, Chech Afrika, et PressAfrik. Cela a été faite manuellement

* Stockage des données dans un fichier Excel. Chaque article est enregistré avec des informations telles que le titre, la revue, id, etc.
* Étiquetage des données : Les articles stockés sont ensuite examinés et étiquetés en fonction de leur véracité ou de leur statut de vérité. Cela a été faite manuellement par nous.



Après avoir collecté et stocké les données à partir des différents sites d'information,nous avons procédé à la lecture de la base de données dans Jupyter Notebook pour commencer les analyses.



**Etape 2 : Prétraitement des données**

* Prétraitement et Normalisation du texte

C’est un processus important qui comprend la suppression des ponctuations, la réduction de tous les caractères en majuscules, la suppression de tous les mots vides et la radicalisation. La plupart du temps, nous appelons ce processus la tokenisation.

Exemple:

Si on a un texte il nous faut d’abord effectuer un prétraitement. Les étapes de la préparation du texte sont :

Texte exemple: "Les bienfaits de l'exercice physique sur la santé» «L'exercice physique régulier est essentiel pour maintenir une bonne santé physique et mentale»..."

*Conversion en minuscules*: "les bienfaits de l'exercice physique sur la santé» «l'exercice physique régulier est essentiel pour maintenir une bonne santé physique et mentale»..."

*Suppression des balise html et bracket*: "les bienfaits de l'exercice physique sur la santé l'exercice physique régulier est essentiel pour maintenir une bonne santé physique et mentale..."

*Suppression de la ponctuation*: "les bienfaits de lexercice physique sur la santé lexercice physique régulier est essentiel pour maintenir une bonne santé physique et mentale"

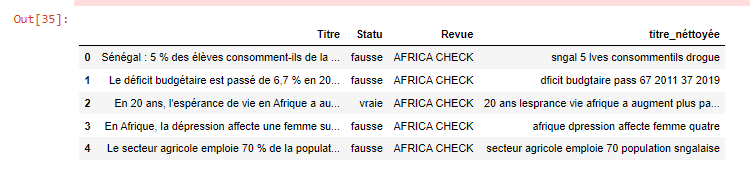
*Suppression des caractères spéciaux*: les bienfaits de lexercice physique sur la santé lexercice physique régulier est essentiel pour maintenir une bonne santé physique et mentale

*Suppression des mots vides*: bienfaits lexercice physique santé lexercice physique régulier essentiel maintenir bonne santé physique mentale"

*Tokenisation du texte:* "bienfaits", "lexercice", "physique", "santé", "lexercice", "physique", "régulier", "essentiel", "maintenir", "bonne", "santé", "physique", "mentale"

*Extraction de la racine* : "bienfait", "lexercic", "physiqu", "sant", "lexercic", "physiqu", "réguli", "essentiel", "maintien", "bon", "sant", "physiqu", "mental"

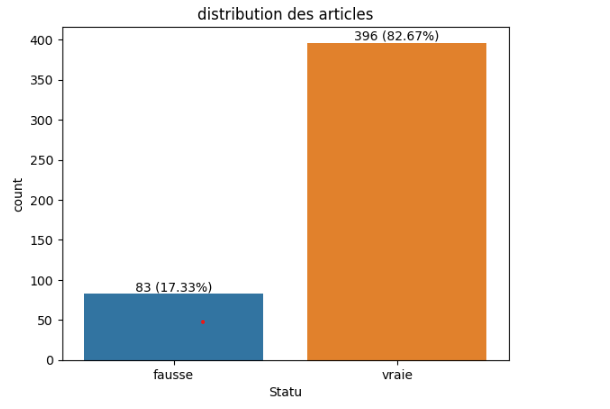
Donc au sortie du résultat on aurons ceci:



***Etape3 : Analyse exploratoire des données***

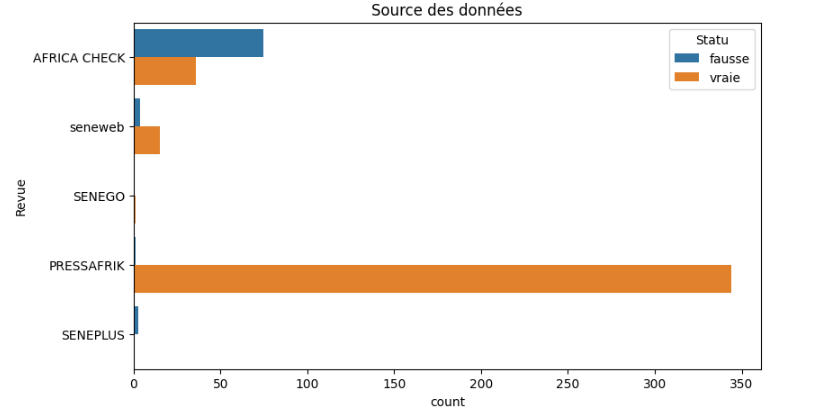
* Visualisation des données

Maintenant pour visualiser la distribution des données fausses aux données vraies on va crée un graphique à barres . Cette analyse descriptive permet de mieux comprendre la répartition des différentes valeurs de la colonne "Status" dans les données.



Après visualisation on peut constater que les données avec le statut "fausse" représentent 17,33% (soit 83) tandis que les données avec le statut "vraie" représentent 82.67%(soit 396).

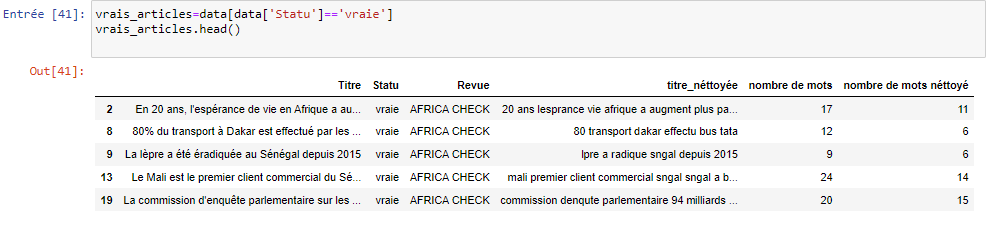
La répartition des revues montre que PRESSAFRIK et AFRIKA CHECK sont le plus représentés.

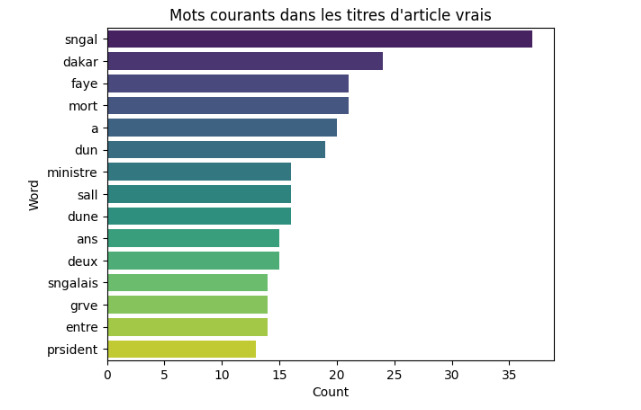


Cela est dû au fait que la plupart des articles sont issus de ces revues.

* Visualisation des nuages de mots des données

Séparation des données (vraies)

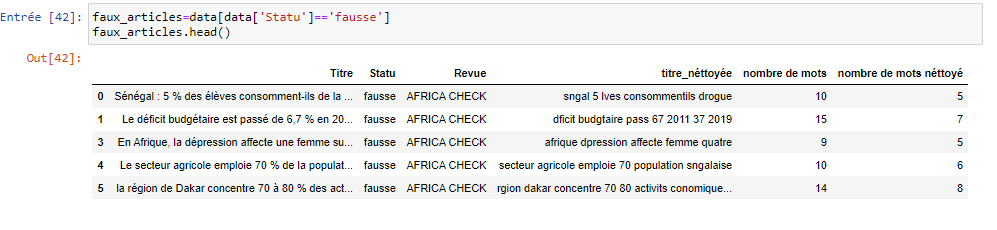


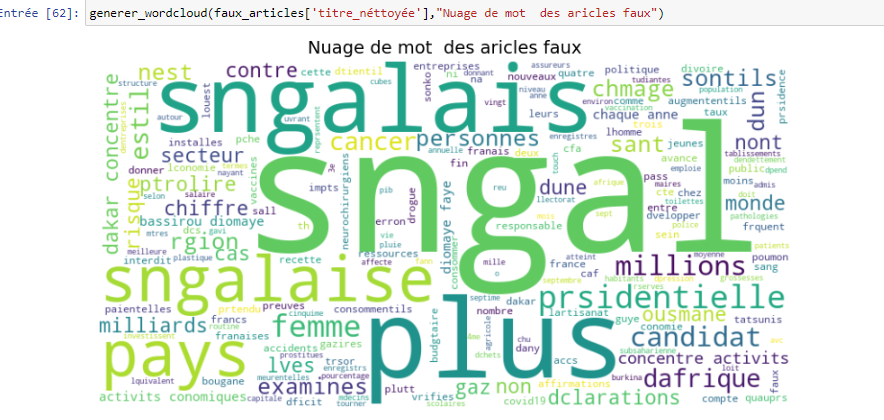
Nuage de mots WordCloud

Les mots les plus fréquents

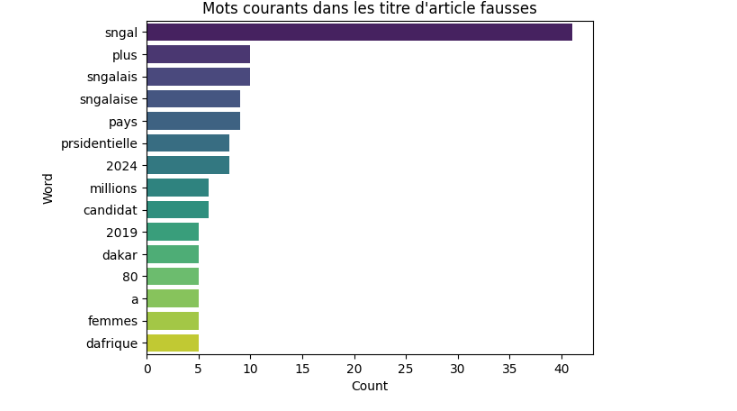
Séparation des données (fausses)



Nuages de mots (world cloud)



Nuage de mots WordCloud



Les mots les plus fréquents

**Etape4 : Conception, entrainement et évaluation d'un model de detection de fake news**

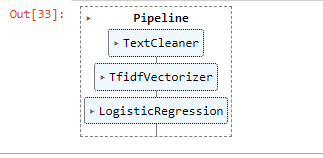
Tout d’abord avant d’entamer cette étape on va d’abord procéder à un pipeline.

**C’est quoi le pipeline ?**

Un pipeline en machine learning est une séquence d'étapes qui permet de transformer et de modéliser les données de manière automatique et cohérente. Voici les composants principaux d'un pipeline typique :

* Prétraitement des données : Cette étape comprend le nettoyage des données (par exemple, la suppression des valeurs manquantes, le traitement des caractère spéciaux etc...), la transformation des variables (par exemple, la normalisation, la discrétisation, l'encodage), et la sélection des caractéristiques pertinentes.
* Modélisation : Une fois que les données sont préparées, le pipeline inclut l'étape où le modèle de machine learning est sélectionné et entraîné sur les données traitées. Cela peut inclure l'ajustement des hyperparamétres du modèle pour obtenir de meilleures performances.
* Évaluation : Après avoir entraîné le modèle, le pipeline évalue sa performance en utilisant des métriques appropriées pour la tâche (par exemple, précision, rappel, F1-score pour la classification).
* Déploiement : Dans certains cas, le pipeline peut également inclure des étapes pour déployer le modèle entraîné dans un environnement de production, prêt à être utilisé pour des prédictions en temps réel.

Dans cadre de notre pipeline on procédera ainsi:



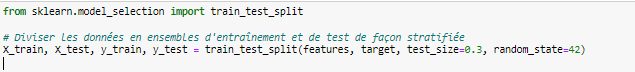
Modèle de prediction choisi

Vectorisation des mots avec TF-IDF

Étape de nettoyage de texte

Type de Modèle

Une fois que nous avons effectué le nettoyage des données, il est essentiel de diviser l'ensemble de données en ensembles d'entraînement et de test. Cette étape est cruciale pour évaluer la performance du modèle de manière impartiale.



Cela nous permettra de faire ensuite une conversion du texte en données numériques exploitables par les algorithmes de machine learning. Cette conversion est nécessaire car la plupart des modèles de machine learning ne peuvent pas traiter directement le texte brut.du texte en numérique.

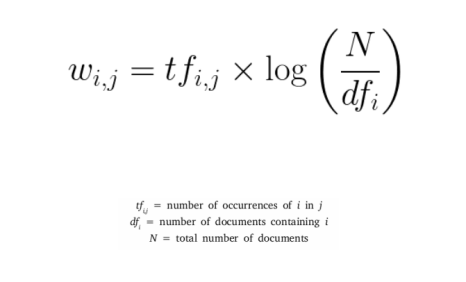
Cette vectorisation du texte consiste à représenter chaque document textuel par un vecteur numérique. Il existe plusieurs techniques de vectorisation, mais pour nous on utilisera le **TF-IDF (Term Frequency - Inverse Document Frequency).**

**Qu’est ce que sait??**

C’est une méthode qui permet de convertir chaque jeton en vecteur calculant l'importance d'un mot dans un document par rapport à une collection de documents. Elle combine la fréquence du terme (TF) et l'inverse de la fréquence dans les documents (IDF).

TF mesure le nombre de fois qu'un terme apparaît dans un document, tandis que IDF évalue l'importance d'un terme dans la collection globale de documents en réduisant le poids des termes fréquents.

En utilisant TF-IDF, on peut attribuer une valeur numérique à chaque terme, reflétant à la fois sa fréquence dans le document et son importance relative dans la collection de documents.



Supposons que nous ayons trois documents :

|  |  |  |  |
| --- | --- | --- | --- |
|  | Document 1 : | Document 2 | Document 3 |
| Phrases | "Le chat est mignon." | "Le chien est fidèle." | "Le chien et le chat jouent ensemble." |

Calculons maintenant le score TF-IDF pour les termes **"chat"** et **"chien"** dans chaque document :

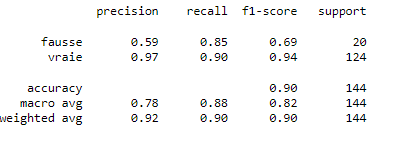
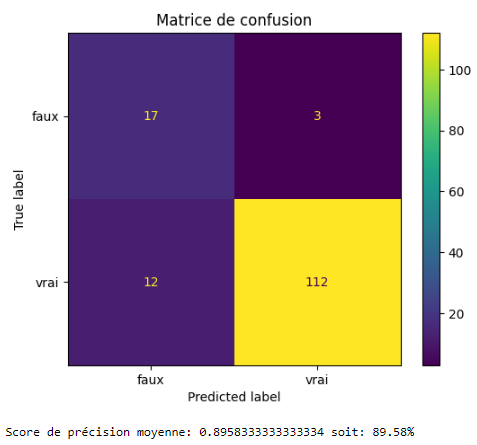
|  |  |  |  |
| --- | --- | --- | --- |
|  | Document 1 : | Document 2 | Document 3 |
| Phrases | "Le chat est mignon." | "Le chien est fidèle." | "Le chien et le chat jouent ensemble." |
| Fréquence des termes (TF) | TF("chat")= 1 | TF("chat") = 0 | TF("chat") = 1 |
| TF("chien")= 0 | TF("chien") = 1 | TF("chien") = 1 |
| Fréquence inverse dans les documents (IDF) | IDF("chat") = log(3/2) = log(1.5) ≈ 0.18 | | |
| IDF("chien") = log(3/2) = log(1.5) ≈ 0.18 | | |
| TF-IDF | TF-IDF("chat") =  1 \* 0.18 ≈ 0.18 | TF-IDF("chat")  = 0 \* 0.18 = 0 | TF-IDF("chat")  = 1 \* 0.18 ≈ 0.18 |
| TF-IDF("chien") =  0 \* 0.18 = 0 | TF-IDF("chien")  = 1 \* 0.18 ≈ 0.18 | TF-IDF("chien")  = 1 \* 0.18 ≈ 0.18 |

**Choix du modèle**

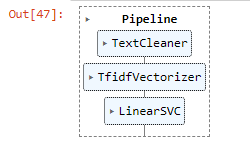
Le choix du modèle ici repose sur l’évaluation des performances d’algorithmes d’apprentissages.Donc pour se faire nous allons effectué une comparaisons de ces différentes algorithmes d’apprentissages considéré et en prendre le plus performant.

**Résultats: accuracy et matrix de confusion**

* SVM

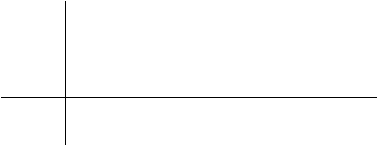
 

Selon l’évaluation établie sur différentes models , le SVM semble le plus performant.Donc notre choix se portera sur ce model.



Maintenant on peut faire des prédictions pour une entrée donnée en vue de voir si cette information est fiable ou non.

Ces prédictions seront faite sur une application que l’on réalisera.



**Mise en Application**

**Partie# 3**

*Contenus:*

* Étape 1 : Création d’une interface utilisateur avec Flask.
* Étape 2 : Ajoutez des fonctionnalités.