Université de Versailles Saint-Quentin-en-Yvelines
UFR des sciences
Département d'Informatique

Compte rendu Final

10 février 2023

Data mining

Encadré par
Madame GARCIA Zaineb

Réalisé par
BOUDJEMAI Ali
AOUCHICHE Salah
HADJ MAHFOUD Kenza

Table des matières

1 Introduction

Dans le cadre de notre projet du module fouille de données nous sommes amenés à effectuer un projet en text mining. Aujourd'hui, Twitter est utilisé par des centaines de millions de personnes dans le monde entier. Le contexte de ce dernier est de mettre en place le processus Text mining qui consiste à analyser des tweets afin d'en extraire des informations liées aux opinions et aux sentiments (Sentiment Analysis) et à identifier les informations subjectives d'un tweet pour extraire l'opinion de l'auteur.

L'objectif de ce projet est de réaliser une analyse exploratoire et visuelle des tweets présents dans notre jeu de données. Dans un second temps, le but sera de parvenir à classifier à l'aide de différents modèles Machine Learning les sentiments des tweets selon. Autrement dit réunir sentiment analysis et NLP.

Le processus consiste à analyser les données des tweets qui sont principalement décrites dans un format non structuré et ne peuvent donc pas être traitées que par machine, pour en extraire des informations utiles. Il se décompose en plusieurs étapes. Au cours de la première étape, les données textuelles sont nettoyées et préparées pour l'analyse. Ensuite, elles sont converties en une représentation numérique qui peut être utilisée pour les algorithmes d'apprentissage automatique. Les algorithmes sont utilisés pour effectuer des tâches telles que la classification de textes, l'extraction d'informations et l'analyse de sentiments. Enfin, les résultats sont interprétés pour en tirer des conclusions.

1.1 Choix du sujet :

Notre choix s'est porté sur l'analyse de sentiment qui vise à extraire des émotions et des sentiments des tweets et mesurer l'impact des messages sur les réseaux sociaux. L'analyse de sentiment peut être définie comme une application particulière de la fouille de données, elle consiste à extraire et à évaluer les sentiments exprimés dans du texte en utilisant des techniques d'apprentissage automatique et de traitement du langage naturel.

Le but de ce choix sera d'utiliser les dernières techniques en text mining pour développer une solution performante pour l'analyse de sentiment.

1.2 Informations sur l'ensemble de données :

Notre jeu de données est un ensemble de tweets extraits à l'aide de l'API Twitter. Les tweets ont été annotés. L'objectif de notre modèle est de déterminer l'orientation sentimentale du texte saisi.

Ce jeu de données recensant au total **(31962 tweets)**, il contient 3 colonnes :

- **id :** représente l'identifiant du tweet.
- **tweet :** représente le contenu d'un tweet.
- **label :** représente l'attribut de classification, elle attribue un **1** si le sentiment du tweet est **(Sexiste/Raciste)** et **0** dans le cas contraire **NON(Sexiste/Raciste)**.

2 Processus d'analyse des sentiments

2.1 Extraction des données :

Les données sont collectées à à l'aide de l'API Twitter, qui sont ensuite stockées dans un fichier **csv**. La qualité des données collectées a un impact direct sur la qualité des résultats de l'analyse de sentiments, car elles contiennent généralement beaucoup de bruits et des éléments inutiles. Il est donc important de veiller à ce que les données soient fiables et représentatives. Une fois les données extraites, elles seront ensuite préparées pour le Prétraitement.

2.2 Text pré-processing :

Comme mentionné dans l'étape précédente, les données devraient être fiables et représentatives afin qu'elle soient prêtes pour l'analyse. Donc cette étape de pré-processing permet de réduire le bruit comme par exemple des mots vides ou des mots qui ne sont pas pertinents pour le cours de l'analyse, dans lequel contribue à s'améliorer les performances du classificateur et à accélérer également le processus de classification. Donc la stratégie qu'on va appliquer dans cette étape de pré-processing est la suivante :

2.2.1 Tokenization

Tokenisation est le processus de décomposition d'une chaîne de texte en plus petits morceaux appelés tokens.La tokenization permet de travailler avec des éléments plus petits et plus significatifs, ce qui facilite le traitement et l'analyse ultérieurs des données textuelles.

2.2.2 Removal of HTML tags :

Cette méthode consiste à supprimer les balises HTML pour récupérer le texte lisible et éliminer les distractions.
Cette étape est souvent nécessaire dans un processus de text mining car les balises HTML peuvent perturber les analyses de données textuelles en fournissant des informations superflues ou en altérant la structure du texte et fausser les résultats.

2.2.3 Delete special characters :

Cette méthode consiste à éliminer les caractères spéciaux présents au niveau des tweets qui sont inutiles ou perturbateurs pour le traitement ultérieur des données (Ponctuation, chiffres, Émojis et symboles graphiques..).
Cette étape permettra d'améliorer la précision et la fiabilité des analyses et des modèles.

2.2.4 Removal of strings :

Cette étape consiste à supprimer les chaînes inutiles dont les chaînes vides de notre ensemble de données.
les mots inutiles peuvent causer des problèmes tels que des comptes de mots incorrects ou des résultats d'analyse de sentiments biaisés.En conséquence, cette étape permettra d'améliorer la qualité de nos résultats.

2.2.5 StopWords :

StopWords est une méthode qui consiste à supprimer les mots couramment utilisés appelés stopwords qui sont considérés comme ayant peu de signification sémantique, cela inclut des mots tels que "the", "and", "of", "to", "in", etc.

En enlevant les stop words, on peut se concentrer sur les mots qui ont une signification plus profonde et qui peuvent être utilisés pour analyser les tweets. Cela peut aider à améliorer l'efficacité, avoir une analyse plus significative et à obtenir des résultats plus précis et pertinents.

2.2.6 Spell checker :

Cette méthode permet de corriger les erreurs d'orthographes de nos tweets en utilisant divers techniques telles que TextBlob qui utilise le correcteur d'orthographe PyEnchant derrière pour procéder à cela.

Les fautes d'orthographes peuvent altérer la qualité des données et rendre les analyses et les traitements ultérieurs moins fiables.

2.2.7 Lemmatization :

La lemmatization est une technique qui consiste à regrouper les différentes formes morphologiques d'un mot (formes plurielles, les temps verbaux, etc.) en une forme de base appelée "lemme".

Le but de la lemmatisation est de réduire les mots pour obtenir une représentation plus standardisée et normalisée de nos données.

Le stemming se concentre sur la forme morphologique d'un mot contrairement à La lemmatisation qui aide à mieux comprendre le sens sémantique et qui offre une précision accrue pour l'analyse de sentiments des tweets, c'est pour cela que nous avons choisi la Lemmatization.

2.2.8 Abréviation :

Les abréviations peuvent changer la signification d'un mot ou d'une phrase, ce qui peut affecter la classification du sentiment.

Cette méthode consiste à remplacer les abréviations des mots par leurs formes complètes, cette étape peut être réalisée en utilisant des algorithmes de reconnaissance de formes.

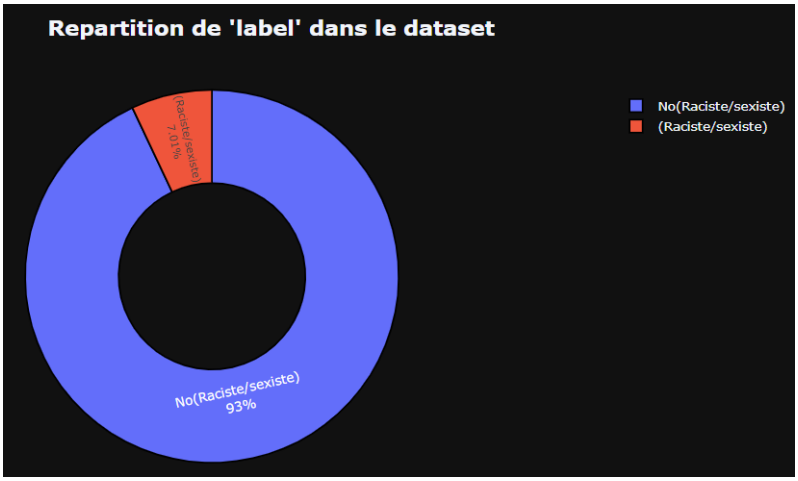
Dans notre cas, après vérification nous avons constaté que nous n'avons pas de mots abrégés à remplacer.

2.3 Visualisation des données Après le pré-traitement :

Après le pré-traitement des données, il est important de visualiser les données pour comprendre leur distribution, comme par exemple leur distribution de la fréquence. Cela peut aider à mieux comprendre les données et à mieux préparer les algorithmes d'apprentissage en machine pour les traiter.

2.3.1 Répartition des données selon le label :

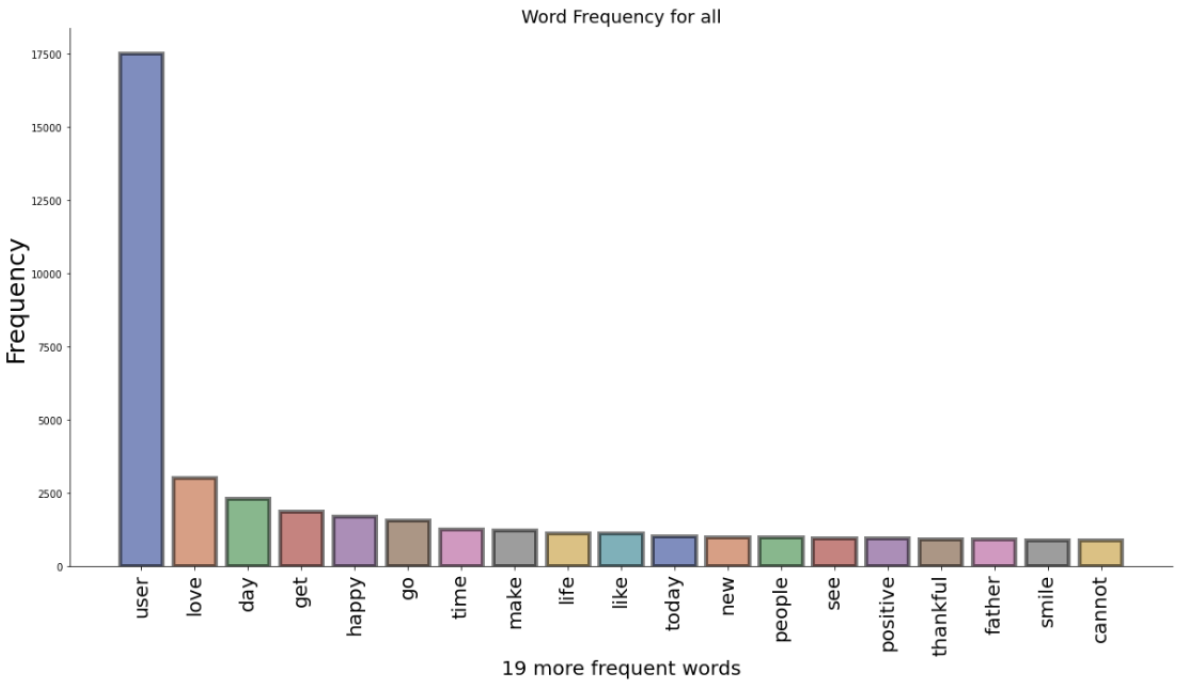
La figure suivante représente la distribution des données en fonction du label.



En visualisant la répartition du label au niveau de notre dataset, on remarque qu'il y a un déséquilibre au niveau de la répartition des données.
En effet on peut voir que pour la valeur 0 (No Raciste/sexiste) on a 93% et les 7% restant pour la valeur 1 (Raciste/sexiste).

2.3.2 La fréquence des mots :

La figure suivante represente les 19 mots les plus fréquent au niveau des tweets.



On remarque que les mots qui reviennent le plus sont user avec une fréquence de 2700 occurrences. love avec une fréquence de 2600 occurrences. et en troisième position on trouve le mot day avec 2500 occurrences.

2.3.3 La fréquence des mots (No) sexiste/raciste :

Les deux figures suivantes montre les mots No sexiste/raciste et sexiste/raciste les plus fréquents dans l'ordre définis.

On peut remarquer que le mot User,love et happy sont les plus fréquents pour la classe No raciste/sexiste.

Pour la figure 2, les mots les plus fréquents sont User,nothing etc.

FIGURE 1 – La fréquence des mots No raciste/sexiste (à gauche)

FIGURE 2 – La fréquence des mots raciste/sexiste (à droite)

2.3.4 Suppression des mots les plus fréquents inutiles :

La suppression des mots les plus fréquent qui ne sont pas utile, autrement dit supprimer les mots les plus fréquents qui ont une corrélation faible avec le label ou bien avec l'attribut de classification.

Comme constaté sur les deux figures précédentes, on voit bien que le mot 'User' c'est le mot le plus fréquent dans les deux classes, donc nous avons procédé à sa suppression car il a pas d'influence sur la classification de nos données et il va pas aider à déterminer le sentiment du tweet.

2.4 Word Embeddings :

Les thecniques de Word Embeddings, est une étape crusial dans le processus de text mining. Elle consiste à convertir les mots ou les phrases en un format numérique qui peut être utilisé par les algorithmes de traitement du langage naturel. Cela permet de représenter les données textuelles sous forme de vecteurs, ce qui les rend plus faciles à analyser.

Il existe plusieurs techniques de Word Embeddings, parmi ces dernieres nous avons choisi d'utiliser la thecnique TF-IDF qui consiste à mesurer l'importance des mots dans un tweet en fonction de leur fréquence d'apparition (TF) et de leur rareté par rapport à l'ensemble des documents (IDF). Cela permet de capturer les mots qui sont pertinents pour l'analyse des sentiments, en particulier ceux qui peuvent être considérés comme des indicateurs de sentiments, tels que les mots **(Raciste/Sexiste)** et **NON(Raciste/Sexiste)**.

L'utilisation de TF-IDF dans l'analyse des sentiments peut aider à améliorer la précision des résultats en considérant les mots clés pertinents pour le sujet spécifique et en les pondérant en fonction de leur fréquence d'apparition et de leur pertinence relative. Enfin, TF-IDF est souvent utilisé en conjonction avec d'autres algorithmes de traitement du langage naturel pour obtenir des résultats plus précis et robustes.

3 Classification des sentiments

L'analyse automatique des sentiments plonge dans le texte et en extrait des données exploitables. Plutôt que de se baser sur des règles prédéfinies, l'analyse automatique des sentiments utilise le machine learning pour comprendre les grandes lignes d'un message. Cette approche automatique utilise des algorithmes de classification par machine learning supervisé, ce qui augmente le niveau de précision et d'exactitude et permet de traiter rapidement les informations en fonction d'une série de critères.

3.1 Choix des algorithmes Machine Learning

L'analyse des sentiments utilise des algorithmes de machine learning pour explorer les données.

Le choix des algorithmes de machine learning pour l'analyse des sentiments est crucial car il peut avoir un impact significatif sur la précision et la fiabilité des résultats. En effet, différents algorithmes peuvent être mieux adaptés à différents types de données et de tâches d'analyse de sentiments.

certains algorithmes peuvent être mieux adaptés pour les données structurées, tandis que d'autres peuvent être plus adaptés pour les données non structurées, telles que le texte.

Il est donc important de comprendre les différences entre les algorithmes et de choisir celui qui convient le mieux à la tâche spécifique et aux données à analyser pour garantir des résultats précis et fiables et de s'assurer qu'ils ont été correctement entraînés et évalués.

Pour notre jeu de données, nous avons utilisé deux algorithmes supervisés qui sont :

- KNN
- DecisionTre

Nous avons choisi ces deux algorithmes en raison de leurs capacité à traiter efficacement des données hétérogènes telles que des textes ce qui les rend particulièrement utiles pour l'analyse des sentiments où les données ou bien les tweets sont souvent de nature textuelle.

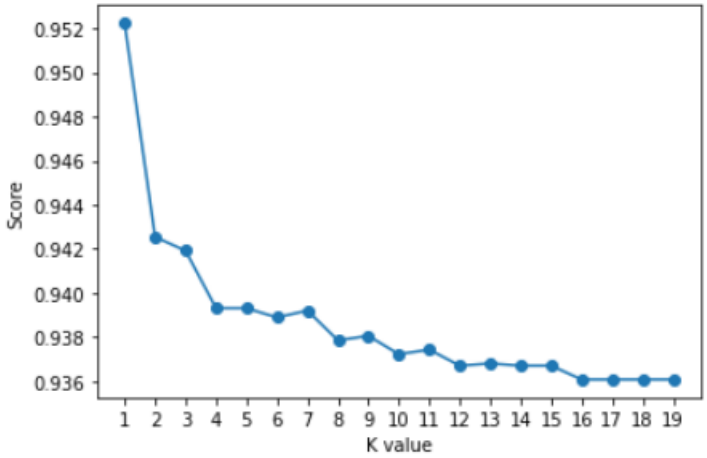
3.2 KNN

KNN est un algorithme de machine learning souvent utilisé pour l'analyse des sentiments car il peut être efficace pour les tâches de classification, KNN peut également prendre en compte plusieurs dimensions ou caractéristiques des données, ce qui peut être utile pour traiter les données complexes telles que le texte.

3.2.1 Application de KNN

Après avoir entraîné le modèle KNN sur les données d'entraînement en utilisant le **Grid Search**, qui calcule la meilleure configuration du modèle qui correspond au meilleur score.

La figure suivante représente les calculs de GridSearch, les score en fonction de l'hyperparamètre **K** qui correspond au nombre de voisins :

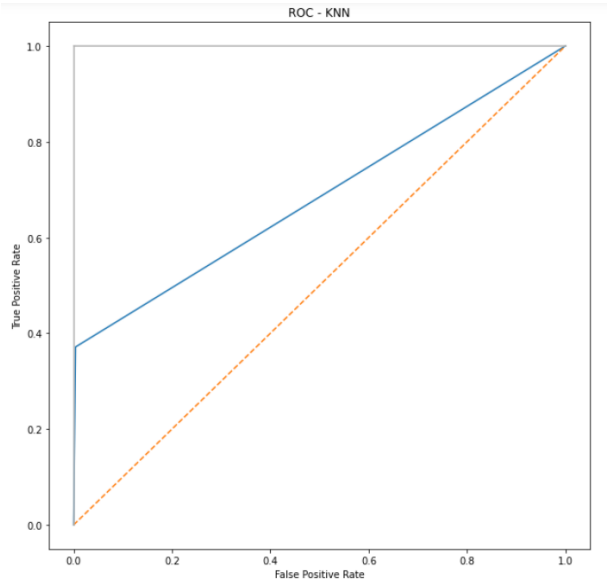


On remarque que le GreadSearch nous a donné la meilleure configuration de knn qui est la suivante : K = 1 avec un score de prédiction de 95%

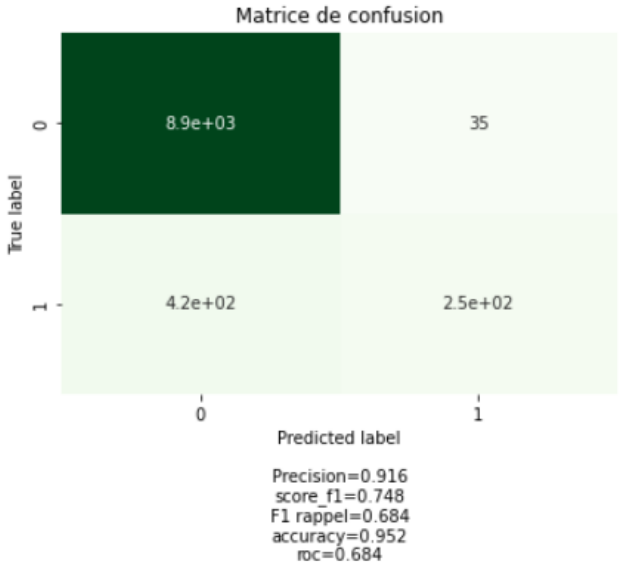
3.2.2 Évaluation de KNN

Concernant l'évaluation du modèle, On a utilisé 5 métriques (precision, score f1, rappel, accuracy, roc), une matrice de confusion et la courbe ROC.

La courbe ROC nous montre la relation entre le taux de vrais positifs et le taux de faux positifs en mesurant la performance du modèle en regardant l'air en dessous de la courbe. Les résultats montrent que notre modèle a une bonne capacité à distinguer les classes positives des négatives.



Aprés avoir evaluer la perfomance et la qualité de l'algorithme KNN on peut remarquer qu'il renvoie :
- 91% de Precision
- 74% de Score F1
- 68% de Recall
- 95% d'Accuracy
- 68% de Roc



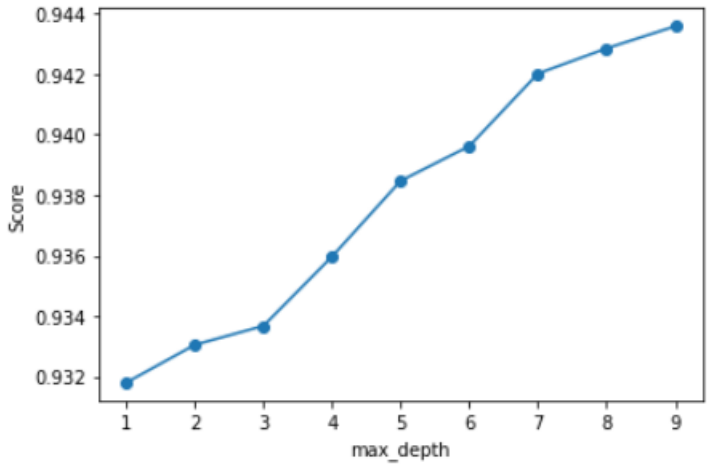
3.3 DecisionTree

L'algorithme DecisionTree est utile pour l'analyse des sentiments en raison de sa capacité à gérer les données de différents types, à prendre en compte toutes les caractéristiques disponibles et à fournir des résultats interprétables et précis.

3.3.1 Application de DecisionTree

Après avoir entraîné le modèle KNN sur les données d'entraînement en utilisant le Grid Search, qui calcule la meilleure configuration du modèle qui correspond au meilleur score.

La figure suivante représente les calculs de GridSearch, les score en fonction de l'hyperparamètre Max_Depth qui correspond à la profondeur maximale de l'arbre :

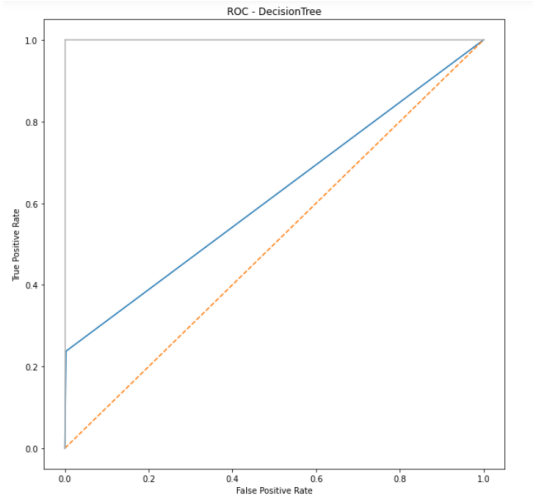


On remarque que le GreadSearch nous a donné la meilleure configuration de DecisionTree qui est la suivante : Max_Depth = 9 avec un score de prédiction de 94%.

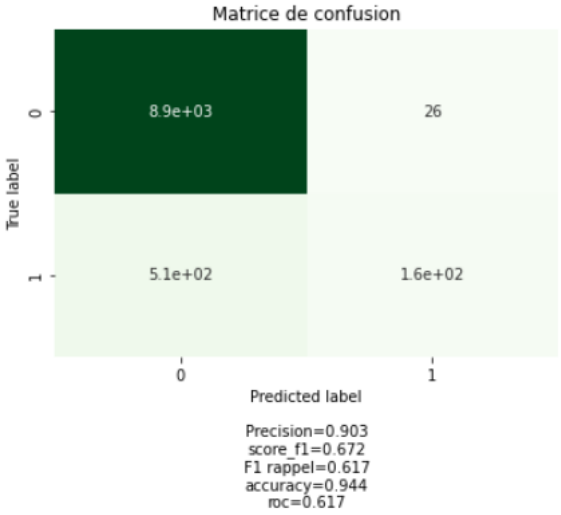
3.3.2 Évaluation de DecisionTree

Concernant l'évaluation du modèle, On a utilisé 5 métriques (precision, score f1, rappel, accuracy, roc), une matrice de confusion et la courbe ROC.

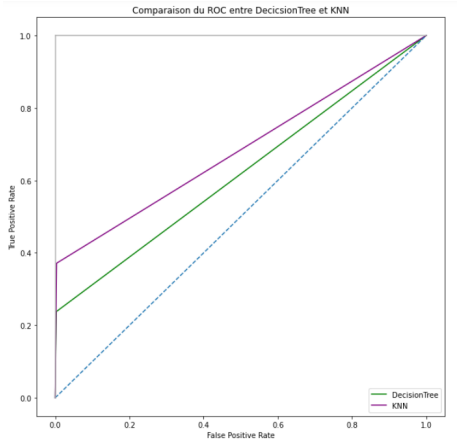
La courbe ROC nous montre la relation entre le taux de vrais positifs et le taux de faux positifs en mesurant la performance du modèle en regardant l'air en dessous de la courbe. Les résultats montrent que notre modèle a une bonne capacité à distinguer les classes positives des négatives.

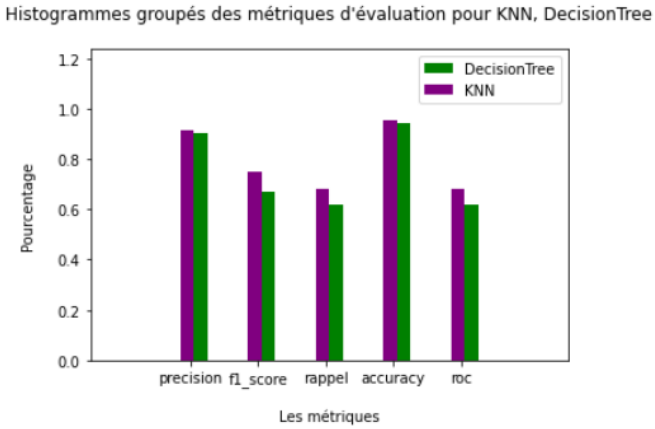


Aprés avoir evaluer la perfomance et la qualité de l'algorithme DecisionTree on peut remarquer qu'il renvoie :
- 90% de Precision
- 67% de Score F1
- 61% de Recall
- 94% d'Accuracy
- 61% de Roc



3.4 Comparaison entre les deux modèles





En visualisant les deux courbe ROC de la figure à gauche, on remarque que KNN est un peu performant que DecisionTree.

En visualisant l'histogramme des métriques étudiées de la figure à droite, on remarque que KNN est peu plus performant que DecisionTree sur toutes les métriques.

4 Interprétation des résultats

on remarque que les deux modèles sont performants, mais y a une petite différence, on voit que KNN et un peu plus performant que decisionTree sur toutes les métriques, on constate que cette différence est dù aux raisons suivantes :

- KNN est plus performant pour la prédiction dans un contexte de text mining avec TF-IDF, en mesurant la similarité entre les instances de données basée sur leurs vecteurs TFIDF.
- KNN est souvent utilisé pour traiter les données non structurées telles que les données textuelles (TWEETS), contrairement à DecisionTree qui est bien adaptés pour les jeux de données Catégorielles, car ils permettent de créer des séparations de données en fonction de la valeur des variables d'entrée.

5 Conclusion

Durant ce projet, nous avons pu encore une fois nous rendre compte de la puissance du NLP qui a permis de classifier de manières fiable le sentiment des tweets. Cette analyse de sentiments Twitter pourrait avoir plusieurs domaines d'application comme par exemple le marketing, de relations publiques, de politique, etc..

A travers ce projet, nous avons couvert les points suivants :

- Comment effectuer une analyse des sentiments Twitter à l'aide de Python et de ses bibliothèques.
- Visualisation des résultats de notre projet d'analyse des sentiments.
- Utilisation de l'apprentissage automatique avec les algorithme Machine Learning afin de classifier les sentiment, puis l'évaluation des modèles utilisés.

La dernière étape est la partie la plus intéressante au niveau de l'analyser des sentiments. Pour cette partie nous avons présenté deux algorithmes différents pour effectuer la classification des sentiments.

La réalisation de ce projet nous a permis de bien comprendre le processus que nous a mis en place **Text Mining**. Ainsi, il nous a aidé à s'entraîner à travailler avec des données et à effectuer des opérations NLP.

Pour conclure, nous avons présenté les principaux travaux portant sur l'analyse des sentiments qui est devenue un domaine de recherche très populaire. De nombreuses recherches ont été menées dans ce domaine, mais il existe encore de nombreux problèmes, car l'analyse des sentiments traite des données non structurées basées sur du texte. Cependant, il est important de prendre en compte les limitations de l'analyse des sentiments, telles que la subjectivité de la polarité et la complexité de la compréhension du langage humain.

6 Liens/Référence

- *https ://scikit-learn.org/*
- *https ://spacy.io/api/doc*
- *https ://datascientest.com/introduction-au-nlp-natural-language-processing*
- *https ://medium.com/search ?q=Sentiments+Analysis*
- *https ://programminghistorian.org/fr/lecons/analyse-de-documents-avec-tfidf*
- *https ://towardsdatascience.com/build-a-spelling-corrector-program-in-python-46bc427cf57f*
- *https ://realpython.com/nltk-nlp-python/*
- *https ://www.crummy.com/software/BeautifulSoup/bs4/doc/*