



# À la quête de vr(AI)

04.03.2024

Projet IA

---

Khedoudja Rym MERAD et Mike DURAN

Groupe GEMA IA school

M2 Data Science

Campus E-Learning

## Table des matières

Résumé.....	2
Introduction.....	3
Données.....	4
Problématique.....	13
Résultats.....	13
Challenges futures.....	24
Mise en place du projet.....	25
Conclusion.....	26
Glossaire.....	27
Annexe.....	28
Remerciements.....	33

## Résumé

Dans le cadre du projet IA de l'année scolaire 2023-2024 et qui porte sur le thème des fake news et des deep fake, nous avons décidé de porter notre sujet sur l'amélioration des détecteurs de fake news déjà existants: ajouter une subtilité capable de détecter le niveau d'erreur dans un article donné sans

Pour cela nous avons téléchargé des données textuelles labellisées que nous avons nettoyées, et traitées afin d'être comprises par la machine et sans biaiser les résultats de l'IA, nous les avons divisées en données d'entraînement et en données de validation.

Puis, nous avons analysé ces données de manière statistique et au sens du NLP afin d'assurer que nos traitements n'ont pas "trop" changé les données et que le résultat du traitement se rapporte et reste fidèle aux données d'origine.

Par la suite nous avons appliqué plusieurs modèles de détections de fake news:

- Arbre de décision
- Régression logistique
- Naive Bayes
- Passif Agressif
- Open AI API

Nous avons appliqué ces modèles avec et sans le BERT Tokenizer.

Chacun a ses points positifs et négatifs, ces domaines d'application, son temps d'apprentissage, ses paramètres et ses **indicateurs de performances**.

Pour finir le vrAI se base sur ces derniers pour donner une sorte de "nutri-score" d'une donnée donnée en entrée afin d'évaluer sa justesse.

## Introduction

Les homosapiens se sont toujours différenciés du reste des vivants par leur capacité de discernement, être capable de différencier le mal du bien, le vrai du faux ... mais plus le temps passe, plus les mensonges et les subterfuges de la misinformation sont de plus complexes et difficiles à détecter cela ne s'arrange pas avec le monde contemporain.

Dans un monde criblé de fake news sous forme de posts sur les réseaux sociaux visant à manipuler et à faire réagir les utilisateurs afin de collecter les avis de l'opinion publique ou tout simplement des Trolls, il est important de garder son sens critique. Cette tâche devient de plus en plus compliquée avec l'émergence des IA génératives de plus en plus réalistes et de la quantité d'informations que nous consommons.

C'est pour cela que dans le cadre du projet IA portant sur le thème des fake news et du deep fake, nous, Khedoudja Rym MERAD et Mike DURAN, avons décidé de donner un coup de pouce à l'humanité en lui donnant un outil capable de trancher si une donnée textuelle est vraie ou fausse.

Bien sûr, nous sommes au courant que nous ne sommes pas les seuls à avoir pensé à créer un détecteur de fake news. C'est pour cela que nous proposons une solution qui permet non seulement de dire qu'une information donnée est vraie ou fausse mais nous avons décidé d'intégrer une subtilité à notre outil **Le True Score: (vrAI)**.

## Données

Dans le domaine de l'IA il est essentiel d'exploiter des données, nous avons donc opté pour le téléchargement de bases de données csv contenant du texte labellisé en anglais de kaggle disponibles [ici](#).

Le fichier csv téléchargé est disponible sous le nom de title\_text.csv.

Nous avons regroupé les données réelles seules en un csv et les données fake en un autre csv que nous avons importé sur notre notebook.

### I. Données Fake

Une base de données contenant des données Fausses ou Fake news qui a:

- un titre
- un contenu
- Sujet
- Date
- un label isFake (True)

## II. Données True


Une base de données contenant des articles vrais ou True news qui a:

- un titre
- un contenu
- Sujet
- Date
- un label isFake (False)

## Analyse Statistique et LDA

Nous commençons par aller sur Data Analysis.ipynb, Nous avons:

- Chargé et classé les bibliothèques par thème et ordre d'utilisation
- Charger les données Fake.csv True.csv à partir du répertoire data
- Etude de la taille et des données vides à l'aide de info()
- Retirer les données vides: Mais il n'y en avait pas
- Reprendre la même taille d'échantillons true et fake 20000 données chacun.
- L'analyse des tailles des titres à montré que les titres les plus longs sont des titres de données fakes
- Dans ce cas pour ne pas biaiser les modèles, nous nous restreindrons aux articles qui ont une taille de maximum 15 772 mots (la taille maximale des true), sinon les modèles se



baseront sur la taille afin de décider si le text est un article vrai ou faux alors que ce n'est pas toujours le cas (la taille n'est pas un critère fiable).

- L'analyse de la taille du contenu texte des article a montré que les articles les plus longs sont des articles fake voir (annexe code 1)
- L'analyse des dates révèle que le marché de l'information est inondé de fake news depuis l'élection de Donald Trump (d'où sa grande présence dans le nuage de mots avec celui de Hillary Clinton et la maison blanche) et nous remarquons que durant 2016-2017 il y a eu plus de fake news que de vrai news.
- Nous avons bien sûr retiré la date afin de ne pas biaiser nos modèles, comme pour la taille la date pourrait influencer la décision du modèle, la date n'est donc pas un critère stable de décision et c'est pour celà qu'il ne fera pas partie des features de prédictions.
- Il existe un autre type de visualisations qui est la visualisation des thèmes, pour celà il existe des méthodes statistiques qui se basent sur la colonne de contexte de la base de données.

Cette visualisation a révélé que les bases de données traitent majoritairement de la politique même si les labels sont très différents. Ce qui nous pousse à faire une autre analyse au sens du NLP pour assurer cette différence de labels n'aura pas d'influence sur les prédictions des modèles

- Il existe une autre méthode, la méthode d'analyse de NLP appelée LDA permet d'avoir en sortie un dashboard capable de définir les différents contexte de chaque article (méthode de classification) il suffit de lui dire le nombre de contextes que nous voulons avoir et il s'occupe de la segmentation des données d'entrée.

Elle révèle que les données Fake et True traitent de sujets différents mais les mots employés sont redondants, en fixant à 5 le nombre de contexte, il semble qu'en réalité il y en ait beaucoup moins car les mots utilisés pour le contexte 5 sont presque les mêmes que pour le contexte 3 (dans les données fake), la segmentation s'est faite sur les zones géographiques (moyen orient, occident) ce qui nous.

Nous remarquons donc que les données parlent majoritairement de politique ou de religion et qu'elles ne sont pas très différentes au sens du NLP.

## Visualisations

Voici quelques visualisations statistiques afin de voir la bonne distribution des données vraies et fausses afin de ne pas biaiser l'apprentissage des modèles. Avant l'échantillonnage voici les histogrammes puis après l'échantillonnage nous aurons 20000 pour les fakes et les true (voir annexe visualisation 1)



*On remarque une bonne distribution des données*



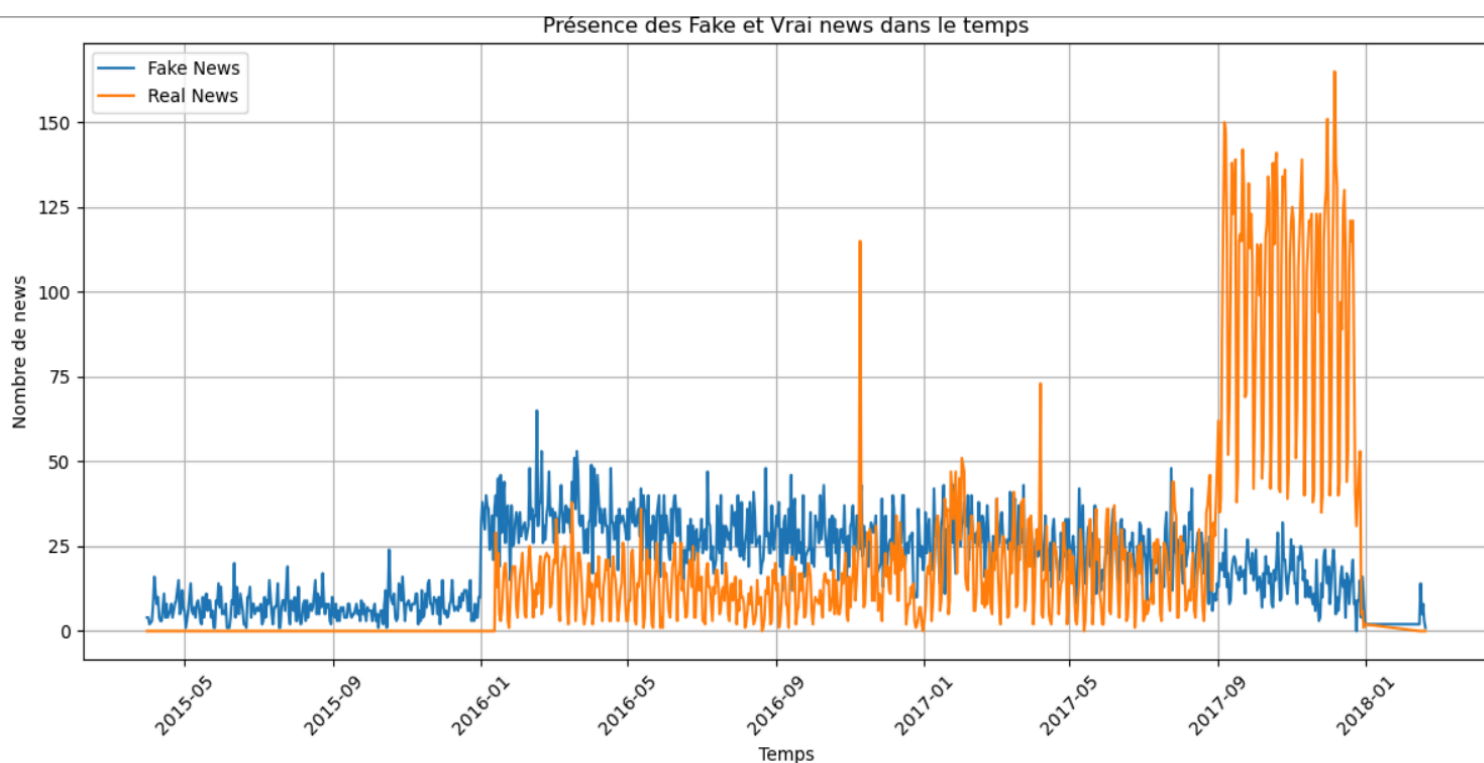
---



## Le nuage de mots des données vraies

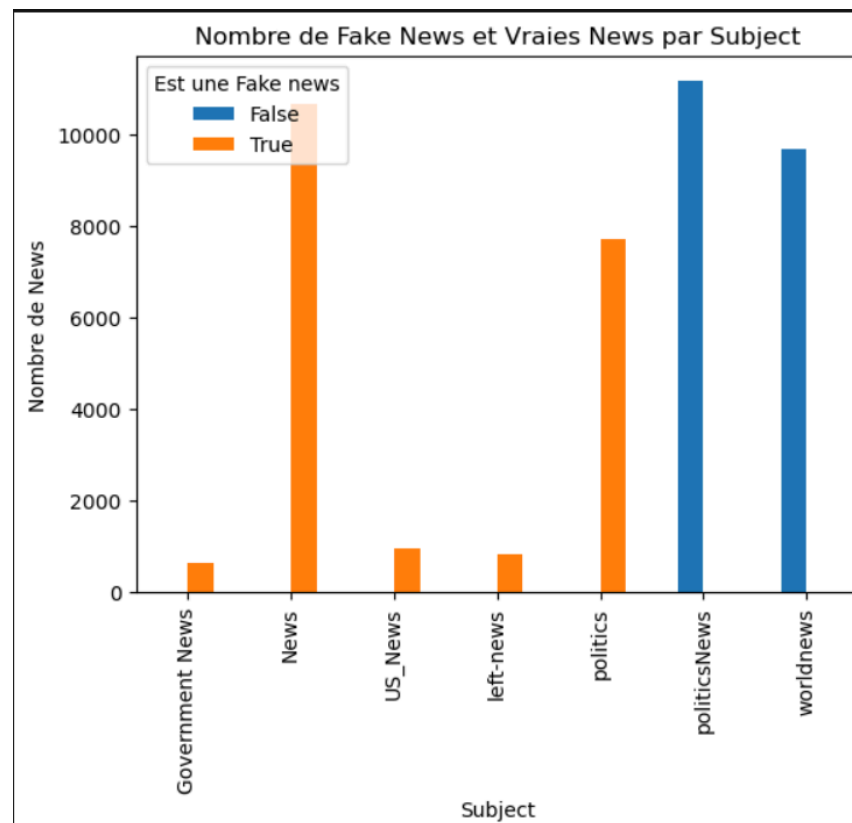
On remarque que pour les deux datasets il y a beaucoup de similarités en ce qui concerne les thèmes abordés ainsi que les mots employés (Donald Trump, Maison blanche...etc)

Les visualisations à l'aide de la colonne de temps nous ont menés à dire qu'on remarque les fakes news étaient à leurs apogées entre 2016 et fin 2017 (première année d'élection de Donald Trump) ce qui coïncide avec sa présence en force sur les nuages de mots.



Les visualisation temporelles labellisées

-Les méthodes de visualisation statistiques nous a permis d'avoir ce graphique

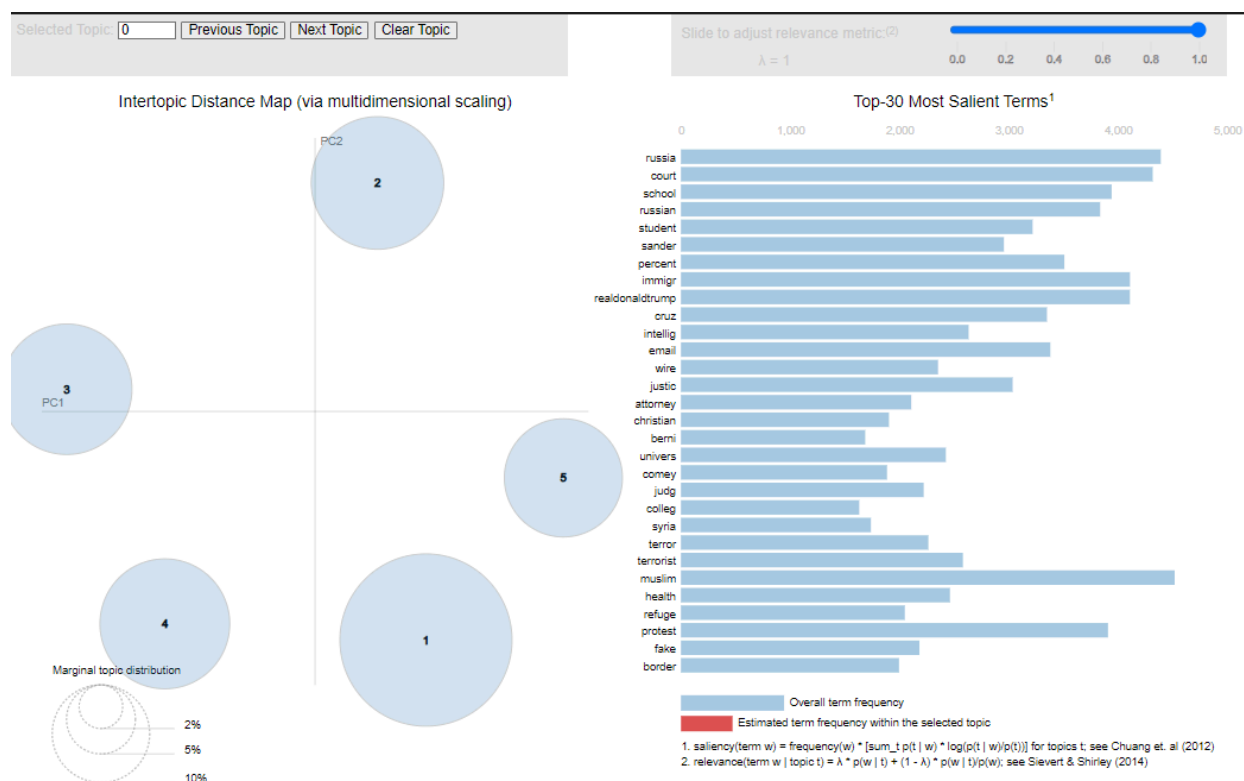


Ceci implique que la diversité des données fake est supérieure à celle des données réelles

Les données réelles quant à elles sont concentrées sur les thématique de la politique et des :un peu comme ce que nous voyons au journal télévisé ou ce que nous lisons dans les journaux, ils ont tendance à avoir une récurrence stable.

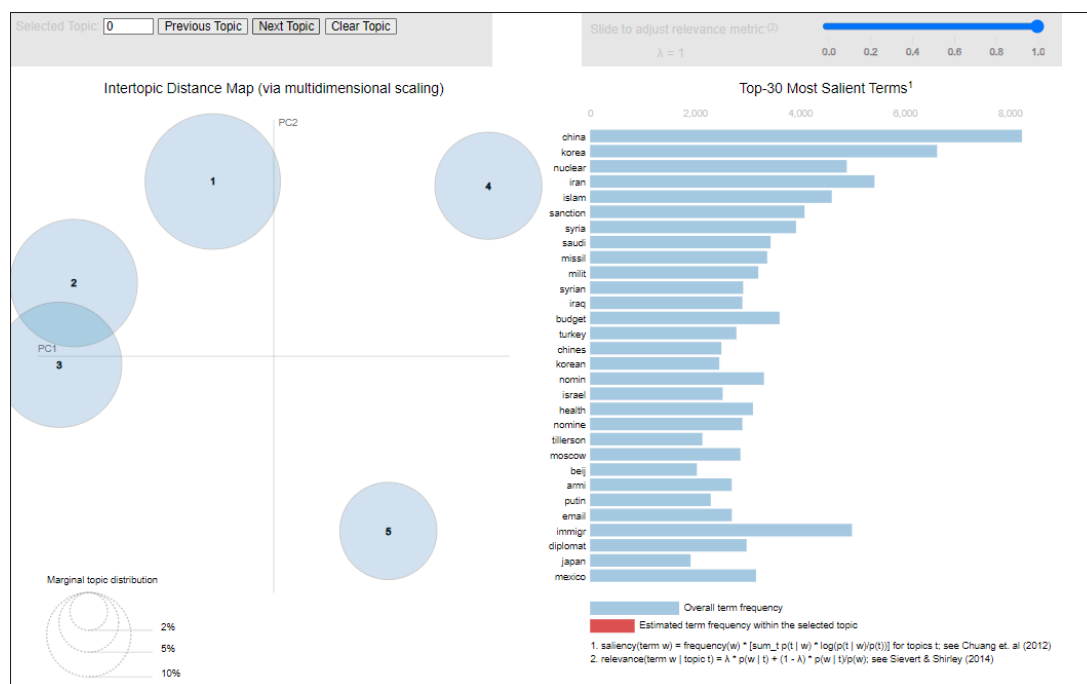
Quant aux données fake, leurs "marché" est dilué et ne se concentrent pas sur une thématique précise sur une longue période, ce sont des données aléatoires.

-La visualisation par LDA des données Fake ressemble à ce dashboard



Comme l'avait démontré l'histogramme précédent, les données fake ont tendance à être aléatoire et ne se concentrent donc pas sur des faits réels mais sont très divers ainsi que les mots utilisés comportent parfois des erreurs d'orthographe comme dans le mot "judg" au lieu de "judge"

-La visualisation par LDA des données True a donné cedashboard



En analysant La LDA des données réelles sous forme d'une ACP, on remarque qu'il y a moins de diversité et d'opposition entre les domaines traités comparé aux données fake. Ce qui confirme l'analyse statistique précédente.

Cette analyse nous a permis d'égaliser le nombre de contextes et de comparer les données true et fake sur une base plus objective qui n'inclut que la compréhension de l'ordinateur au sens du NLP

## Problématique

**Comment améliorer les détecteurs de fake news déjà existant?**


## Résultats

Avant de commencer de parler des modèles d'intelligence artificielle, nous allons commencer pas à pas et expliquer toutes les étapes théoriquement ainsi que les résultats obtenus suite à leur application à nos données.

## Prétraitement

Nous avons utilisé une fonction de prétraitement au sens du NLP capable de:

- Tokenizer: transformer les mots en vecteurs numériques compréhensibles
- Lemmatiser
- Faire le stemming ou racinisation: Éliminer les suffixes et le préfixes pour se rapporter à un espace de mots plus petit et plus facilement manipulable et exploitable
- Éliminer les stop words: les mots avec un nombre de lettres inférieur à 3 ou les mots comme "did" (faire) ou ceux qui n'ont pas d'influence sur le sens de la phrase
- Supprimer les tabulations
- Retirer la ponctuation



En résumé, cette fonction une fois appliquées aux données brutes (validation.csv, true.csv et fake.csv) elle renvoie un dossier clean data qui contient les

## Distance entre les données brutes et les données pré traitées

Nous ferons quand même attention à ce que le prétraitement n'ai pas eu trop d'impact sur le sens des phrases. Pour cela il suffit de calculer la distance au sens du NLP entre les données brutes et les données nettoyées. Nous pouvons ainsi maîtriser le seuil de "traitement" pour gagner en précision dans les modèles d'Intelligence artificielle (en éliminant les détails superflus) sans perdre le sens originel des phrases des articles.

(Voir annexe distance)

## Données de Validation:

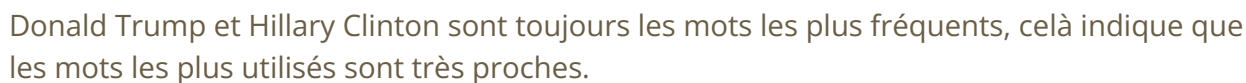
Afin d'évaluer les performances des modèles il est essentiel de tester sur des données autres que les données d'entraînement.

C'est pour cela que nous avons la base de données validation\_data.csv

Nous lui avons appliqué les mêmes traitements que les bases fake.csv et true.csv dans le but de ne pas confronter les modèles à des formats de données auxquels il ne s'étaient pas entraînés.

Il est essentiel d'analyser les données de validation.csv pour voir si les données sont distribuées de la même manière et si les données sont trop différentes des données d'entraînement.

Le nuage de mot est le suivant:



Selected Topic: 0 Previous Topic Next Topic Clear Topic

Slide to adjust relevance metric (2)

$\lambda = 1$

0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)

PC1

PC2

Marginal topic distribution

2%  
5%  
10%

Top-30 Most Salient Terms<sup>1</sup>

0 500 1,000 1,500 2,000 2,500 3,000 3,500

iran  
cruz  
nuclear  
russian  
syria  
rubio  
protest  
is  
israel  
deleg  
iowa  
islam  
china  
water  
corney  
weapon  
agreement  
syrian  
market  
negot  
hampshir  
putin  
romney  
peac  
climat  
kassich  
saudi  
ryan  
calicut  
sanction

Overall term frequency  
Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) \* [sum\_t p(t | w) \* log(p(t | w)/p(t))]] for topics t; see Chuang et. al (2012)  
2. relevance(term w | topic t) =  $\lambda \cdot p(w | t) + (1 - \lambda) \cdot p(w | t)p(w)$ ; see Sievert & Shirley (2014)



Il y a une bonne distribution des données par rapport aux 5 contextes par rapport à la map, cette représentation fait rappeler à un mélange des données fake et true.

Cela implique que les données de validation sont bien distribuées en ce qui concerne les thèmes abordés au sens du NLP

## Données Traitées (\_processed)

Nous avons traité les données brutes et avons stocké le résultat dans une colonne à côté des colonnes existantes, nous les avons nommées `title_processed` et `text_processed` (dans chacun des dataset)

mais aussi sur de nouveaux csv qui sont les fichiers suivants:

-True\_processed.csv

-Fake\_processed.csv

- validation\_clean.csv

## TF IDF tokenizer:

En plus de la tokenization que nous avons définie avec la fonction "from scratch" (voir annexe code processing). Il est possible de faire une tokenization avec le TF IDF qui est autre qu'une mesure d'évaluation des mots par rapport au text complet, en d'autres termes, le TF IDF calcule l'importance d'un mot dans le corpus et donne son "poids". C'est une approche pour associer une valeur numérique à chacun des mots présents dans la colonne de texte et qu'elle devient comprise par la machine.

Son but: Tester si d'autres tokenizations donneraient de meilleurs rendements par rapport aux modèles

Remarque: le TF IDF n'était pas la solution optimale pour les résultats, il est très long d'utilisation et ne permet pas d'avoir une bonne accuracy.

## BERT tokenizer:

Il est aussi possible de tokenizer avec la méthode BERT (Bidirectional Encoder Representations from Transformers) qui est un modèle de deep learning développé par google et qui permet comme tous les tokenizers d'associer une valeur numérique à chaque mot de la colonne texte afin que cette dernière soit comprise par nos ordinateurs.

Son but: Pouvoir varier les résultats et les interprétations.

Nous avons finalement remarqué que le BERT Tokenizer était le meilleur, car il est plus rapide et permet d'avoir une meilleur accuracy dans nos modèles.

## La courbe ROC-AUC

Dans le contexte d'une classification binaire (vrai/faux), il est essentiel d'utiliser la courbe ROC-AUC pour évaluer notre modèle. Nous allons examiner deux aspects :

- ROC (Receiver Operating Characteristic): Cette courbe illustre le rapport entre le taux de vrais positifs et le taux de faux positifs. Elle permet de visualiser la capacité du modèle à discriminer entre les deux classes.
- AUC (Area Under the Curve): C'est une mesure qui résume la courbe ROC en calculant l'aire sous celle-ci et au-dessus de la diagonale qui va du coin inférieur gauche au coin supérieur droit du graphique. Un score AUC proche de 1 indique que notre modèle fait des

prédictions presque parfaites. En revanche, un score proche de 0,5 suggère que nos prédictions sont principalement dues au hasard, ce qui signifie que le modèle n'apprend pas de manière optimale.

## Modèle 1: Arbre de décision

L'arbre de décisions est un modèle de machine learning qui permet, comme son nom l'indique de prendre une décision  $y$  (fake, true) à partir des features  $x$  (la colonne de contenu text), ce modèle permet d'avoir des résultats facilement interprétables.


Le modèle d'arbre de décision présente d'excellentes performances avec des paramètres optimisés, avec les paramètres suivants:

- critère de `gini`
- une profondeur maximale de `10`
- mode de division `best`.

Sur les données de test, il a une précision de 98.7%, et une précision de 98.56% sur les données de validation. Le modèle montre une précision (97.99%), un rappel (99.48%), et un score F1 de 98.73%, ce qui démontre une excellente capacité à équilibrer la précision et le rappel. Les taux de True Positive (50.37%) et True Negative (48.33%) sont élevés avec des très petites erreurs de False Positives (1.03%) et False Negatives (0.26%).

Le score ROC-AUC de 98.69% a une capacité à distinguer entre les classes positives et négatives quasi parfaites.

Cependant, en prenant compte l'évaluation sur un dataset différent révèle une baisse importante de la performance, avec une précision de 49.65%. On a aussi une augmentation



massive des faux positifs (48.51%), indiquant que le modèle a surpris sévèrement, il a donc une faible capacité de généralisation à de nouvelles données.

Le surajustement est typique des arbres de décision, particulièrement avec des paramètres qui permettent une complexité accrue (comme une profondeur maximale élevée), contrastant avec des modèles peut-être plus robustes mais moins performants sur les données de formation, comme la régression logistique qui a montré des signes de généralisation légèrement meilleurs. (Voir annexe visualisation: matrice de confusion decision tree et faux positifs)


## Modèle 2: Régression Logistique

Est un modèle de machine learning supervisé qui permet de prédire l'appartenance d'une donnée d'entrée à une classe, dans notre cas true / fake news.

voir annexe code régression Logistique

Le modèle de régression logistique a été optimisé avec des paramètres tels qu'un coefficient de régularisation  $c$  de 1 et un maximum de 500 itérations, qui atteint un accuracy de 74.16% sur les données de test, ce qui est cohérent avec son score de validation de 74.04%. On a donc une bonne performance globale avec un score F1 de 73.72%, indiquant un équilibre satisfaisant entre précision et rappel.

Nous avons de meilleures performances que les modèles passifs agressifs et le modèle naive bayes, ayant une précision de 72.73% lorsqu'il prédit une instance comme positive, et il parvient à identifier correctement 74.74% des vrais positifs. Le score ROC-AUC de 74.13% témoigne de sa capacité à distinguer efficacement les classes positives et négatives.



Mais comme dans les autres modèles, quand on l'utilise sur le modèle différent, la performance du modèle chute dramatiquement à une précision de seulement 52.53%, avec une augmentation des faux positifs (29.84%) et une baisse de précision des vrais positifs (31.71%).

Cette baisse de performance souligne des difficultés de généralisation et suggère que le modèle peut être surajouté aux données d'entraînement.

Pour améliorer la fiabilité et l'efficacité du modèle dans de nouveaux contextes, il est recommandé d'ajuster les hyperparamètres, d'améliorer le traitement des données, ou d'intégrer des techniques de régularisation plus strictes. Il faut que le modèle arrive à mieux généraliser.

### Modèle 3: Naive Bayes

Modèle probabiliste simple basé sur le théorème de Bayes ainsi que sur l'hypothèse de naïveté supposant que les caractéristiques sont indépendantes entre elles, il a la particularité d'être rapide mais pas assez performant comparé aux autres modèles (voir annexe pour la matrice de confusion ainsi que la courbe de ROC AUC)

Optimisation: Nous mettons en place un algorithme de recherche sur grille, ce qui nous permet de trouver les meilleurs hyperparamètres pour le modèle. Nous commençons avec `param_grid` comportant plusieurs paramètres à tester.

Le paramètre le plus important est `alpha`, qui est le paramètre de lissage pour le modèle. Il est utilisé pour gérer les données non vues ; plus le `alpha` est élevé, plus le lissage est important.

En termes simples, cela signifie que le modèle sera moins sensible aux données d'entraînement

## Modèle 4: Passif agressif

Quand on parle d'un modèle passif agressif, nous parlons d'un algorithme linéaire de classification et de régression. C'est un algorithme d'apprentissage en ligne, Il traite les données séquentiellement, ajustant le prédicteur à chaque nouvelle instance pour améliorer la prédiction future.

Le modèle se base sur deux principes:

- Passive: Si notre modèle prédit une instance juste, alors les poids et les biais ne seront pas changés. L'état actuel de notre modèle sera maintenu.
- Aggressive: Si notre modèle fait une mauvaise prédiction sur une instance reçue, les poids et biais de notre modèle seront changés par le paramètre de régularisation  $C$ , le but étant de minimiser l'erreur. Il change les paramètres pour que la prédiction actuelle soit 'juste'.

Le modèle utilisé par défaut est la fonction de coût hinge. Voici comment mathématiquement les poids et biais seront changés:

$$w_{\text{new}} = w + \alpha y x$$

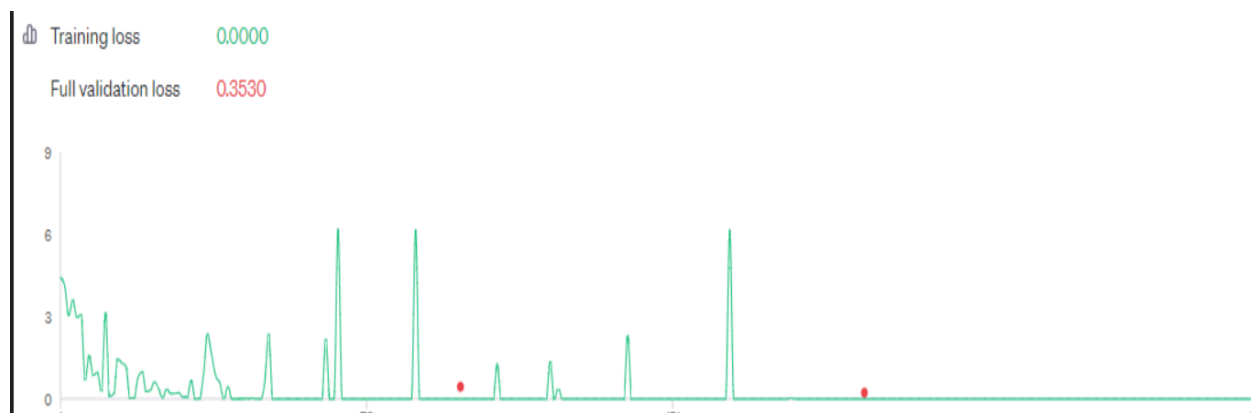
$$b_{\text{new}} = b + \alpha y$$

où  $w$  est le vecteur de poids,  $b$  est le biais,  $x$  est le vecteur de caractéristiques de l'instance mal classée,  $y$  est l'étiquette correcte de l'instance, et  $\alpha$  est le taux d'apprentissage calculé en fonction de l'erreur et du paramètre de régularisation  $C$

Comme nous traitons des données de texte, et que nous devons faire une classification binaire, il se va de dire que les text vont avoir des pattern différents, en utilisant le model passive agressif, nous pouvons entraîner le modèle à définir une méthode pour les identifier.

## Modèle 5: Open AI

Modèle de deep learning développé par le groupe Open AI qui a créé le célèbre chat bot chat gpt. Assez lourd nous n'avons pas pu avoir plus de résultats à part la visualisation suivante:



Il exprime la loss lors de l'apprentissage et la compare aux valeurs de la loss dans l'étape de validation, Il est clair que ce genre de model fonctionne mieux avec de plus grands flux de données.

## Benchmark

Rien de mieux qu'un tableau pour comparer objectivement les résultats obtenus  
(Nous avons préféré comparer les modèles parfaitement fonctionnels afin de ne pas alourdir la lecture)

Model	Accuracy avec le test	ROC-AUC
Arbre de décision	49.65%	98.69%
Régression logistique	59.61%	74.13%
Naive bayes	59%	61.4%
Passif agressif	49%	98%

Le but n'est pas de choisir un seul modèle mais de laisser toutes les informations au lecteur au consommateur de notre solution IA.

## Le vrAI

Le vrAI est le nom de notre IA qui n'est autre qu'une jauge qui ressemble au nutri score et qui pourrait être son analogue au sens du NLP, c'est à dire avoir en retour des notes de A à Fx comme à l'IA school mais dans le but de noter la justesse de l'article en se basant sur l'accuracy des models.

Une true news restera true quoiqu'il arrive, mais est ce que les données fausses sont fausses à 100%?



Pour chaque modèle nous pouvons multiplier la réponse binaire reçue en sortie par son accuracy.

Le 0 reçu pour les vraies news (indiquant que le modèle n'a pas détecté de fake) même multiplié par un grand nombre il restera 0.

```
vrAI

vrAI = 0
acc = logreg.score(tfidf_test, y_test)
if acc > 0.90:
    vrAI = 'Fx'
elif 0.70 < acc <= 0.90:
    vrAI = 'F'
elif 0.50 < acc <= 0.70:
    vrAI = 'E'
elif 0.30 < acc <= 0.50:
    vrAI = 'D'
elif 0.15 < acc <= 0.30:
    vrAI = 'C'
elif 0.05 < acc <= 0.15:
    vrAI = 'B'
else:
    vrAI = 'A'
```

## Challenges futures

Avec le temps la quantité de données augmentera, certaines fake news pourraient devenir vraies (comme l'annonce du décès d'une personne), il faudra repenser à réentraîner les données avec d'autres plus récentes et mises à jour et labellisées correctement.

**Migration sur le cloud:** ajouter des données sera un poids pour nos machines alors à défaut d'attendre 2h avant d'avoir une réponse de notre vrAI, on pourrait migrer vers le cloud afin d'y ajouter une touche de légèreté cela permettrait aussi de gagner en sécurité, mais aussi d'enregistrer nos expériences et observations sur le flux de MIFlow, de la même manière le passage au cloud permettrait l'utilisation d'autres modèles tels que fast text (facilement utilisable sur le cloud).

**Exploiter les données de titre:**

**Compter le nombre de !**

**Voir si l'auteur n'est pas un certain mister X**

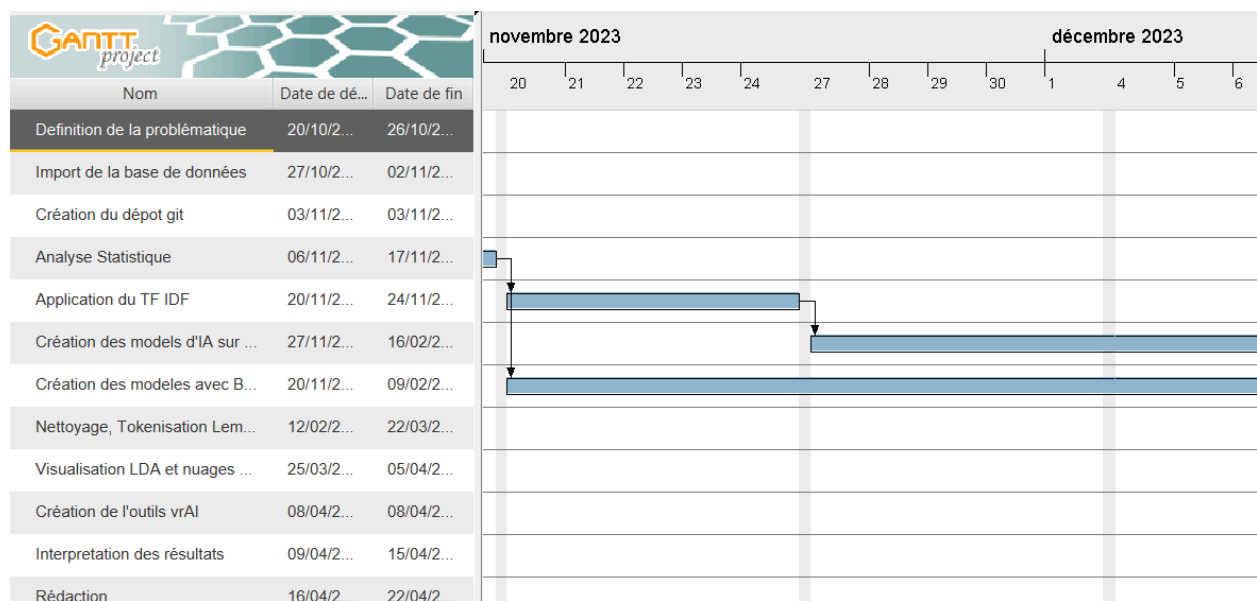
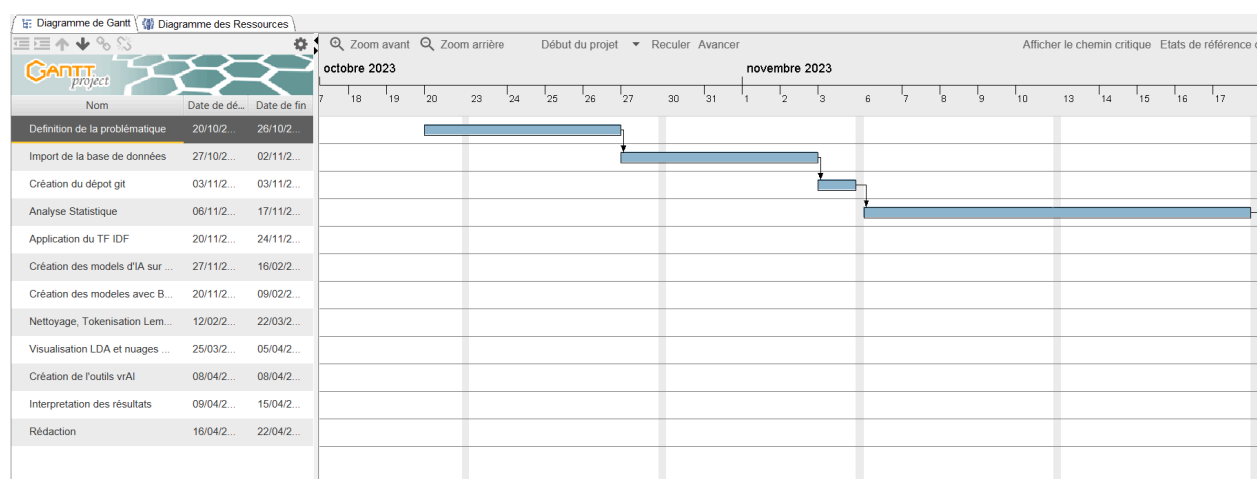
Investir sur des machines plus puissantes car les modèles ont besoin de toutes les ressources de l'ordinateur parfois pendant plusieurs jours

## Mise en place du projet

Pour ce projet IA nous avons décidé de répartir les tâches de manière à allier nos forces

Khedoudja Rym MERAD : Rédaction du support, Recherche et prétraitement des données, analyse au sens du NLP, benchmark des modèles d'IA et création du tableau true score, apport d'idées et théorie mathématique.

Mike DURAN: Recherche, implémentation et évaluation des modèles d'IA, résolution des conflits git, analyse au sens statistique, implémentation de la théorie au fur et à mesure.



-Vous trouverez l'intégralité du projet Gantt dans le rendu sous le nom de fichier



“Projet IA 2023-2024.gant”.

Nous avons fait en sorte de ne pas se limiter à seulement un rendu scolaire, et d’élargir l’application du projet à la vie professionnelle, c’est pour cela que nous continuerons d’améliorer ce projet même après avoir fait un premier rendu.

## Conclusion

Oui il est possible d’avoir un outil capable de donner au-delà d’une simple réponse binaire quand il s’agit de prédire la justesse d’un article!

Nous avons pu démontrer qu’il existait plusieurs manières de construire un score pour les articles en se basant sur les indicateurs de performance (matrice de confusion, accuracy...etc) des modèles d’IA comme l’arbre de décision, le bayes naïf, passif agressif, Régression Logistique ainsi que OpenAI afin de dire à quel point une fake news peut être fausse ou s’il est possible de déceler une lueur de vrai.

Il est tout à fait possible de répondre à la problématique si nous nous limitons à quelques modèles tels que decision tree malgré le fait que nous ayons rencontré quelques difficultés à la compilation de certains modèles car nos machines ne sont pas assez puissantes et donc le vrAI n’a pas encore pu être testé sur tous les modèles.

Cependant nous ne nous limitons pas au rendu du projet IA et pensons développer davantage le vrAI et pourquoi pas un jour le commercialiser.

## Glossaire

IA: Intelligence artificielle

Fake News: Nouvelles fausses

Deep Fake: Vidéos ou photos générées avec l'IA

NLP: Natural Language processing

## Annexe

### Annexe bibliographie

Introduction statistiques: <https://e-enfance.org/informer/fake-news/>

### Annexe code

Annexe code 1:

La taille des contenus fake vs la taille des contenus true

text length		text length	
count	20813.000000	count	20820.000000
mean	1535.007639	mean	1474.373871
std	1399.352505	std	1025.190753
min	5.000000	min	91.000000
25%	977.000000	25%	575.000000
50%	1326.000000	50%	1375.000000
75%	1795.000000	75%	1987.250000
max	32212.000000	max	15761.000000

Taille des titres des données fake vs taille des titres des données true

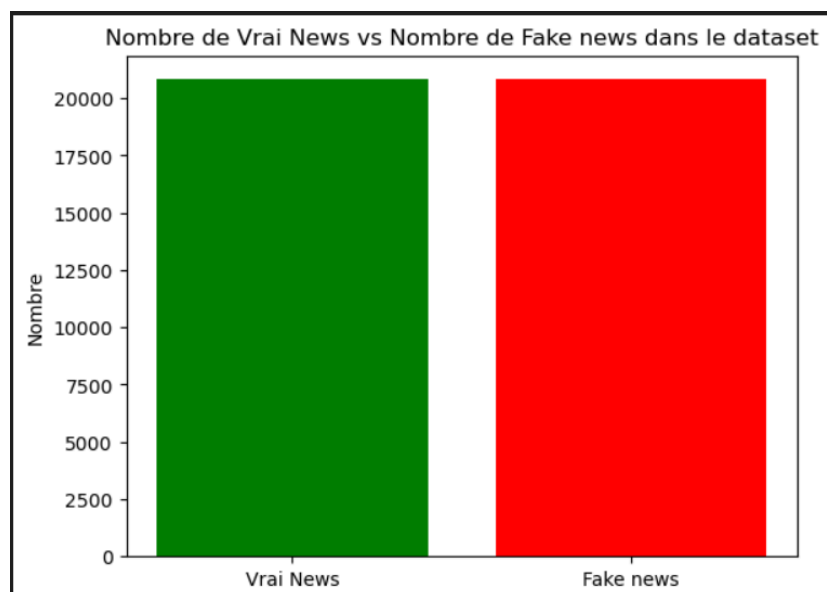
title length		title length	
count	20813.000000	count	20820.000000
mean	64.159948	mean	49.624448
std	18.751068	std	8.010538
min	8.000000	min	16.000000
25%	52.000000	25%	44.000000
50%	62.000000	50%	50.000000
75%	72.000000	75%	55.000000
max	222.000000	max	89.000000

Annexe régression Logistique

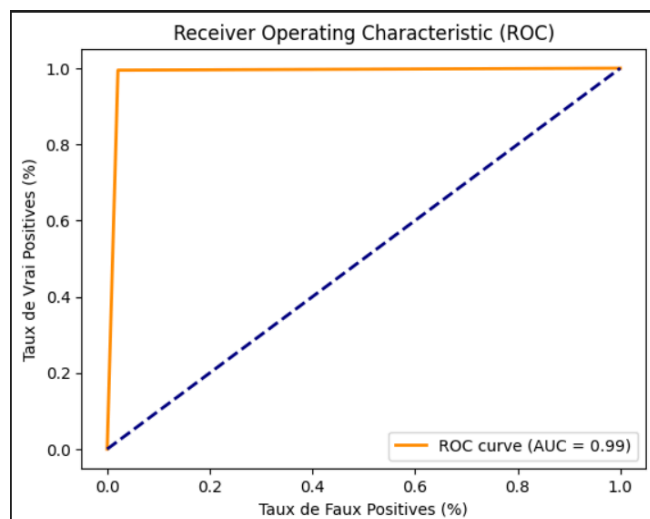
## Annexe visualisations

Annexe visualisation 1:

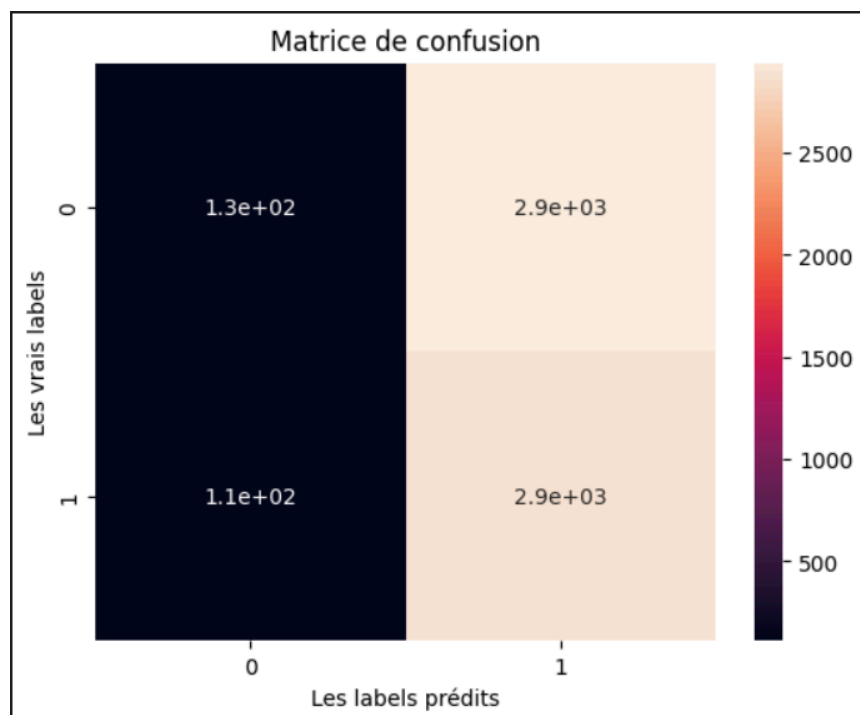
suite à l'échantillonnage nous obtenons naturellement 2000 données true et 2000 données fake:



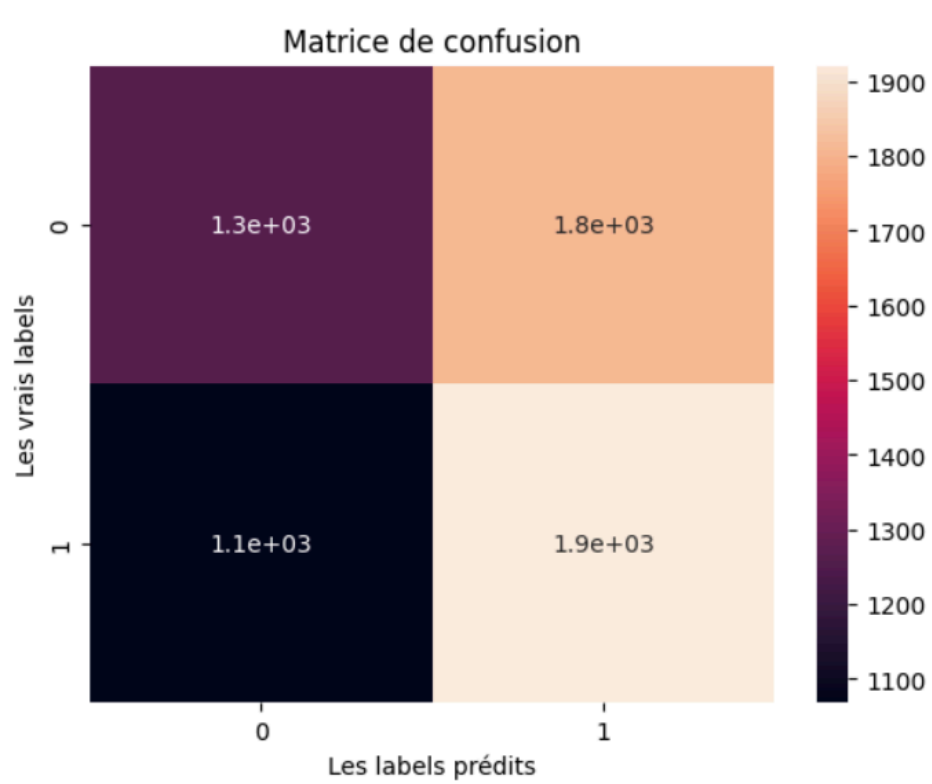
Taux de faux positifs avant ajustement des hyperparamètres:



Matrice confusion decision tree:

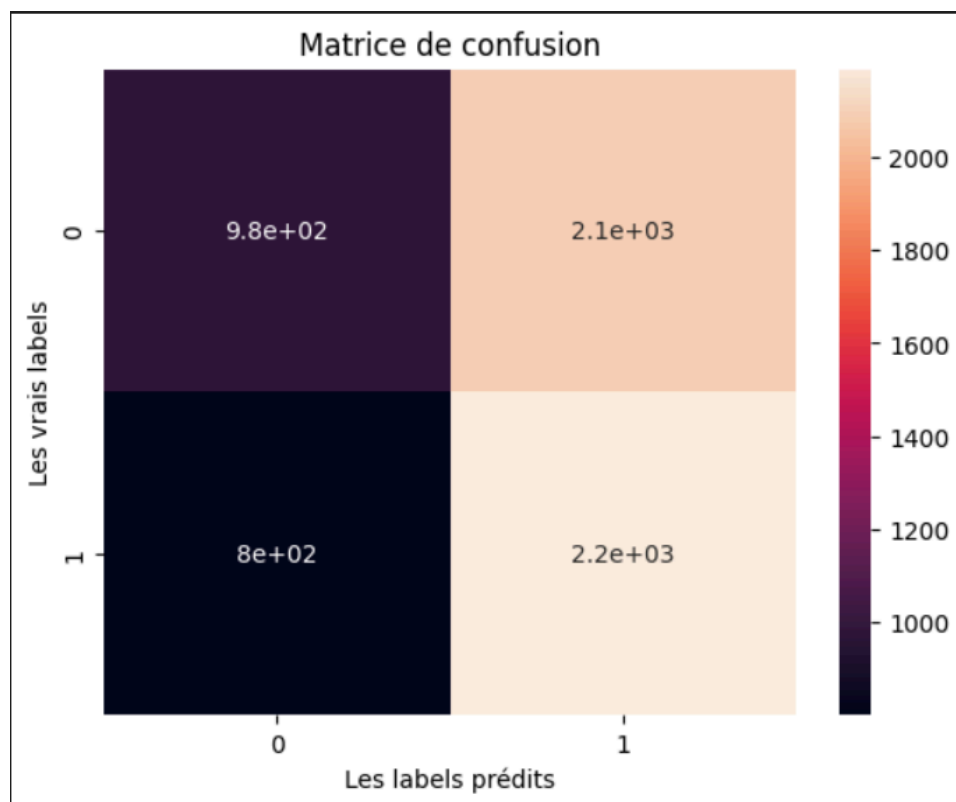


Matrice confusion Régression logistique:

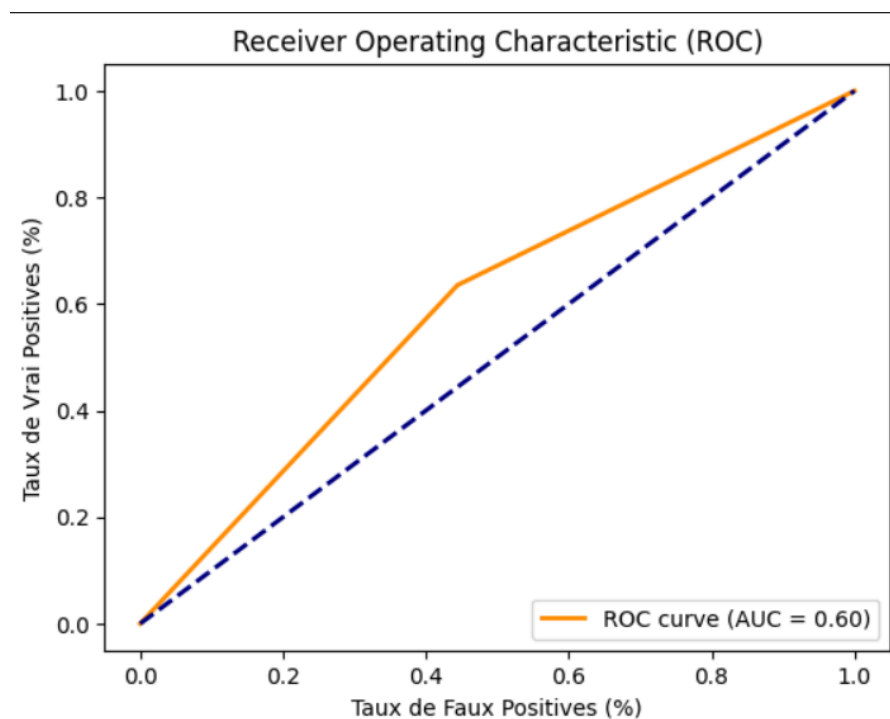




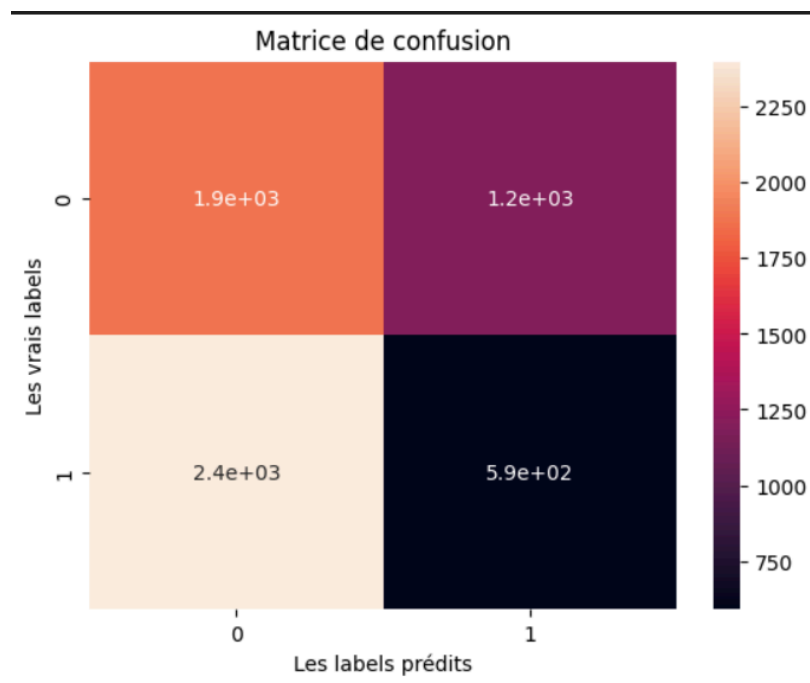
Matrice confusion Passif agressif:



-ROC AUC naïf Bayes



Matrice confusion Naïve Bayes:




## Remerciements

Nous voulions remercier à travers ce modeste travail tout l'ensemble de nos familles respectives ainsi que les membres du groupe GEMA et particulièrement à Dr.Carlos SUREDA qui a été un acteur principal de ce projet.

## Architecture du projet

Notre projet IA se compose de:

- Ce document
- Un fichier gantt project
- Un dossier data: Pour les données csv brutes true.csv, fake.csv et validation.csv

- 
- 8 Notebooks Models: une pipeline pour les modèles pré entraînés
  - Un notebook: Data analysis.ipynb (pour analyser les données)
  - Un dossier clean data: pour stocker les données prétraitées (généralisé par le code d'analyse)
  - Un ReadMe