

***Rapport de Projet IA***

Filière : Ingénierie des Systèmes Informatiques.

Par :

***AYOUB EL ALAMI et ILIASS BARGACHE***

Sous la direction de :

***Mme HIND OUAZZANI***

# ***Table des Matieres***

[***Table des Matieres*** 2](#_Toc135334609)

[***Table Des Figures*** 2](#_Toc135334610)

[***bibliotheques utilisés*** 4](#_Toc135334611)

[***Remerciement*** 5](#_Toc135334612)

[***Introduction*** 6](#_Toc135334613)

[Problématique 6](#_Toc135334614)

[Context 7](#_Toc135334615)

[***Chapitre 1 : DATA*** 8](#_Toc135334616)

[Introduction 8](#_Toc135334617)

[Collection des donneés 8](#_Toc135334618)

[Netoyage des données 9](#_Toc135334619)

[Nettoyage du contenu des tweets et traitement des données 9](#_Toc135334620)

[Vizualisation 10](#_Toc135334621)

[Conclusion 10](#_Toc135334622)

[***Chapitre 2 : MODELE*** 11](#_Toc135334623)

[Introduction 11](#_Toc135334624)

[Separation des donnees 11](#_Toc135334625)

[Creation de modele 12](#_Toc135334626)

[Algorithme d’apprentissage 13](#_Toc135334627)

[Utlilisant GridSearchCV 13](#_Toc135334628)

[Utilisation du modèle LinearSVC 15](#_Toc135334629)

[Utilisation de GridSearchCV avec SVC 17](#_Toc135334630)

[Conclusion 19](#_Toc135334631)

[***Chapitre 3 : INTERFACE*** 19](#_Toc135334632)

[Introduction 19](#_Toc135334633)

[Flask 20](#_Toc135334634)

[Interface 21](#_Toc135334635)

[***Conclusion*** 23](#_Toc135334636)

# ***Table Des Figures***

[Figure 1 : Data Set 8](#_Toc135334376)

[Figure 2 : Apres Nettoyage 9](https://d.docs.live.net/2d56c4ac9179941a/Document%201.docx#_Toc135334377)

[Figure 3 : Avant Nettoyage 9](https://d.docs.live.net/2d56c4ac9179941a/Document%201.docx#_Toc135334378)

[Figure 4 : plot\_Vizualisation 10](https://d.docs.live.net/2d56c4ac9179941a/Document%201.docx#_Toc135334379)

[Figure 5 : modele\_Function\_1 11](#_Toc135334380)

[Figure 6 : Modele\_function\_2 13](#_Toc135334381)

[Figure 7 : GrideSearchCV 14](#_Toc135334382)

[Figure 8 : GrideSearchCV\_Results 14](#_Toc135334383)

[Figure 9 : LinearSVC 16](#_Toc135334384)

[Figure 10 : LinearSVC\_Results 16](https://d.docs.live.net/2d56c4ac9179941a/Document%201.docx#_Toc135334385)

[Figure 11 : GridSearchCV\_with\_SVC 18](#_Toc135334386)

[Figure 12 : GrideSearch\_with\_SVC 18](#_Toc135334387)

[Figure 13 : Interface 21](#_Toc135334388)

[Figure 14 : Negative\_Result\_Interface 22](#_Toc135334389)

[Figure 15 : Positive\_Result\_Interface 22](#_Toc135334390)

[Figure 16 : Neutral\_Result\_Interface 22](#_Toc135334391)

# ***bibliotheques utilisés***

NLTK ( Natural Language Toolkit )

PANDAS

NUMPY

SEABORN

MATPLOTLIB

TEXTBLOB

SKLEARN

# ***Remerciement***

Avant tout, nous remercions le bon DIEU de nous avoir aidés à accomplir ce travail.

Nous tenons à saisir cette occasion et adresser nos profonds remerciements et nos profondes reconnaissances à :

Notre encadrant Mme HIND OUAZZANI

pour sa disponibilité, ses conseils, son orientation et ses encouragements tout au long de notre recherche.

***Introduction***

Problématique

Comment développer un modèle d'analyse de sentiment des tweets efficace et précis malgré les spécificités de la langue et les contraintes inhérentes aux tweets, afin de fournir des informations pertinentes pour comprendre les opinions et les attitudes des utilisateurs sur les réseaux sociaux ?

Nous sommes ravis de vous présenter notre projet d'analyse de sentiment des tweets à l'aide de l'intelligence artificielle. Dans le monde actuel où les réseaux sociaux occupent une place centrale, comprendre les sentiments et les opinions exprimés sur Twitter est devenu essentiel pour de nombreuses entreprises et organisations.

Notre objectif principal est de développer un système intelligent qui peut évaluer avec précision le sentiment des tweets, en les classant comme positifs, négatifs ou neutres. En utilisant les avancées récentes dans le domaine de l'apprentissage automatique et du traitement du langage naturel, nous souhaitons fournir une solution efficace pour analyser de grandes quantités de données textuelles en temps réel

Pour mener à bien notre projet, nous avons collecté un vaste ensemble de tweets en français, couvrant différentes thématiques et émotions. Nous avons ensuite entraîné notre modèle d'intelligence artificielle à reconnaître et à comprendre les nuances du langage utilisé dans les tweets afin de pouvoir prédire avec précision le sentiment exprimé.

## Context

Avec l'avènement des réseaux sociaux, les plateformes comme Twitter ont transformé la façon dont les individus communiquent et partagent leurs opinions en temps réel. Ces plateformes sont devenues des sources de données inestimables pour comprendre les préférences, les tendances et les opinions des utilisateurs. Dans ce contexte, l'analyse de sentiment des tweets, qui vise à déterminer si un tweet exprime un sentiment positif, négatif ou neutre, est devenue une discipline essentielle pour les entreprises, les chercheurs et les décideurs.

# ***Chapitre 1 : DATA***

## Introduction

Les données sont le pilier essentiel d'un projet d'IA, fournissant la matière première nécessaire pour l'entraînement et le perfectionnement des modèles. Sans des données de qualité et diversifiées, les systèmes d'IA manqueraient de fondations pour apprendre et effectuer des prédictions ou des décisions précises.

## Collection des donneés

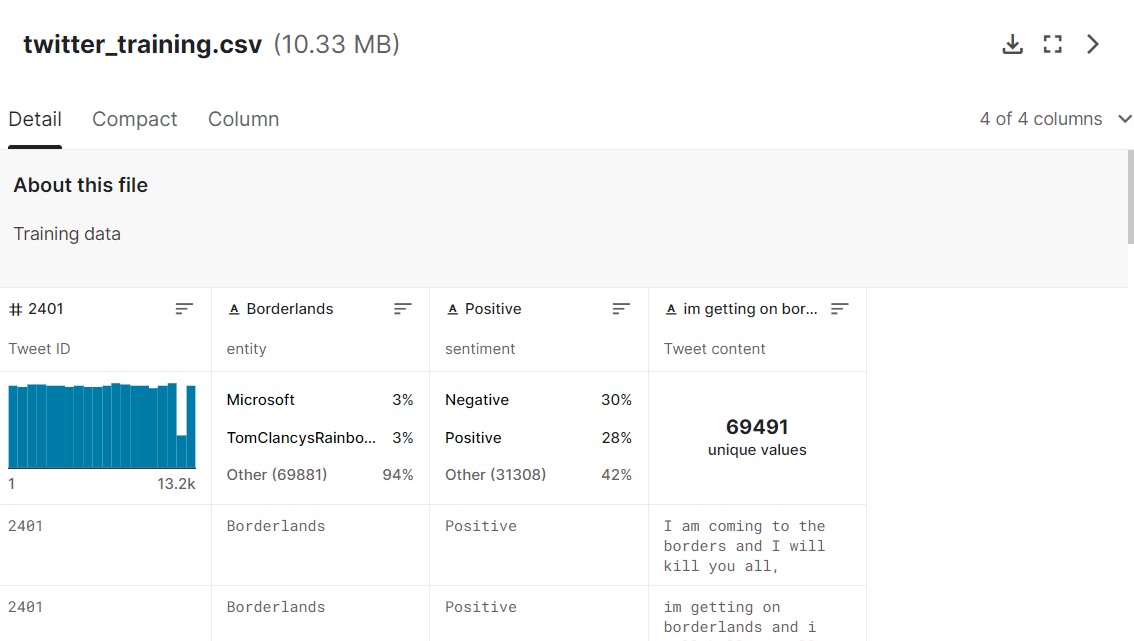


Figure : Data Set

<https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis>

Notre jeu de données contient 4 colonnes et 74682 lignes ,. la colonne 1 représente l'identifiant du tweet, la deuxième entité, le troisième sentiment et le dernier contenu du tweet

## Netoyage des données

Après la collection des données, nous avons commencé le prétraitement qui consiste à supprimer les données dupliquées ou qui n'ont pas de sens et d’ajouter des données manquantes .

Apres cette etape le nombre de données a diminuer de 74682 a 71656 lignes .

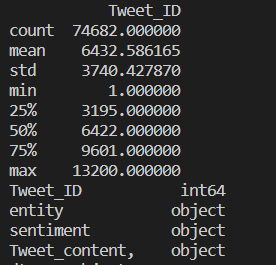
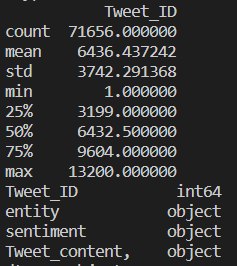


Figure : Apres Nettoyage

Figure : Avant Nettoyage

## Nettoyage du contenu des tweets et traitement des données

A l’aide de la biblioteque RE et NLTK(tokenize , stem,stopwords) on a reussi a creer une fonction qui effectue un prétraitement des données selon les etapes suivants :

* Mettre le texte en minuscules.
* Supprimer les liens URL et les mentions de sites web.
* Supprimer les caractères spéciaux tels que les symboles de ponctuation.
* Tokeniser le texte en mots en utilisant la méthode word\_tokenize de la bibliothèque NLTK .
* Filtrer les mots en supprimant ceux qui se trouvent dans (stop\_words) .
* Retourner le texte filtré sous forme d'une chaîne de caractères où les mots sont séparés par des espaces.
* Appliquer la fonction de stemming à notre données

Le stemming est un processus linguistique qui réduit les mots à leur forme de base ou à leur racine (appelée le "stem"). Cela permet de regrouper les mots apparentés et de réduire la dimensionnalité du vocabulaire

## Vizualisation

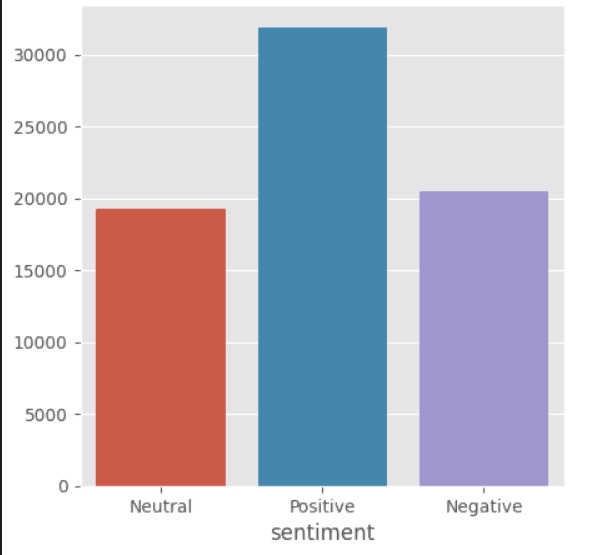
Apres les etapes precedantes une seul chose reste a faire est de vizualiser notre data a l’aide de la bilbioteque matplotlib pour une meilleure comprehension :

Figure : plot\_Vizualisation

## Conclusion

Dans ce chapitre, nous avons exploré les étapes cruciales de la collecte, du nettoyage et du traitement des données dans le cadre de notre projet d'IA. Nous avons souligné l'importance de rassembler des données de haute qualité et diversifiées pour garantir l'efficacité et la fiabilité de nos modèles d'IA.

Dans le prochain chapitre, nous plongerons dans la phase passionnante du développement, de l'entraînement et de l'évaluation du modèle

# ***Chapitre 2 : MODELE***

## Introduction

Dans ce chapitre, nous plongeons dans le monde fascinant de la création et de l'entraînement des modèles, où les données rencontrent les algorithmes pour donner naissance à des systèmes puissants et intelligents. Construire un modèle est une étape cruciale dans le domaine de l'apprentissage automatique, car c'est ce qui permet à une machine de comprendre et de généraliser à partir des données disponibles .

## Separation des donnees

Diviser les données en ensembles d'apprentissage et de test est une étape importante dans le développement de modèles d'apprentissage automatique. Cela permet d'évaluer les performances du modèle sur des données qu'il n'a pas encore vues.

Alors en a utiliser la fonction train\_test\_split pour une division aleatoire de notre données

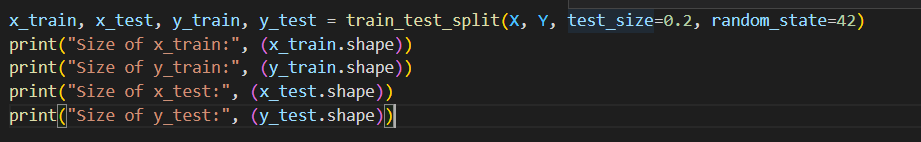


Figure : modele\_Function\_1

## Creation de modele

Dans cette partie on effectuer plusieurs etapes parmis lesquelles :

* Importation du module "warnings" et filtrage des avertissements pour les ignorer. Cela permet de masquer les avertissements affichés lors de l'exécution du code.
* Instanciation d'un objet "logreg" de la classe LogisticRegression, qui représente le modèle de régression logistique.
* Entraînement du modèle en utilisant les données d'apprentissage (x\_train et y\_train) à l'aide de la méthode "fit".
* Prédiction des étiquettes pour les données de test en utilisant la méthode "predict" du modèle entraîné.
* Calcul de l'exactitude (accuracy) du modèle en comparant les prédictions avec les étiquettes réelles à l'aide de la fonction "accuracy\_score" le resultat était 96.15%.
* Affichage de la matrice de confusion (confusion matrix) en utilisant la fonction "confusion\_matrix" .
* Affichage d'un rapport de classification (classification report) en utilisant la fonction "classification\_report" .

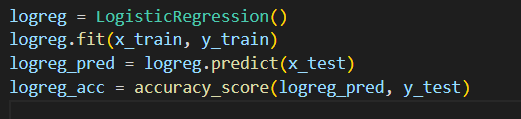


Figure : Modele\_function\_2

## Algorithme d’apprentissage

### Utlilisant GridSearchCV

Dans la premiere amélioration, nous utilisons la méthode GridSearchCV de la bibliothèque scikit-learn pour optimiser les hyperparamètres de notre modèle .

Nous définissons le param\_grid, qui est un dictionnaire contenant différentes valeurs de l'hyperparamètre C que nous voulons explorer.

GridSearchCV va alors effectuer une recherche exhaustive à travers ce grille de valeurs, en entraînant et en évaluant le modèle LogisticRegression avec chaque combinaison d'hyperparamètres.

Une fois la recherche terminée, nous utilisons grid.best\_params\_ pour obtenir les meilleurs hyperparamètres trouvés par GridSearchCV.

Ensuite, nous utilisons le modèle optimisé pour effectuer des prédictions sur les données de test , et nous calculons la précision (accuracy) en comparant les prédictions (y\_pred) avec les vraies étiquettes (y\_test).

Enfin, nous affichons la matrice de confusion (confusion\_matrix), qui montre le nombre de prédictions correctes et incorrectes pour chaque classe, ainsi que le rapport de classification (classification\_report), qui donne des mesures telles que la précision, le rappel et le score F1 pour chaque classe.

***Accuracy = 96.73 (0.58 improvement)***

***Code :***

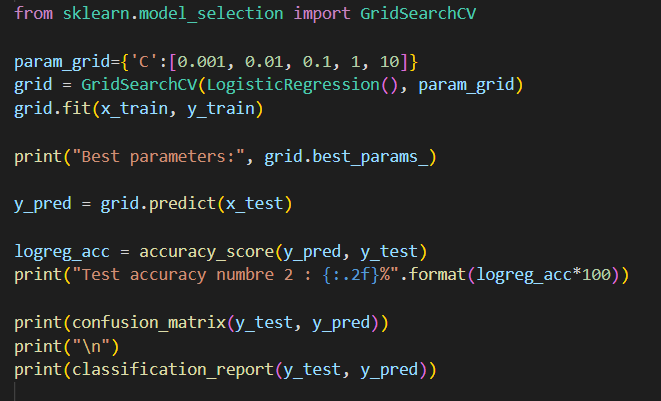


Figure : GrideSearchCV

***Result :***

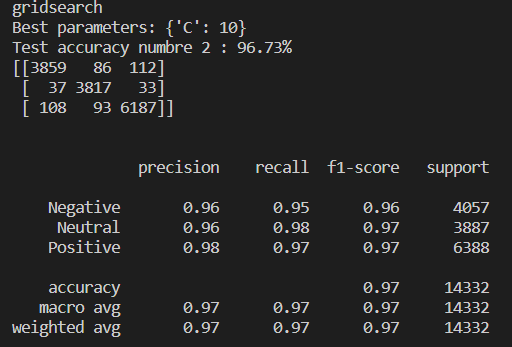


Figure : GrideSearchCV\_Results

### Utilisation du modèle LinearSVC

Dans cette amélioration, nous utilisons le modèle LinearSVC de la bibliothèque scikit-learn pour effectuer la classification. LinearSVC est un modèle de machine learning basé sur les machines à vecteurs de support (SVM) pour la classification linéaire.

Nous instancions un objet SVCmodel en utilisant LinearSVC(). Ensuite, nous entraînons le modèle en utilisant les données d'entraînement (x\_train, y\_train).

Une fois le modèle entraîné, nous effectuons des prédictions sur les données de test (x\_test) en utilisant la méthode predict. Nous calculons ensuite la précision (accuracy) en comparant les prédictions (svc\_pred) avec les vraies étiquettes (y\_test).

Enfin, nous affichons la matrice de confusion (confusion\_matrix).

L'utilisation du modèle LinearSVC dans cette amélioration nous permet de classifier les données et d'évaluer les performances du modèle en termes de précision. Les résultats de la matrice de confusion et du rapport de classification fournissent une évaluation plus détaillée des performances du modèle pour chaque classe.

***Accuracy = 96.95 (0.22 improvement )***

***Code***

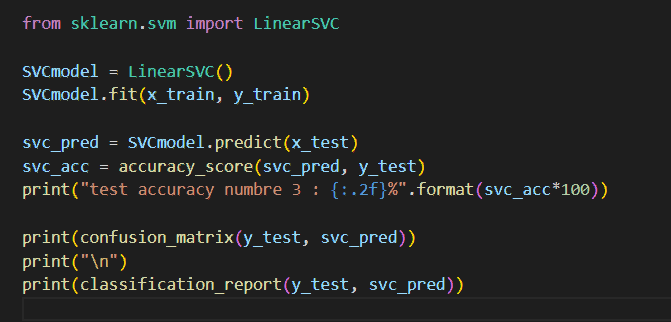


Figure : LinearSVC

***Result :***

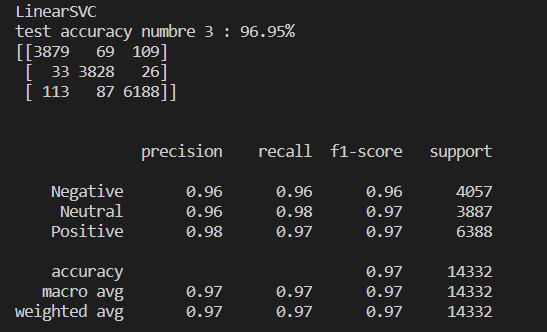


Figure : LinearSVC\_Results

### Utilisation de GridSearchCV avec SVC

Dans cette amélioration, nous utilisons à nouveau la méthode GridSearchCV de la bibliothèque scikit-learn pour optimiser les hyperparamètres du modèle SVC

Nous définissons le param\_grid, qui est un dictionnaire contenant différentes valeurs des hyperparamètres que nous souhaitons explorer. Les hyperparamètres inclus sont 'C' pour la régularisation, 'kernel' pour la fonction noyau, 'degree' pour le degré du noyau polynomial et 'gamma' pour le coefficient du noyau.

GridSearchCV effectue une recherche exhaustive à travers ce grille de valeurs, en entraînant et en évaluant le modèle SVC avec chaque combinaison d'hyperparamètres.

Une fois la recherche terminée, nous utilisons grid.best\_params\_ pour obtenir les meilleurs hyperparamètres trouvés par GridSearchCV.

Ensuite, nous utilisons le modèle optimisé pour effectuer des prédictions sur les données de test (x\_test), et nous calculons la précision (accuracy) en comparant les prédictions (y\_pred) avec les vraies étiquettes (y\_test).

Enfin, nous affichons la matrice de confusion (confusion\_matrix),

L'utilisation de GridSearchCV avec le modèle SVC nous permet d'optimiser les hyperparamètres pour améliorer les performances de classification. Les résultats de la matrice de confusion et du rapport de classification fournissent une évaluation plus détaillée des performances du modèle pour chaque classe.

***Accuracy = 96.95 ( no improvement )***

***Code :***

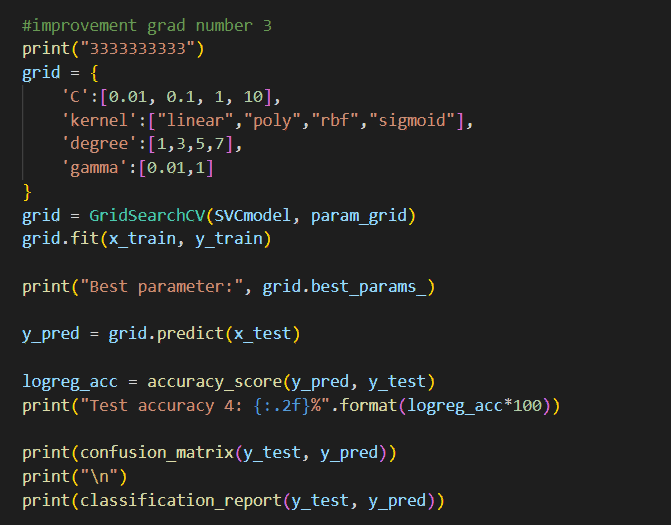


Figure : GridSearchCV\_with\_SVC

***Result :***

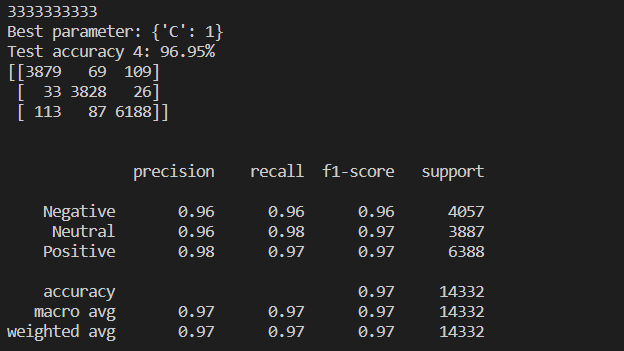


Figure : GrideSearch\_with\_SVC

## 

## Conclusion

En résumé, la création et l'entraînement des modèles sont des étapes cruciales pour obtenir des systèmes d'intelligence artificielle performants. Comprendre les concepts et les techniques liés à la création de modèles et à l'optimisation des hyperparamètres nous permet de développer des modèles plus précis et efficaces. Ce chapitre nous a fourni une base solide pour explorer plus en profondeur les diverses techniques et avancées dans le domaine de l'apprentissage automatique.

# ***Chapitre 3 : INTERFACE***

## Introduction

L'interface graphique joue un rôle crucial dans la réussite d'un tel projet, car elle facilite l'interaction entre les utilisateurs et le modèle d'IA. Dans cette section, nous explorerons les raisons pour lesquelles nous avons choisi de développer une interface graphique pour notre projet

## Flask

Flask est un framework web léger et flexible écrit en Python. Il est largement utilisé pour développer des applications web, y compris des interfaces graphiques, en raison de sa simplicité et de sa facilité d'utilisation. Flask suit le modèle de conception MVC (Modèle-Vue-Contrôleur) et est connu pour sa simplicité et sa facilité d'apprentissage.

grâce à flask , nous avons pu lier notre interface graphique qui a été faite en html et css pour la lier avec notre modele

## Interface

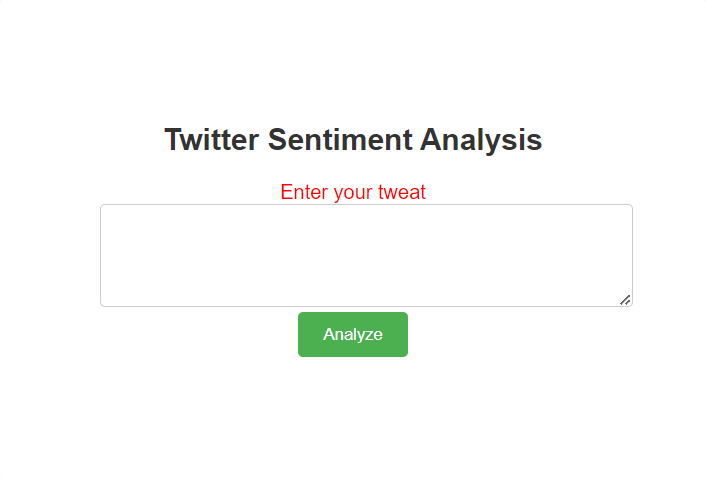


Figure : Interface

L'interface est composé d’une champ de texte ou on va ecrire ou coller notre tweet ou commentaire et d’un button qui vas lance l’action et montre le resulta ci-dessous:

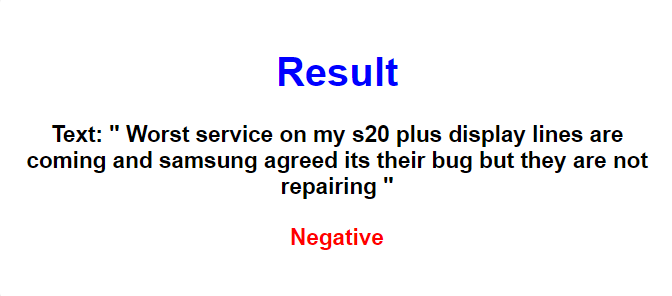


Figure : Negative\_Result\_Interface



Figure : Positive\_Result\_Interface

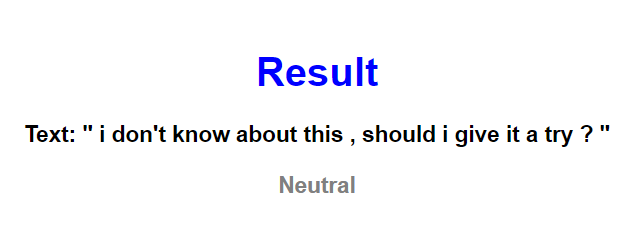


Figure : Neutral\_Result\_Interface

## ***Conclusion***

En conclusion, notre projet a démontré l'efficacité de l'analyse de sentiment basée sur l'apprentissage automatique pour prédire le sentiment des textes.

Cependant, il reste des opportunités d'amélioration, notamment en explorant d'autres algorithmes et en ajustant les paramètres du modèle pour améliorer la précision globale et la capacité de prédiction des sentiments neutres.

Ce projet a permis de mettre en évidence l'importance de l'analyse de sentiment dans divers domaines tels que le marketing, les médias sociaux et le service client.

Les résultats obtenus peuvent être utilisés pour prendre des décisions éclairées et mieux comprendre le ressenti des utilisateurs. Dans l'avenir, une expansion de ce projet pourrait inclure une analyse de sentiment en temps réel et une exploration de données supplémentaires pour améliorer davantage les performances du modèle.