

Rapport projet : Solution Factory

OPINIFY



Professeur : MOUMOUH, Houssam

Equipe : 221

Module : FT601

Sommaire

Introduction	3
I. Réalisation du projet.....	4
1. Exigence et contraintes.....	4
2. Présentation des data set	5
3. Les librairies utilisées	6
4. Préparation des données	7
5. Analyse exploratoire des données	9
6. Modèle d'apprentissage automatique	15
7. Evaluation des modèles	18
8. Test Spotify.....	21
9. Interface utilisateur.....	27
II. Impact économique sur nos potentielles clients	38
1. La concurrence.....	38
2. Nos objectifs futurs.....	38
Conclusion	39

Introduction

Chaque jour, les entreprises font face à un défi constant : améliorer leurs produits et services pour répondre aux attentes toujours plus élevées de leurs clients. Dans ce contexte, l'analyse des avis clients joue un rôle crucial, en leur fournissant des informations précieuses sur la satisfaction des utilisateurs et les domaines à améliorer. Cependant, cette tâche peut s'avérer fastidieuse et chronophage lorsqu'il s'agit d'analyser un grand nombre d'avis. C'est là que notre projet, OPINIFY, entre en jeu.

OPINIFY est un projet ambitieux d'une durée d'un mois qui vise à aider les entreprises à améliorer leurs produits et services grâce à une analyse semi-automatisée des avis clients. Notre équipe, composée de quatre personnes passionnées et expérimentées, s'est réunie dans le but de simplifier ce processus et de fournir aux entreprises des informations claires et pertinentes sur le ressenti de leurs utilisateurs.

L'objectif de notre projet découle directement de la difficulté rencontrée par les entreprises pour extraire des informations significatives à partir des avis clients. L'analyse manuelle de chaque commentaire est une tâche fastidieuse qui limite leur capacité à réagir rapidement et efficacement aux retours des utilisateurs. Nous y avons donc identifié une opportunité de créer une solution semi-automatisée qui permettrait aux entreprises de relever ces informations de manière efficace.

La problématique centrale de notre projet est donc la suivante : comment automatiser l'analyse des avis clients pour permettre aux entreprises d'obtenir rapidement des informations précieuses sur le ressenti des utilisateurs ? En répondant à cette question, nous viserons par la suite à fournir aux entreprises une meilleure compréhension des réactions des consommateurs, ce qui leur permettra de prendre des décisions éclairées et d'apporter des améliorations significatives à leurs produits et services.

Pour répondre à cette problématique, verrons des différents point clés de la réalisation puis l'impact sociétal et économique chez nos potentiels client.

I. Réalisation du projet

Pour réaliser ce projet, nous avons utilisé 2 data set : un provenant d'Amazon, et l'autre de Spotify. Le data set d'Amazon a été le data set utilisé pour entraîner nos modèles, tandis que le data set Spotify a été le data set utilisé pour tester notre projet.

1. Exigence et contraintes

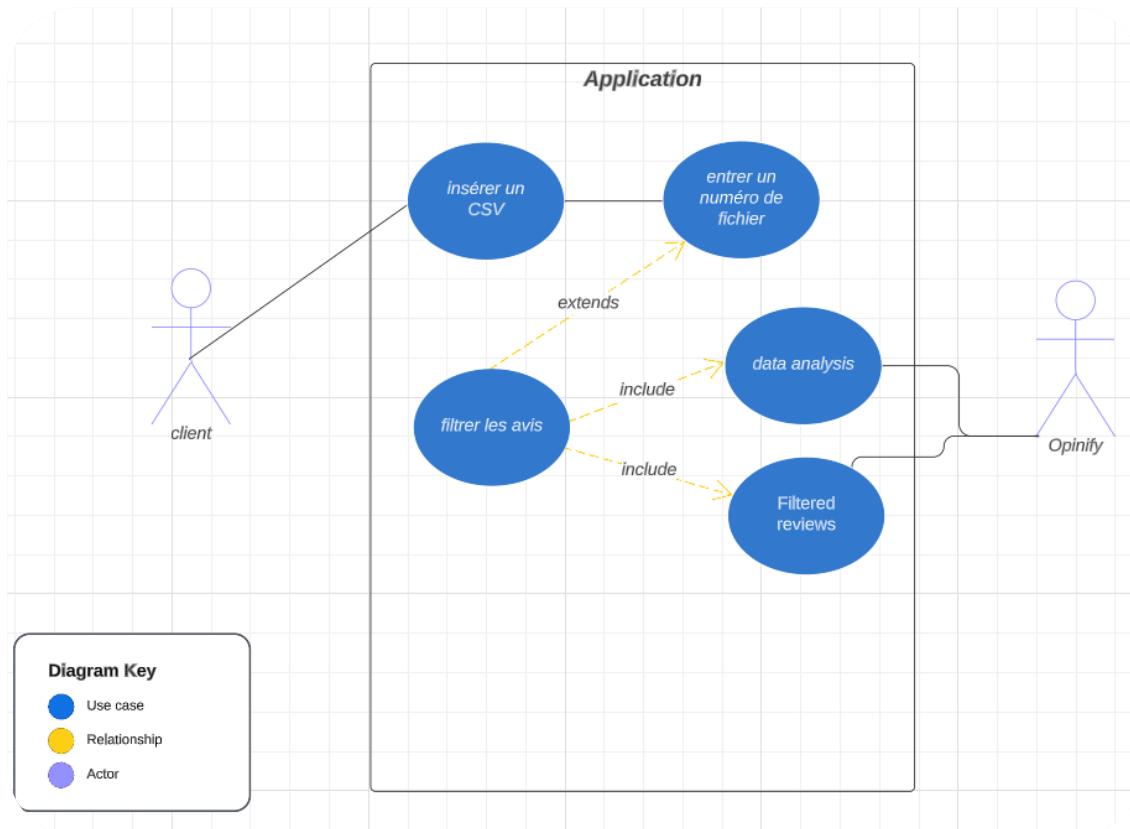
Les exigences sont les suivantes :

- Créer un service répondant au besoin du marché actuel
- Créer un site attractif et intuitif
- Créer un service rapide, optimisé et performant
- Crée un produit peu onéreux, le budget pour ce projet étant limité
- Pouvoir traiter les avis de différentes bases de données
- Créer un service peu couteux

Les contraintes sont les suivantes :

- Utilisation de nouvelles librairies python
- Relier notre code python au site web (interface utilisateur)
- Comprendre de nouveaux modèles de Machine Learning
- Faire un site responsive et agréable à utiliser pour les utilisateurs.

Pour mieux comprendre le fonctionnement de notre service, voici son Diagramme séquentiel :



2. Présentation des data set

Pour réaliser ce projet, nous avons récolté les 2 data set sur Kaggle, une plateforme en ligne populaire hébergeant de nombreuses bases de données en libre accès. C'est sur ce site que nous avons pu trouver notre data set d'entraînement « AmazonReview » et notre data set de test « SpotifyReview »

AmazonReview :

Index	sentiment	review
0	2	Despite the fact that I have only played a ...
1	1	I bought this charger in Jul 2003 and it wo...
2	2	Check out Maha Energy's website. Their Powe...
3	2	Reviewed quite a bit of the combo players a...
4	1	I also began having the incorrect disc prob...

Composé de 400 000 individus, ce data set comporte 2 colonnes :

- **Sentiment** : Répertorie des sentiments annotés numériquement. Le 1 correspond au sentiment négatif et le 2 au sentiment positif.
- **review** : Comporte les différents commentaires.

SpotifyReview :

Index	Time_submitted	Review	Rating	Total_thumbsup	Reply
0	2022-07-09 15:00:00	Great music service, the audio is high qual...	5	2	nan
1	2022-07-09 14:21:22	Please ignore previous negative rating. This app is super great. I give it five stars+	5	1	nan
2	2022-07-09 13:27:32	This pop-up "Get the best Spotify experienc...	4	0	nan
3	2022-07-09 13:26:45	Really buggy and terrible to use as of recently	1	1	nan
4	2022-07-09 13:20:49	Dear Spotify why do I get songs that I didn...	1	1	nan

Composé de 62 000 lignes, ce data set est composé de 5 colonnes :

- **Time_submitted** : Permet de voir à quelles date et heures le review et rating ont été saisie.
- **Review** : Constitué des différents commentaires.
- **Rating** : Comporte les différentes notes allant de 1 à 5.
- **Total_thumbsup** : Le nombre de like du commentaire.
- **Reply** : S'il y a eu une réponse au commentaire.

3. Les librairies utilisées

Afin de pouvoir commencer à réaliser les différentes étapes de programmation, nous nous sommes renseignés sur les différentes librairies que nous allions devoir utiliser :

```
import pandas as pd
import re
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import time
```

Pandas : Très populaire pour la manipulation et l'analyse de données. Elle offre des structures de données efficaces (notamment les DataFrames) pour organiser et manipuler des ensembles de données tabulaires. Pandas permet de charger des données à partir de diverses sources, de nettoyer les données, de les transformer, de les agréger et de les analyser de manière pratique et flexible.

re : "re" (raccourci pour Regular Expression) fournit des fonctionnalités pour travailler avec des expressions régulières. Les expressions régulières sont des motifs de recherche qui permettent de rechercher et de manipuler du texte de manière avancée.

nltk : Natural Language Toolkit (NLTK) est utilisée pour le traitement du langage naturel. Elle fournit des outils et des ressources pour effectuer des tâches de traitement du langage naturel, telles que la tokenization, le stemming, la lemmatisation, la classification de texte, l'analyse syntaxique, la génération de texte, etc. NLTK offre également une grande variété de corpus de texte et de modèles pré-entraînés pour faciliter le développement.

sklearn : Scikit-learn / sklearn est une bibliothèque d'apprentissage automatique (machine learning). Elle fournit des outils pour réaliser une grande variété de tâches d'apprentissage automatique, y compris la classification, la régression, le clustering, la réduction de dimensionnalité, la sélection de modèles et bien d'autres. Scikit-learn propose une implémentation cohérente et conviviale d'algorithme d'apprentissage automatique et offre des fonctionnalités pour l'évaluation des modèles, la préparation des données et le traitement des flux de travail de l'apprentissage automatique.

matplotlib : Matplotlib est une bibliothèque qui permet de créer des visualisations et des tracés de données. Elle offre une grande flexibilité pour créer des graphiques en 2D, des histogrammes, des diagrammes en boîte, des graphiques en barres, des diagrammes de dispersion et bien plus encore. Matplotlib permet également de personnaliser les graphiques avec des étiquettes, des titres, des légendes, des couleurs et des styles.

seaborn : seaborn permet de faire de la visualisation de données basée sur matplotlib. Elle offre une interface haut niveau pour créer des graphiques statistiques attrayants et informatifs. seaborn simplifie la création de graphiques tels que les diagrammes de dispersion avec ajustement de régression, les matrices de corrélation, les diagrammes en violon, les diagrammes en boîte, les diagrammes de densité, etc.

numpy : Bibliothèque fondamentale pour le calcul numérique en Python. Elle fournit des structures de données performantes pour représenter et manipuler des tableaux multidimensionnels ainsi que des fonctions mathématiques et d'algèbre linéaire pour effectuer des opérations sur ces tableaux.

time : Fournit des fonctionnalités pour travailler avec le temps.

4. Préparation des données

Dans un premier temps, nous avons chargé notre data set d'entraînement : « AmazonReview » :

Input :

```
#CHARGEMENT DU DATASET
df_OG = pd.read_csv('AmazonReview.csv') #garder une version original
df = pd.read_csv('AmazonReview.csv')
```

Ce code lit le fichier csv et crée deux dataframes, "df_OG" pour conserver la version originale des données et "df" pour une utilisation ultérieure. Cela nous permet d'avoir une visibilité sur notre data set et de prendre connaissance de celui-ci.

Input :

```
print("-----VISUALISATION DU DATASET-----")
df.info() #Affiche les informations des variables (colonne) : le type, et les valeurs non nulles
print(df.describe()) #Affiche les statistiques descriptives pour le dataset
```

On utilise `df.info()` pour afficher les informations des variables et `df.describe` pour afficher les statistiques descriptives du dataset.

Output :

```
-----VISUALISATION DU DATASET-----
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 399999 entries, 0 to 399998
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   sentiment    399999 non-null   int64  
 1   review       399999 non-null   object 
dtypes: int64(1), object(1)
memory usage: 6.1+ MB
sentiment
count    399999.000000
mean     1.499999
std      0.500001
min     1.000000
25%    1.000000
50%    1.000000
75%    2.000000
max     2.000000
```

Ici, nous pouvons voir les différentes informations de df en **rouge** et les différentes valeurs statistiques en **orange**.

Input :

```
print("-----PREPARATION DU DATASET-----")
#pour une meilleur lecture du dataset
df['sentiment'] = df['sentiment'].replace({1: 'negatif', 2: 'positif'})
```

La ligne de code remplace les valeurs dans la colonne "sentiment" du dataframe. Les valeurs 1 sont remplacées par 'négatif' et les valeurs 2 sont remplacées par 'positif'. Cette opération vise à attribuer des étiquettes plus explicites aux valeurs existantes dans la colonne "sentiment".

Input :

```
print("-----compter les commentaires positifs et négatifs-----")
positif_count = df[df['sentiment'] == 'positif'].shape[0]
negatif_count = df[df['sentiment'] == 'negatif'].shape[0]
print("Nombre de commentaires positifs :", positif_count)
print("Nombre de commentaires négatifs :", negatif_count)
```

Ce code compte le nombre de commentaires positifs et négatifs dans le dataframe en se basant sur la colonne "sentiment" et affiche ces deux nombres. Cela permet de quantifier la répartition des commentaires positifs et négatifs dans le dataframe.

Output :

```
-----compter les commentaires positifs et négatifs---
Nombre de commentaires positifs : 199999
Nombre de commentaires négatifs : 200000
```

On peut voir ici que le data set comporte 50% de commentaire positifs (2) et 50% de négatif (1)

Input :

```
print("-----Supprime ligne si valeurs manquantes-----")
df = df.dropna()
```

Permet de supprimer les individus contenant des valeurs vides

Input :

```
print("-----Prétraitement des commentaires-----")
def preprocess_text(text):
    text = re.sub(r'[^w\s]', '', text.lower()) #supprime tous les caractères non alphanumériques et non espaces + texte en minuscule
    text = ' '.join(word_tokenize(text)) #mise au propre du texte
    stop_words = set(stopwords.words('english')) #chargement des mots courants non utile
    text = ' '.join([word for word in word_tokenize(text) if word not in stop_words]) #suppressions des mots non utile
    return text
df['review'] = df['review'].apply(preprocess_text)
```

Ce code effectue le prétraitement des commentaires en supprimant les caractères non alphanumériques et non espaces, en mettant le texte en minuscules, en supprimant les mots non utiles (stopwords) et en stockant les commentaires prétraités dans la colonne "review" du dataframe.

Input :

```
print("-----suppression des commentaires présents plusieurs fois-----")
doublons = df.duplicated(subset=['review']) #identifie les lignes en double
print("présence de doublons avant traitement : ",doublons.any())
#print(doublons)
df_unique = df.drop_duplicates(subset=['review']) #supprime les lignes en double
doublons_unique = df_unique.duplicated(subset=['review']) #identifie les lignes en double
print("présence de doublons après traitement : ",doublons.unique.any())
```

Ce code identifie et supprime les commentaires en double dans le dataframe en se basant sur la colonne "review". Il affiche également des messages indiquant la présence de doublons avant et après le traitement.

Output :

```
-----suppression des commentaires présents plusieurs fois-----
présence de doublons avant traitement :  True
présence de doublons après traitement : False
```

On peut voir que les doublons ont bien été supprimés après le traitement.

Input :

```
print("-----création du csv final-----")
df_unique.to_csv('Amazon_final.csv', index=False)
```

Après ces différentes étapes de traitement, nous obtenons un data set final, traité que nous allons utiliser pour les prochaines étapes.

Index	sentiment	review
0	positif	despite fact played small portion game musi...
1	negatif	bought charger jul 2003 worked ok design ni...
2	positif	check maha energys website powerex mhc204f ...
3	positif	reviewed quite bit combo players hesitant d...
4	negatif	also began incorrect disc problems ive read...

5. Analyse exploratoire des données

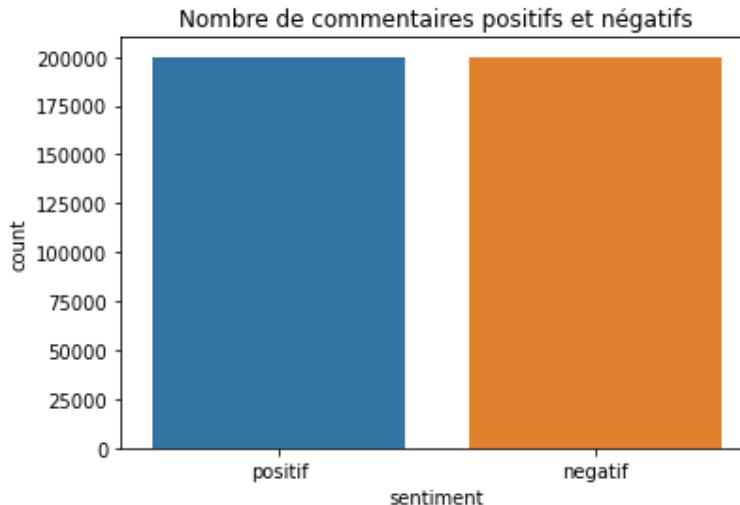
Après avoir réalisé l'étape de prétraitement des données, nous allons utiliser notre nouveau data set traité afin de mener une analyse exploratoire sur nos données.

Input :

```
##% countplot des positifs et négatifs
sns.countplot(x='sentiment', data=df)
plt.title('Nombre de commentaires positifs et négatifs')
plt.show()
```

Ce code crée un graphique countplot qui montre le nombre de commentaires positifs et négatifs dans le dataframe df. Le graphique permet de visualiser la répartition des commentaires en fonction de leur sentiment.

Output :



Nous pouvons ici voir la distribution de notre variable sentiments positif et négatif qui est de 50% pour les positifs et négatifs.

Input :

```
% commentaires positifs et négatifs
df_com['longueur_commentaires'] = df_com['review'].apply(len)

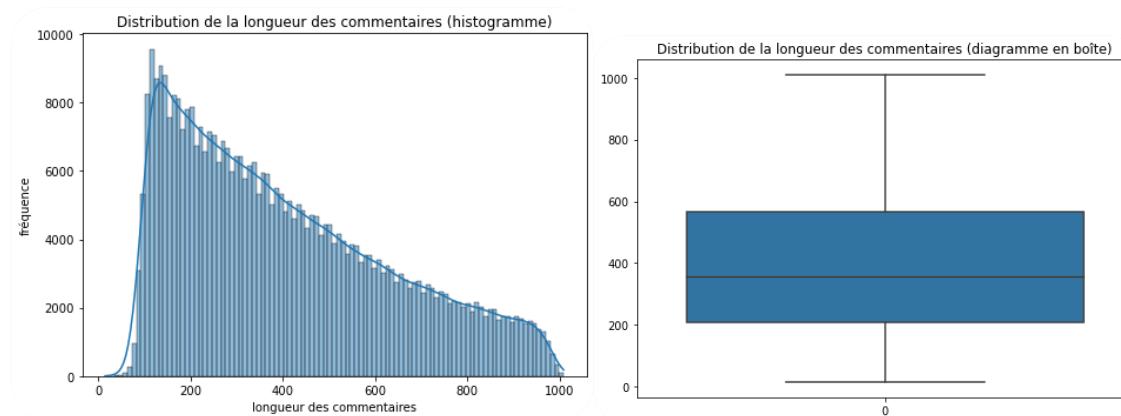
#histogramme
plt.figure(figsize=(7,5))
sns.histplot(df_com['longueur_commentaires'],kde=True)
plt.xlabel("longueur des commentaires") #Titre de l'axe x
plt.ylabel('fréquence') #Titre de l'axe y
plt.title('Distribution de la longueur des commentaires (histogramme)') #Titre de chaque histogramme
plt.tight_layout() #Ajuste le padding des figures
plt.show() #Affichage

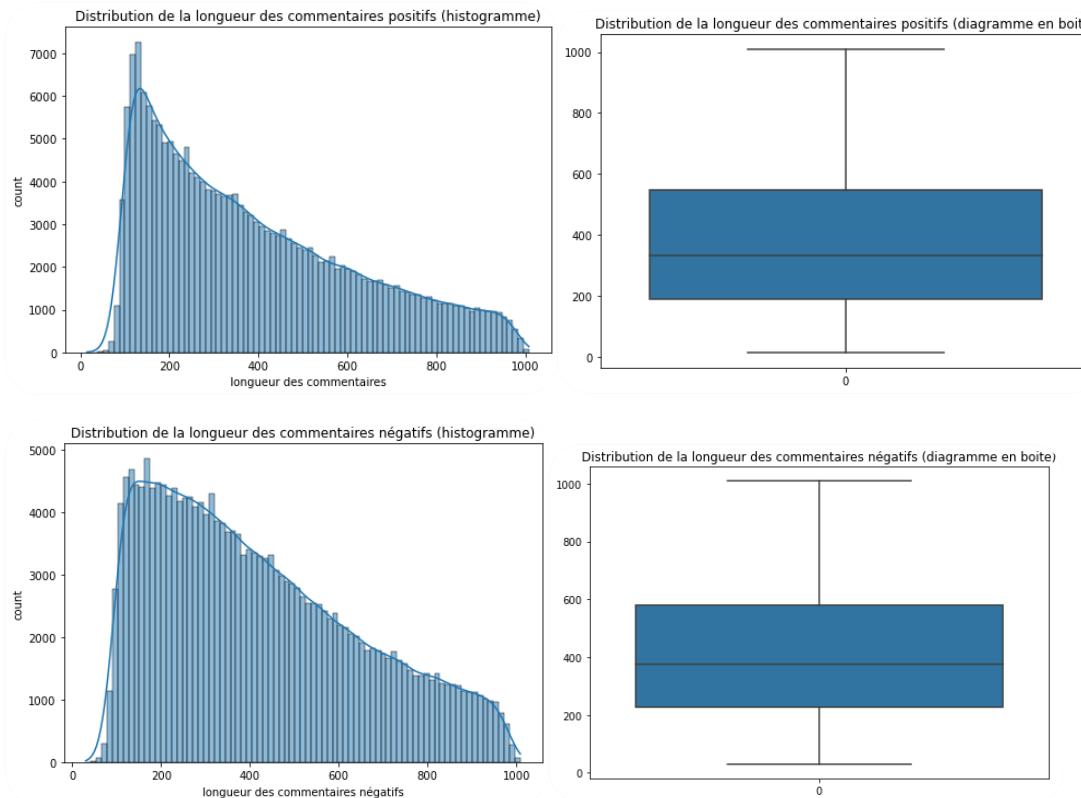
#diagramme en boîte
plt.figure(figsize=(7,5)) # Creation de la figure avec configuration de la largeur et la hauteur
sns.boxplot(df_com['longueur_commentaires']) #Creation du diagramme en boîte de la variable en question
plt.title('Distribution de la longueur des commentaires (diagramme en boîte)') #Titre de chaque diagramme en boîte
plt.tight_layout() #Ajuste le padding des figures
plt.show() #Affichage
```

En résumé, ce code analyse la longueur des commentaires dans le dataframe df_com en créant un histogramme et un diagramme en boîte pour visualiser la distribution de cette longueur.

L'histogramme montre la fréquence de chaque longueur de commentaire, tandis que le diagramme en boîte présente la distribution sous forme de boîtes, avec des indications sur les quartiles et les valeurs aberrantes éventuelles.

Output :





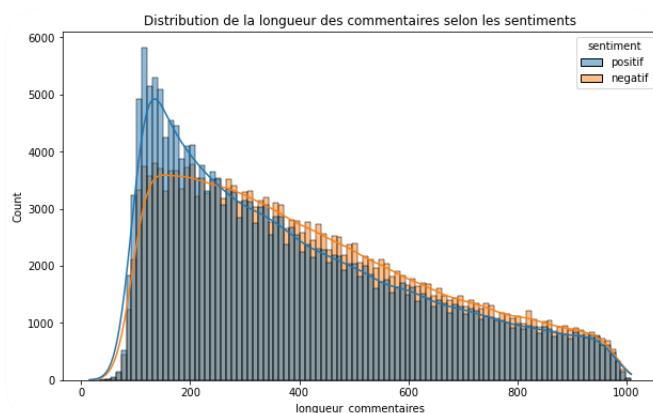
Grace aux histogrammes et boxplot, nous arrivons à bien comprendre la distribution de nos variables et l'évolution de celle-ci dans notre data frame.

Input :

```
#%% commentaires positifs et négatifs
#histogramme avec couleur selon positif ou negatif
plt.figure(figsize=(10, 6))
sns.histplot(data=df_com, x='longueur_commentaires', hue='sentiment', kde=True)
plt.title('Distribution de la longueur des commentaires selon les sentiments')
plt.show()
```

Ce code crée un histogramme avec des couleurs différencierées pour visualiser la distribution de la longueur des commentaires selon leur sentiment dans le dataframe df_com. Cela permet d'observer la répartition des longueurs de commentaires positifs et négatifs de manière visuelle et comparative.

Output :



On peut voir ici que la majorité des commentaires négatifs se trouve sur le commentaire les moins composé.

Input :

```
#%% commentaires positifs
df_positif = df_com.loc[df_com['sentiment'] == 'positif'].reset_index(drop=True)

#histogramme
plt.figure(figsize=(7,5))
sns.histplot(df_positif['longueur_commentaires'],kde=True)
plt.xlabel("longueur des commentaires") #Titre de l'axe x
plt.ylabel('count') #Titre de l'axe y
plt.title('Distribution de la longueur des commentaires positifs (histogramme)') #Titre de chaque histogramme
plt.tight_layout() #Ajuste le padding des figures
plt.show() #Affichage

#diagramme en boite
plt.figure(figsize=(7,5)) # Creation de la figure avec configuration de la largeur et la hauteur
sns.boxplot(df_positif['longueur_commentaires']) #Creation du diagramme en boite de la variable en question
plt.title('Distribution de la longueur des commentaires positifs (diagramme en boite)') #Titre de chaque diagramme en boite
plt.tight_layout() #Ajuste le padding des figures
plt.show() #Affichage
```

Ce code effectue une analyse spécifique des commentaires positifs dans le dataframe df_com en créant un histogramme et un diagramme en boîte pour visualiser la distribution de la longueur de ces commentaires positifs. Les graphiques permettent de mieux comprendre la distribution de la longueur des commentaires positifs et d'identifier d'éventuelles tendances ou valeurs aberrantes spécifiques à cette catégorie de commentaires.

Output :



Cela permet d'avoir un aspect visuel des différents mots positifs les plus présent.

Input :

```
%# commentaires negatifs
df_negatif = df_com.loc[df_com['sentiment'] == 'negatif'].reset_index(drop=True)

#histogramme
plt.figure(figsize=(7,5))
sns.histplot(df_negatif['longueur_commentaires'],kde=True)
plt.xlabel("longueur des commentaires négatifs") #Titre de l'axe x
plt.ylabel('count') #Titre de l'axe y
plt.title('Distribution de la longueur des commentaires négatifs (histogramme)') #Titre de chaque histogramme
plt.tight_layout() #Ajuste le padding des figures
plt.show() #Affichage

#diagramme en boite
plt.figure(figsize=(7,5)) # Creation de la figure avec configuration de la largeur et la hauteur
sns.boxplot(df_negatif['longueur_commentaires']) #Creation du diagramme en boite de la variable en question
plt.title('Distribution de la longueur des commentaires négatifs (diagramme en boite)') #Titre de chaque diagramme en boite
plt.tight_layout() #Ajuste le padding des figures
plt.show() #Affichage
```

Ce code effectue une analyse spécifique des commentaires négatifs dans le dataframe df_com en créant un histogramme et un diagramme en boîte pour visualiser la distribution de la longueur de ces commentaires négatifs. Les graphiques permettent de mieux comprendre la distribution de la

longueur des commentaires négatifs et d'identifier d'éventuelles tendances ou valeurs aberrantes spécifiques à cette catégorie de commentaires.

Output :



Cela permet d'avoir un aperçu des différents commentaires négatifs.

Input :

```

#% commentaires : mots fréquents (comptage)
positive_comments = df[df['sentiment'] == 'positif']['review']
negative_comments = df[df['sentiment'] == 'négatif']['review']

#concaténation + minuscule + division en mot individuel + comptage de la fréquence de chaque mot + affichage des 10 mots les plus fréquents
positive_word_counts = pd.Series(' '.join(positive_comments).lower().split()).value_counts()[:10]
negative_word_counts = pd.Series(' '.join(negative_comments).lower().split()).value_counts()[:10]

print("Mots fréquents dans les commentaires positifs :")
print(positive_word_counts)

print("\nMots fréquents dans les commentaires négatifs :")
print(negative_word_counts)

```

Ce code permet de compter les mots les plus fréquents dans les commentaires positifs et négatifs du dataframe df, et d'afficher les résultats. Cela permet d'identifier les mots qui apparaissent le plus souvent dans chaque catégorie de commentaires et d'obtenir un aperçu des termes fréquemment utilisés dans les commentaires positifs et négatifs.

Output :

```
Mots fréquents dans les commentaires positifs :  
book      98846  
one       67445  
great     64809  
like       50985  
good      50236  
read       38671  
would     34794  
love       33479  
well       33241  
get        30955  
dtype: int64
```

Cela nous permet de faire une comparaison avec le wordcloud des mots les plus fréquent dans les commentaires positifs.

```
Mots fréquents dans les commentaires négatifs :  
book      96003  
one       71429  
like      57044  
would    56148  
get       41207  
good      40342  
dont      39221  
time      35725  
even      33780  
movie     32460
```

Cela nous permet de faire une comparaison entre le wordcloud des mots qui ressorte le plus dans les commentaires négatifs.

Input :

```
% commentaires : mots fréquents (nuage de mots)

# Concaténation des commentaires positifs
pos_comments = df[df['sentiment'] == 'positif']['review']
positive_text = ' '.join(pos_comments)

# Création du nuage de mots pour les commentaires positifs
wordcloud_positive = WordCloud(background_color='white').generate(positive_text)

# Affichage du nuage de mots pour les commentaires positifs
plt.figure(figsize=(8, 8))
plt.imshow(wordcloud_positive, interpolation='bilinear')
plt.axis('off')
plt.title('Nuage de mots - Commentaires positifs')
plt.show()

# Concaténation des commentaires négatifs
neg_comments = df[df['sentiment'] == 'négatif']['review']
negative_text = ' '.join(neg_comments)

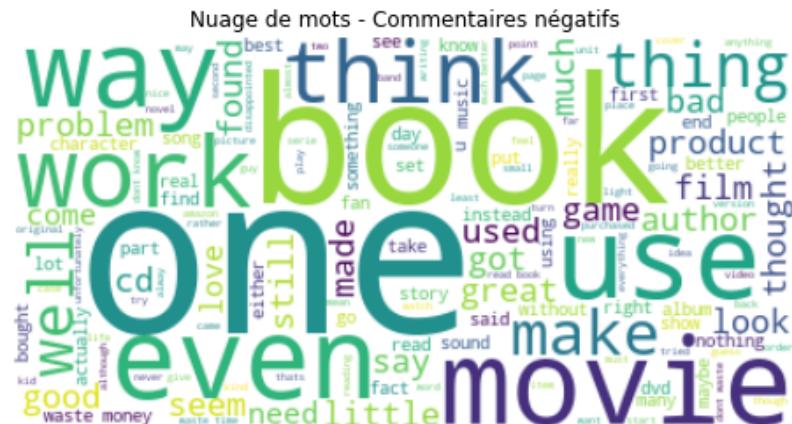
# Création du nuage de mots pour les commentaires négatifs
wordcloud_negative = WordCloud(background_color='white').generate(negative_text)

# Affichage du nuage de mots pour les commentaires négatifs
plt.figure(figsize=(8, 8))
plt.imshow(wordcloud_negative, interpolation='bilinear')
plt.axis('off')
plt.title('Nuage de mots - Commentaires négatifs')
plt.show()
```

Ce code crée des nuages de mots à partir des commentaires positifs et négatifs du dataframe df. Les nuages de mots permettent de visualiser les mots les plus fréquemment utilisés dans chaque catégorie de commentaires, avec une taille de police plus grande pour les mots les plus fréquents. Cela peut fournir des informations visuelles sur les termes dominants dans les commentaires positifs et négatifs, aidant ainsi à l'analyse et à la compréhension du contenu des commentaires.

Output :





6. Modèle d'apprentissage automatique

Après avoir identifié notre variable Target et Features, nous nous sommes intéressés au modèle de machine learning qui pourrait convenir le plus à notre projet : Transformer, RNN, Bag of Word et Logistic Regression.

Transformer :

En effet dans un premier temps, nous nous sommes penchés vers l'utilisation du modèle d'apprentissage Transformer. Ce modèle permet de capturer les dépendances contextuelles entre les éléments d'une séquence. Ayant révolutionné le traitement du langage naturel en fournissant des performances de pointe dans des tâches comme la traduction automatique ou la génération de texte, nous pensions que ce modèle était idéal à notre projet.

Cependant, après avoir pris connaissance de ce modèle et de l'avoir inséré dans notre code, nous nous sommes rendus compte que notre machine manquait de ressource pour le faire tourner. Étant trop gourmand et demandant beaucoup trop de temps de compilation et d'entraînement, nous avons préféré basculer vers un autre modèle.

RNN (Récurrent Neural Network) :

Après l'étude du modèle des transformateurs, nous avons basculé vers le modèle RNN qui est un modèle similaire. En effet, étant basé sur une architecture de réseau de neurones récurrents, ce modèle permet de traiter des données séquentielles en prenant en compte les dépendances temporelles. Étant très réputé, il est utilisé dans diverses tâches de traitement de langage naturel et de reconnaissance de formes et de séries temporelles.

Cependant tout comme le modèle des transformateurs, ce modèle demande trop de ressources et prend énormément de temps à s'entraîner malgré les différents paramètres que l'on peut ajuster :

Entrainement du modèle beaucoup trop long :

```

  ✓ Rétraitement des avis
  Préparer les données d'entraînement et de test
  Créer une classe personnalisée pour le dataset
  Créer une fonction pour préparer les données
  Tokenization et création du vocabulaire
  Paramètres du modèle
  Création du modèle RNN
  Création du modèle
  Entraînement du modèle
Epoch 1/5: 100%|██████████| 5000/5000 [19:40<00:00,  4.23it/s]
Epoch 1/5, Loss: 2638.4248965382576
Epoch 2/5: 100%|██████████| 5000/5000 [18:39<00:00,  4.46it/s]
Epoch 2/5, Loss: 1404.5716015323997
Epoch 3/5: 25%|██| 1254/5000 [04:40<17:55,  3.48it/s]

```

Paramètres du modèle :

```
print("Paramètres du modèle")
embedding_size = 100 #Représentation des mots sous vecteur de taille
hidden_size = 128 #Taille d'état caché (combien de neurones sont utilisés)
num_classes = 2 #Nombre de classe de sortie (donnée classée en 2 classes)
num_layers = 1 #Nombre de couches de neurones récurrents empilées
batch_size = 64 #Nombre d'échantillon utilisé par itération dans l'entraînement
num_epochs = 5 #Nombre total d'étape d'entraînement
```

Nous avons essayé de modifier le nombre d'étape entraînement en établissant num_epochs à 2, 5 et 20. Pour 2 étapes, nous avons remarqué que le modèle prend moins de temps à s'entraîner mais engendré des problèmes d'underfitting. Pour 5 étapes d'entraînement, notre modèle fonctionne mais prend plus de 5h pour pouvoir compiler entièrement (1h par étape). Enfin pour 20 étapes, l'entraînement prend énormément de temps et engendre des problèmes d'overfitting.

Par conséquent même si notre accuracy était de 78% avec un nombre total d'entraînement de 5 étapes, nous avons décidé de ne pas sélectionner ce modèle.

Bag of Word :

Ce troisième modèle que nous avons étudié a été beaucoup plus valorisant de nos attentes. En effet, les sacs de mots permettant de construire un vocabulaire à partir de l'ensemble des mots uniques présents dans le corpus de texte. Chaque mot unique est ensuite associé à un indice numérique permettant de créer un vecteur pour chaque élément du vecteur correspond au nombre d'apparitions/fréquence d'apparition d'un mot spécifique. En développant ce modèle dans notre code, nous avons obtenu une accuracy de 75% avec un temps d'exécution relativement faible.

Performance du modèle:
Exactitude (Accuracy): 0.751
Précision (Precision): 0.7510318601153422

Cependant, sachant que nous pouvions obtenir une meilleure valeur d'accuracy avec un autre modèle, nous avons décidé de ne pas sélectionner celui-ci.

Logistic Regression :

Le modèle d'apprentissage de régression logistique est une technique de classification binaire qui utilise une fonction sigmoïde pour prédire la probabilité d'appartenance à la classe positive. Étant simple, interprétable et rapide d'exécution, ce modèle a pu répondre à nos attentes en nous fournissant une accuracy de 87%.

Afin de réaliser ce modèle, nous avons effectué les étapes suivantes :

Input :

```
# Diviser les données en ensembles d'entraînement et de test
X = data['review']
y = data['sentiment']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Ce code effectue une division des données en ensembles d'entraînement et de test. Il extrait les colonnes "review" et "sentiment" du dataframe "data" et les assigne respectivement aux variables "X" et "y". Ensuite, il utilise la fonction **train_test_split** de scikit-learn pour diviser les données en ensembles d'entraînement et de test.

L'ensemble de test est défini à 20% des données totales, ce qui signifie que 80% des données seront utilisées pour l'entraînement et 20% pour les tests. La valeur **random_state=42** est utilisée pour fixer la graine aléatoire, ce qui permet d'obtenir des résultats reproductibles lors de la division des données.

Input :

```
# Créer une instance de TfidfVectorizer et transformer les données textuelles en vecteurs
vectorizer = TfidfVectorizer()
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)
```

Ce code utilise la classe TfidfVectorizer de scikit-learn pour transformer les données textuelles en vecteurs numériques représentant la fréquence inverse de la documentation (TF-IDF).

La première ligne crée une instance de TfidfVectorizer appelée "vectorizer". La deuxième ligne utilise la méthode `fit_transform()` de "vectorizer" pour transformer les données textuelles d'entraînement "X_train" en vecteurs TF-IDF. La troisième ligne utilise la méthode `transform()` de "vectorizer" pour transformer les données textuelles de test "X_test" en vecteurs TF-IDF.

Après l'exécution de ce code, les données textuelles d'entraînement et de test sont converties en vecteurs numériques TF-IDF, prêts à être utilisés dans des modèles d'apprentissage automatique.

Input :

```
# Entrainer un modèle de régression logistique
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)
```

Ce code entraîne un modèle de régression logistique en utilisant la classe LogisticRegression de scikit-learn.

La première ligne crée une instance de LogisticRegression avec une configuration spécifiée pour le nombre maximal d'itérations (`max_iter=200`). Nous avons fixé ce nombre car à 100 itérations, la régression n'avait eu le temps de converger vers une solution optimale.

La deuxième ligne utilise la méthode `fit()` pour entraîner le modèle avec les données d'entraînement (`X_train`) et les étiquettes correspondantes (`y_train`).

Après l'exécution de ce code, le modèle de régression logistique est entraîné et prêt à être utilisé pour effectuer des prédictions sur de nouvelles données.

Input :

```
# Faire des prédictions sur les données de test
y_pred = model.predict(X_test)
end_time = time.time()
execution_time = end_time - start_time
print("Temps d'exécution :", execution_time, "secondes")
```

Ce code utilise le modèle de régression logistique entraîné pour faire des prédictions sur les données de test. Il mesure également le temps d'exécution de cette opération et l'affiche à des fins de suivi.

Output :

```
Temps d'exécution : 136.99000215530396 secondes
```

Nous pouvons voir qu'en plus d'avoir une précision très bonne, notre temps d'exécution sur ce modèle est relativement court avec 137 secondes. Ce qui fait de ce modèle, le plus optimal comparé à l'autre.

7. Evaluation des modèles

Après avoir entraîné notre modèle de régression logistique, nous allons évaluer notre modèle par le biais de différentes étapes :

Input :

```
print("-----EVALUATION DU MODÈLE-----")

# Calculer l'exactitude des prédictions
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Ce code évalue la performance du modèle en calculant son exactitude sur les données de test et affiche ce résultat. L'exactitude est une mesure courante pour évaluer la précision globale d'un modèle de classification.

Output :

```
Accuracy: 0.8690625
```

Input :

```
# Calculer le score F1
f1 = classification_report(y_test, y_pred)
print("F1 Score:\n", f1)
```

Ce code calcule et affiche le score F1 qui est une mesure de la performance globale du modèle en tenant compte à la fois de la précision et du rappel. Cela fournit une indication de l'équilibre entre la précision et le rappel du modèle dans la classification des données de test.

Output :

F1 Score:	precision	recall	f1-score	support
negatif	0.87	0.87	0.87	40086
positif	0.87	0.87	0.87	39914
accuracy			0.87	80000
macro avg	0.87	0.87	0.87	80000
weighted avg	0.87	0.87	0.87	80000

Nous avons affiché la matrice des F1-score, nous permettant de connaître plusieurs précisions importantes à la performance de notre modèle :

La précision nous a permis d'évaluer la proportion des exemples positifs correctement identifiés parmi tous les exemples classés comme positifs par notre modèle.

Le recall (rappel) représente la capacité de notre modèle à identifier correctement les instances positives parmi toutes les instances réellement positives. Cela nous a permis de mesurer la proportion des vrais positifs par rapport au nombre total d'instances réellement positives.

Le F1-score est la combinaison de la précision et du recall en une seule valeur. Il nous permet de nous fournir une évaluation globale du modèle.

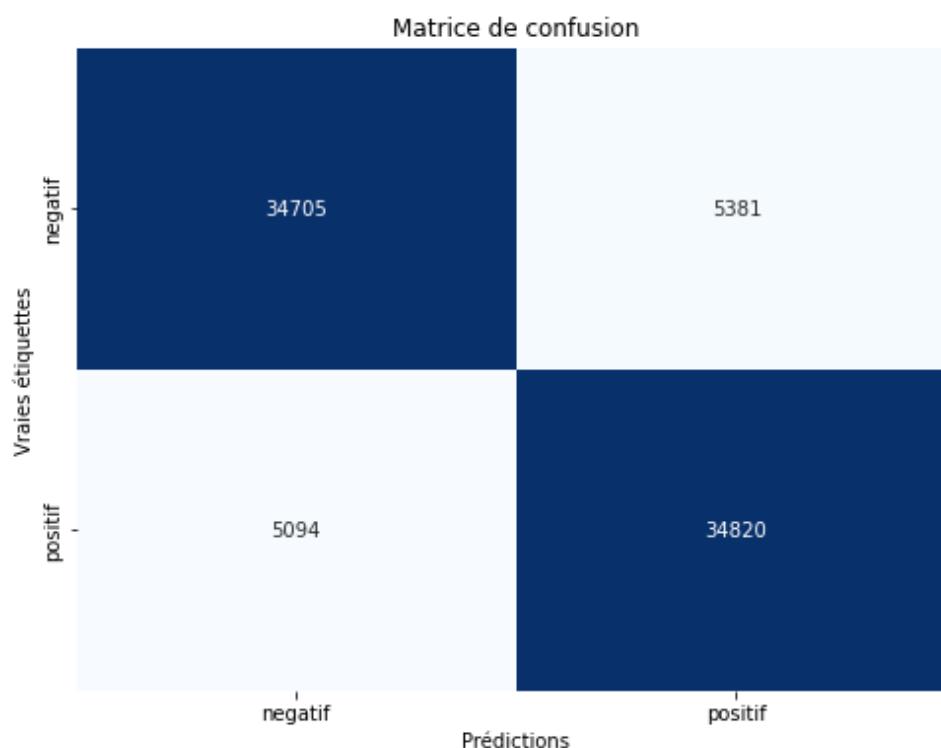
Le support fait référence au nombre d'occurrences dans l'ensemble de données pour chaque classe.

Input :

```
# Créer une matrice de confusion
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False,
            xticklabels=model.classes_, yticklabels=model.classes_)
plt.xlabel("Prédictions")
plt.ylabel("Vraies étiquettes")
plt.title("Matrice de confusion")
plt.show()
```

Ce code génère une matrice de confusion sous forme de heat map pour représenter visuellement les performances du modèle de régression logistique sur les données de test. La matrice de confusion permet d'évaluer la capacité du modèle à classifier correctement les données en montrant les vraies étiquettes sur l'axe y et les prédictions du modèle sur l'axe x. Les cellules de la matrice indiquent le nombre de prédictions correctes et incorrectes pour chaque paire d'étiquettes.

Output :



Nous pouvons voir que le taux de vrai négatif et de 34 705 et de vrais positif 34 820 sur un échantillon d'environ 80 000 commentaire, ce qui est un très bon score. Ce résultat concorde bien avec les résultats de notre matrice des F1-score et notre accuracy de 87%.

Input :

```
# Vérifier si la somme des valeurs dans la matrice de confusion est égale au nombre total d'individus
if total_individuals == len(y_test):
    print("La somme des valeurs dans la matrice de confusion correspond au nombre total d'individus.")
else:
    print("La somme des valeurs dans la matrice de confusion ne correspond pas au nombre total d'individus.")
```

Ce code vérifie si la somme des valeurs dans la matrice de confusion correspond au nombre total d'individus dans l'ensemble de données de test. Si la condition est vérifiée, un message indiquant que la somme des valeurs dans la matrice de confusion correspond au nombre total d'individus est affiché. Sinon, un message indiquant que la somme des valeurs dans la matrice de confusion ne correspond pas au nombre total d'individus est affiché. Cette vérification est importante pour s'assurer de l'intégrité des résultats de la matrice de confusion et éviter des erreurs ou des incohérences dans le calcul des performances du modèle.

Output :

```
La somme des valeurs dans la matrice de confusion correspond au nombre total d'individus.
```

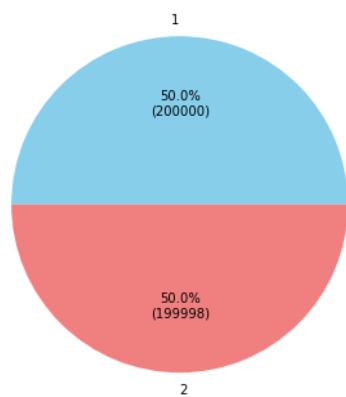
Input :

```
# Diagramme circulaire pour le dataset de base
base_sentiment_counts = df_base['sentiment'].value_counts()
plt.figure(figsize=(6, 6))
base_sentiment_counts.plot(kind='pie', autopct=lambda pct: f'{pct:.1f}%\n({int(pct/100*len(df_base))})',
                           colors=['skyblue', 'lightcoral'])
plt.title("Répartition des commentaires dans le dataset de base")
plt.ylabel("")
plt.show()
```

Ce code crée un diagramme circulaire pour représenter visuellement la répartition des commentaires dans le dataset de base en fonction des classes de sentiment. Chaque part du diagramme représente la proportion de commentaires dans chaque classe, avec des étiquettes indiquant à la fois le pourcentage et le nombre absolu de commentaires correspondant à chaque part.

Output :

Répartition des commentaires dans le dataset de base

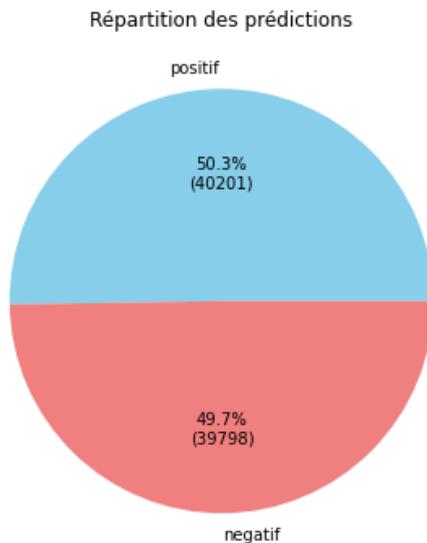


Nous pouvons voir que la répartition des commentaire positif et négatif est de 50% pour les deux.

Input :

```
# Diagramme circulaire pour les prédictions
prediction_counts = pd.Series(y_pred).value_counts()
plt.figure(figsize=(6, 6))
prediction_counts.plot(kind='pie', autopct=lambda pct: f'{pct:.1f}%\n{int(pct/100*len(y_pred))}', 
                       colors=['skyblue', 'lightcoral'])
plt.title("Répartition des prédictions")
plt.ylabel("")
```

Ce code crée un diagramme circulaire pour représenter visuellement la répartition des prédictions faites par le modèle en fonction des classes de sentiment. Chaque part du diagramme représente la proportion de prédictions dans chaque classe, avec des étiquettes indiquant à la fois le pourcentage et le nombre absolu de prédictions correspondant à chaque part.

Output :

Au niveau de la prédiction des sentiment positif et négatif pour les commentaires, nous pouvons retrouver un taux de 50% pour les positifs et 49% pour les négatifs ce qui est un très bon score étant donné qu'il correspondent presque bien aux répartitions dans la base de données.

Avec des résultats de 87% pour chaque élément de notre matrice des F1-score, une matrice de corrélation avec de très bon résultat, des diagrammes circulaires montrant une prédiction des sentiments presque parfaite et un temps d'exécution faible au niveau de notre modèle entraînement, nous pouvons confirmer que ce modèle est très bon pour notre projet.

8. Test Spotify

Après avoir préparé nos données, entraîné notre modèle avec le data set « Amazon » et évalué notre modèle pour être sûr des bonnes performances de ce dernier, nous pouvons mener une séquence de tests sur le data set de « Spotify » afin d'utiliser notre projet en situation réelle :

Préparation des données :

Nous appliquons à ce data set, les différents traitements vus précédemment concernant la préparation de données :

- Suppression des valeurs et colonnes null (NaN)

- Suppression des colonnes inutile
- Suppression de la ligne s'il y a absence de commentaire
- Suppression de tous les caractères non alphanumériques
- Suppression des mots inutile à la compréhension d'un commentaire
- Suppression des doublons

Output :

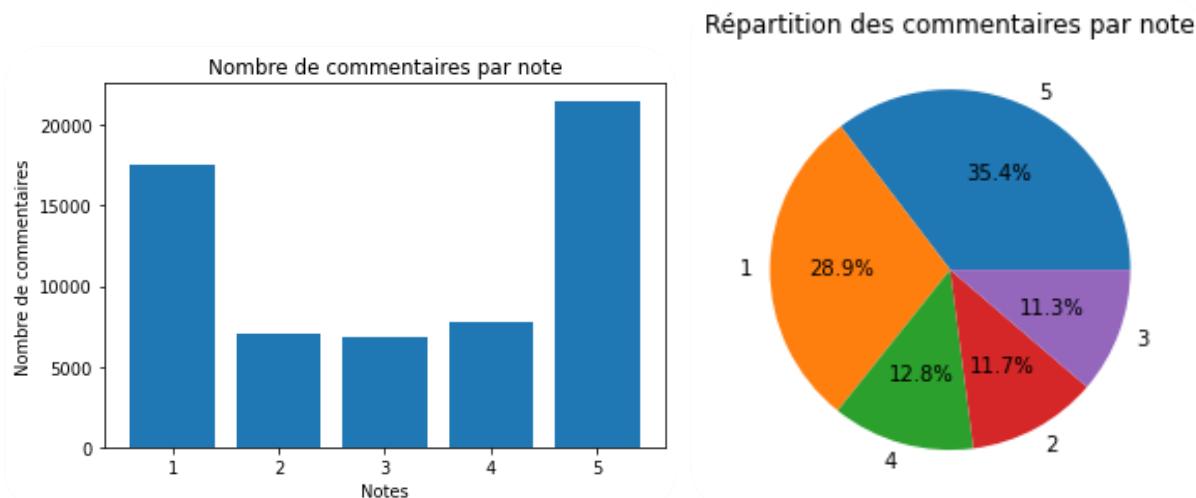
```
Name: Date, Length: 61594, dtype: int64
La colonne spécifiée ne contient pas de valeurs NaN.
présence de doublons avant traitement : True
présence de doublons après traitement : False
```

Après ces différentes étapes, nous pouvons voir que la préparation du data set de Spotify est bien terminé. Nous pouvons voir la réponse du terminal qui nous confirme certaine information cruciale : non présence de valeur null et de doublon après le traitement.

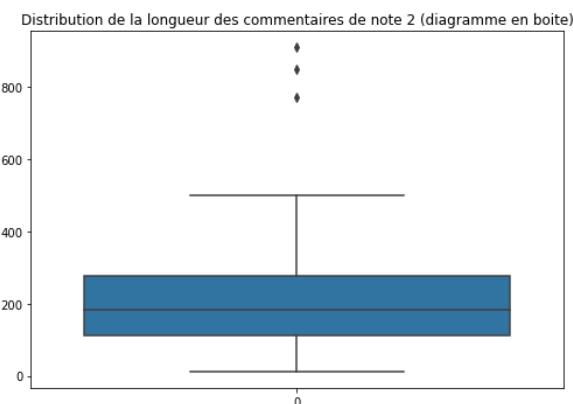
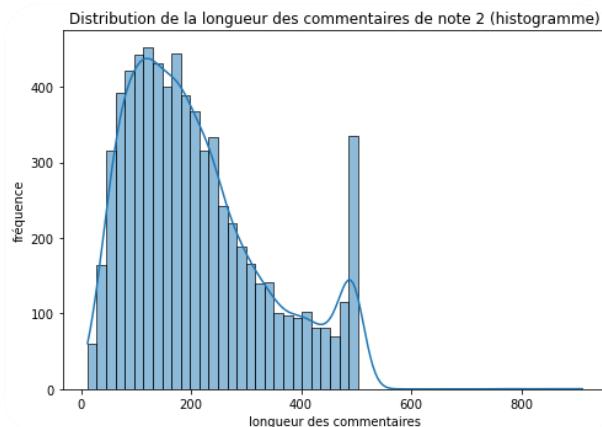
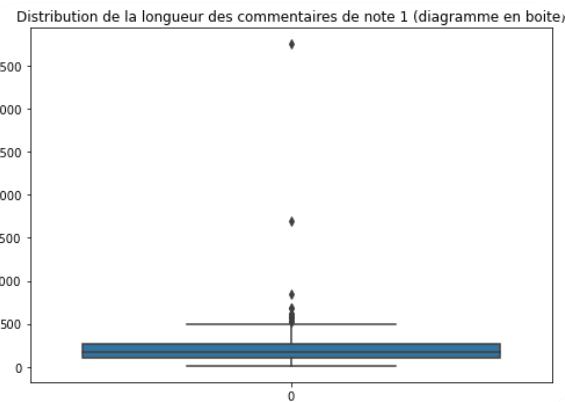
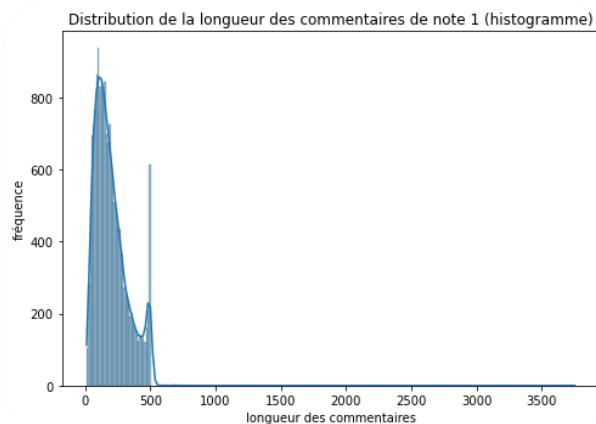
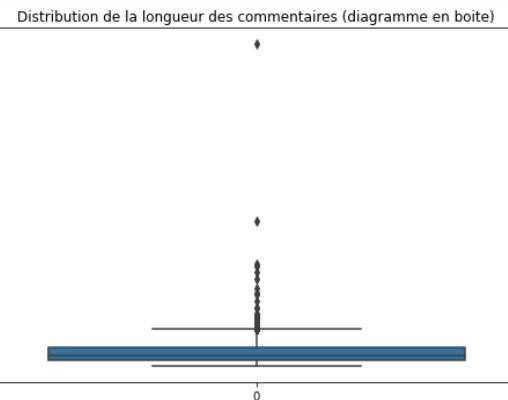
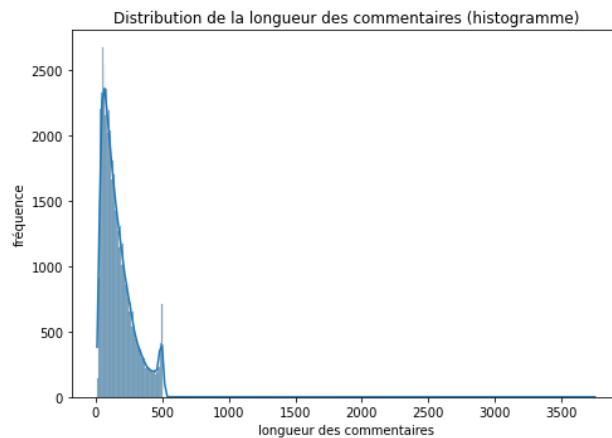
Analyse exploratoire des données :

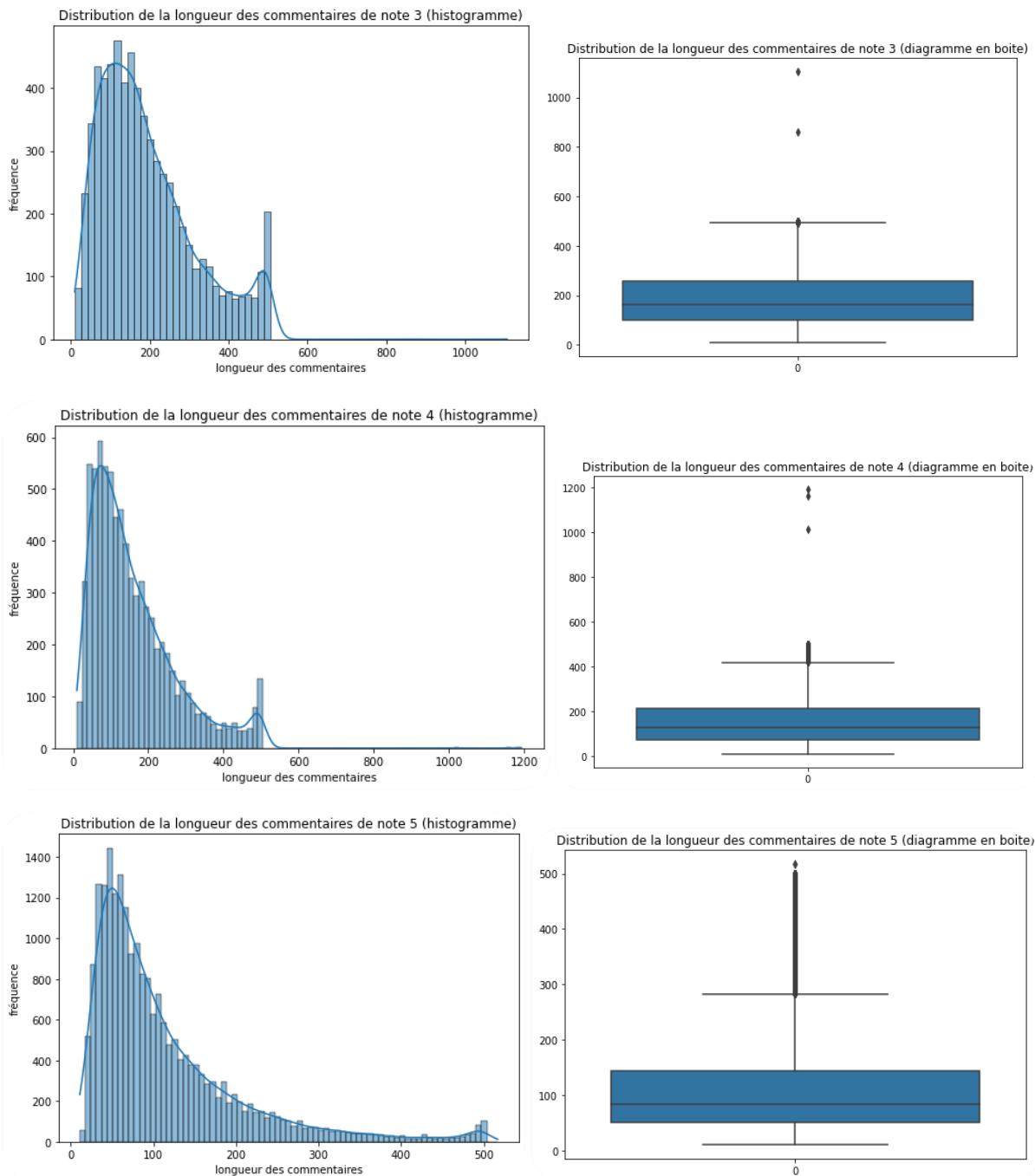
Nous allons ici analyser et exploiter les données de notre data set après l'étape de prétraitement. Cette étape consiste à comprendre la base de données notamment en affichant la répartition de nos variables :

Output :



Ces 2 premiers diagrammes nous permet de comprendre la répartition des commentaires en fonction de chacune des notes attribuées (1 à 5).





Ces différents diagramme (Histogramme et boxlot) nous permettent de comprendre la répartition des commentaire en fonction de chacune des notes (1 à 5) dans notre data set.

Output :

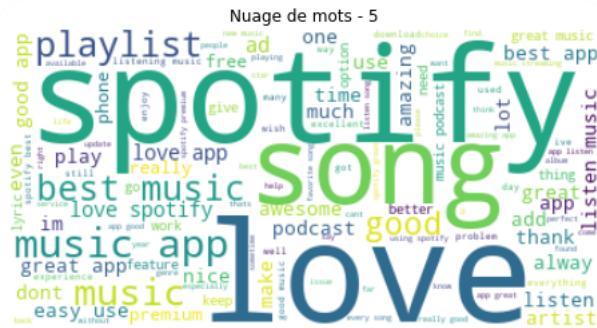
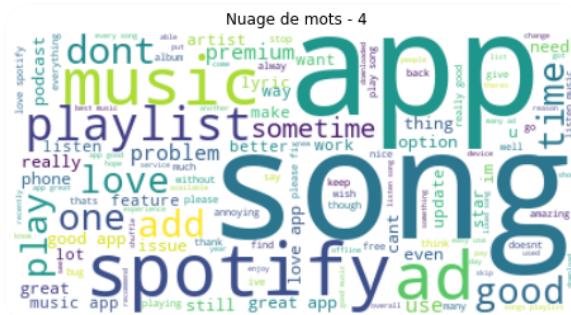
Mots fréquents dans les commentaires :	Mots fréquents dans les commentaires de note 1 :
app 35281	app 11596
music 25018	song 5914
spotify 18680	songs 5706
songs 18183	spotify 5439
song 14670	music 5418
play 11305	play 5090
like 9758	cant 4482
listen 9303	even 3469
cant 9100	premium 3267
premium 8926	listen 2939
dtype: int64	dtype: int64

Mots fréquents dans les commentaires de note 2 :	Mots fréquents dans les commentaires de note 3 :
app 4623	app 4061
songs 2790	songs 2819
song 2491	music 2338
music 2390	song 2235
spotify 2152	spotify 1905
play 2138	play 1635
cant 1641	like 1481
playing 1466	cant 1285
like 1316	ads 1285
premium 1288	good 1164
dtype: int64	dtype: int64

Mots fréquents dans les commentaires de note 4 :	Mots fréquents dans les commentaires de note 5 :
app 4424	music 11938
music 2934	app 10577
songs 2713	spotify 7173
spotify 2011	love 6013
good 1992	songs 4155
song 1813	great 3948
like 1684	good 3541
ads 1443	best 3510
great 1420	listen 2970
love 1320	like 2909
dtype: int64	dtype: int64

mot en commun fréquent dans les commentaires de toutes les notes : {'music', 'app', 'songs', 'spotify'}



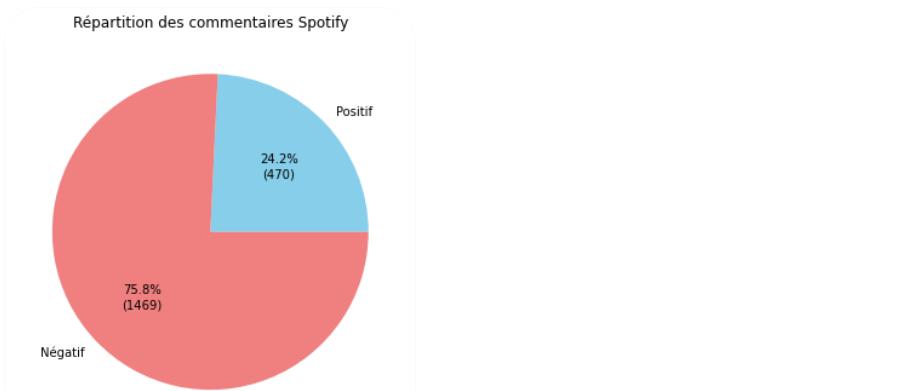


Ces différents affichages concernant les mots les plus fréquents sont utiles pour comprendre et analyser les différents domaines et points d'amélioration. Cela permet aussi de déduire vers quels axes les points d'améliorations devront être orientés.

Apprentissage automatique :

```
--TEST SPOTIFY--  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 61594 entries, 0 to 61593  
Data columns (total 5 columns):  
 #   Column      Non-Null Count Dtype  
 ---  
 0   Time_submitted    61594 non-null object  
 1   Review        61594 non-null object  
 2   Rating         61594 non-null int64  
 3   Total_thumbsup 61594 non-null int64  
 4   Reply          216 non-null  object  
dtypes: int64(2), object(3)  
memory usage: 2.3+ MB  
None
```

Entrez le mot à rechercher : shuffle
 nombre de commentaires négatifs : 1469
 nombre de commentaires positifs : 470



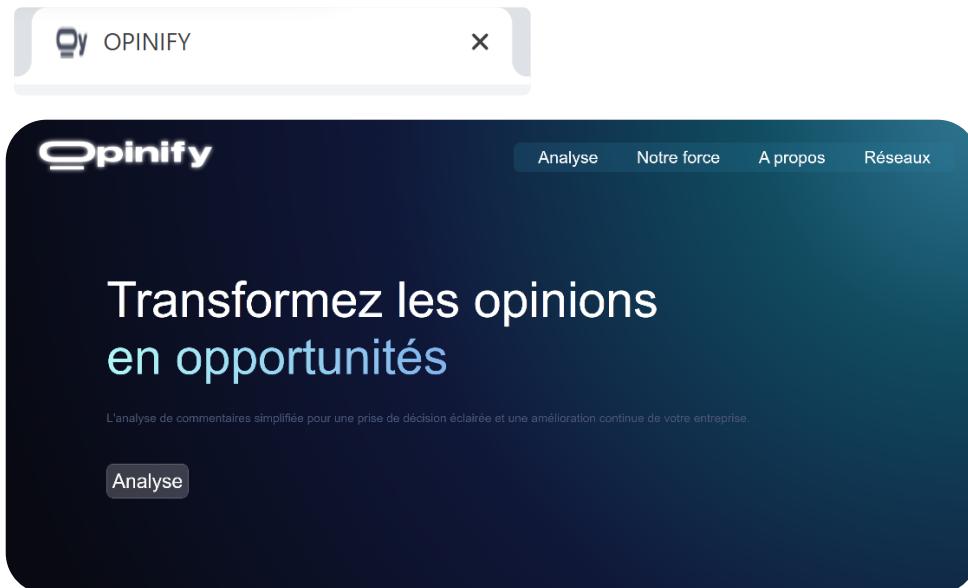
Ces différentes informations nous permettent d'avoir des infos sur notre dataset Spotify. De plus, nous avons décidé d'ajouter une fonctionnalité qui permet d'obtenir des informations sur un mot choisi. Cette fonctionnalité nous permet de voir combien de commentaires comprenant le mot en question ont été émis, ce qui nous a permis par la suite de créer un diagramme circulaire permettant de visualiser la répartition du mot en fonction du sentiment positifs ou négatifs.

9. Interface utilisateur

Afin de combiner notre structure HTML, CSS, et notre code Python, nous avons utilisé la librairie Flask qui est un Framework très utile pour lier notre code python à notre site.

La structure de notre site web a été soigneusement conçue pour offrir aux utilisateurs une expérience de navigation fluide et intuitive. Voici un aperçu des différentes pages disponibles et du contenu qu'elles proposent :

a) La page d'accueil du site :



La page d'accueil de notre site Web constitue la vitrine principale de notre service. Elle est conçue pour captiver et engager les visiteurs dès leur arrivée. Cette section cruciale présente les éléments suivants : un slogan accrocheur, une brève description du service que nous proposons et trois raisons convaincantes pour lesquelles les utilisateurs devraient choisir notre entreprise.

Slogan :



« *Transformez les opinions en opportunités* » est notre slogan qui incarne l'essence de notre entreprise et nos valeurs. Il est soigneusement choisi pour refléter notre identité de marque et créer une impression forte dès le premier contact. Notre objectif est de captiver l'attention du visiteur et de lui donner un aperçu immédiat de ce que nous offrons.

Description du service :

L'analyse de commentaires simplifiée pour une prise de décision éclairée et une amélioration continue de votre entreprise.

La description du service est une brève présentation de notre offre principale. Celle-ci est claire, concise et convaincante afin de permettre aux visiteurs de comprendre rapidement ce que nous proposons. Cette description met en évidence les principaux avantages et fonctionnalités de Opinify, en mettant l'accent sur ce qui nous distingue de nos concurrents.

Raisons de nous choisir :

Pourquoi nous ?

GAIN DE TEMPS

Notre entreprise propose des solutions automatisées et efficaces pour l'analyse de commentaires. Fini la tâche fastidieuse et chronophage d'analyser manuellement un grand nombre de commentaires ! Grâce à nos outils avancés, nous sommes en mesure d'analyser rapidement et efficacement les commentaires, vous permettant de gagner un temps précieux et d'économiser des efforts considérables.

EXPERTISE

Notre entreprise est spécialisée dans l'analyse de commentaires en utilisant des techniques avancées de traitement du langage naturel et de l'apprentissage automatique. Grâce à notre expertise, nous sommes en mesure d'extraire des informations précises et significatives à partir des commentaires de vos clients. En identifiant les sentiments, les opinions et les tendances, nous vous fournissons des insights puissants pour prendre des décisions stratégiques éclairées.

CONFiance

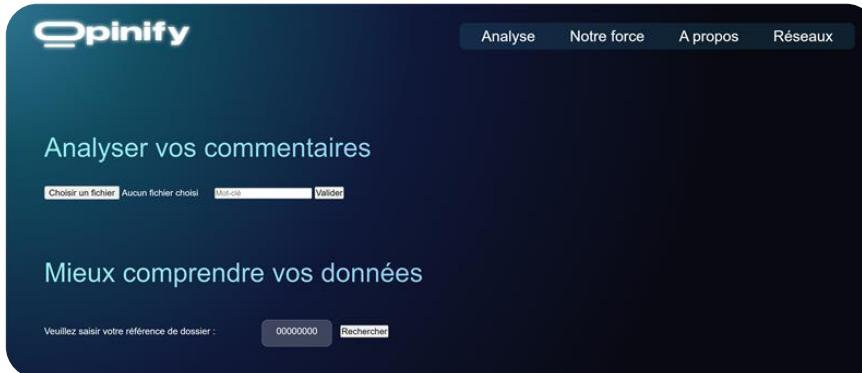
Chez nous, la confiance de nos clients est primordiale. Nous comprenons l'importance de la confidentialité des données et nous nous engageons à ne jamais les divulguer sans leur accord. Nos clients peuvent avoir une totale confiance en notre entreprise, sachant que leurs données sont sécurisées et traitées avec la plus grande confidentialité.

Sur la page d'accueil, nous mettons en évidence trois raisons spécifiques pour lesquelles les utilisateurs devraient choisir notre service. Ces raisons peuvent inclure des éléments tels que :

1. **Fiabilité** : Nous soulignons notre engagement envers la fiabilité et la qualité de service.
2. **Innovation** : Nous mettons en avant notre capacité à innover et à rester à la pointe de notre secteur.
3. **Service client exceptionnel** : Nous mettons l'accent sur notre engagement envers une expérience client exceptionnelle.

Ces trois raisons sont soigneusement sélectionnées pour susciter l'intérêt des visiteurs et les inciter à explorer davantage notre site Web. Elles doivent être présentées de manière concise, claire et convaincante, avec des éléments visuels attrayants pour renforcer l'impact de notre message.

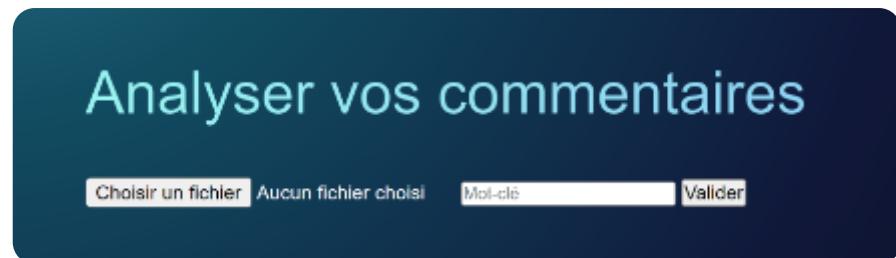
b) Analyse



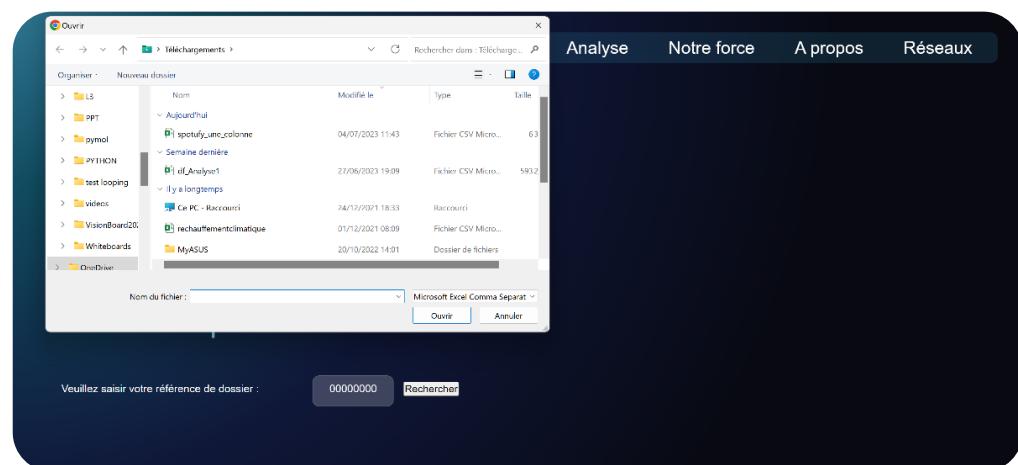
The screenshot shows the Opinify website interface. At the top, there is a navigation bar with the brand name "Opinify" on the left and links for "Analyse", "Notre force", "A propos", and "Réseaux" on the right. Below the navigation, there is a large call-to-action button labeled "Analyser vos commentaires". Underneath this button, there is a form field with a placeholder "Choisir un fichier" and a note "Aucun fichier choisi". To the right of the file input, there are buttons for "Valider" and "Annuler". Further down, another section titled "Mieux comprendre vos données" is visible, featuring a search bar with the placeholder "Veuillez saisir votre référence de dossier:" and a "Rechercher" button. The overall design is clean and modern, with a dark background and white text.

Analyse d'un data set :

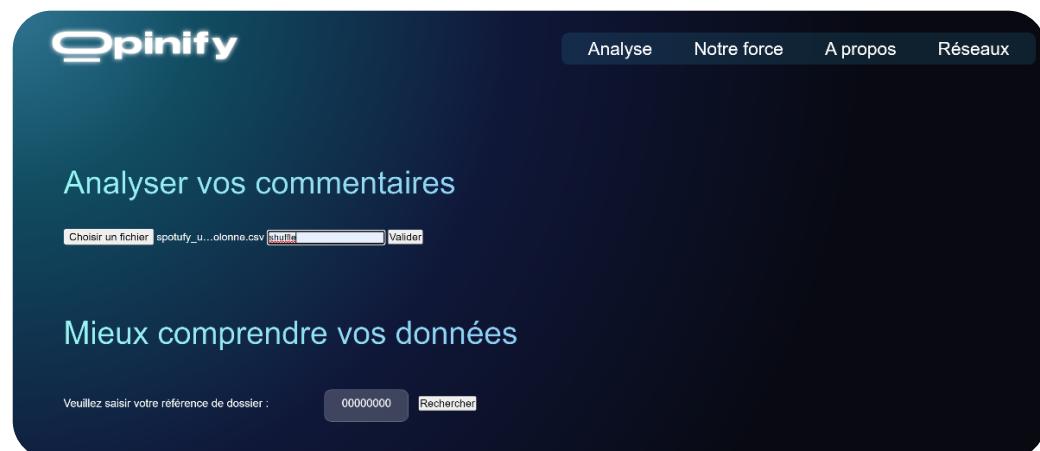
Cette fonctionnalité est automatisée car, il suffit de sélectionner un dataset de type CSV (qui contient 1 colonne de commentaires), pour que les résultats s'affichent. La fonctionnalité est user proof, seuls les fichiers de type CSV sont acceptés



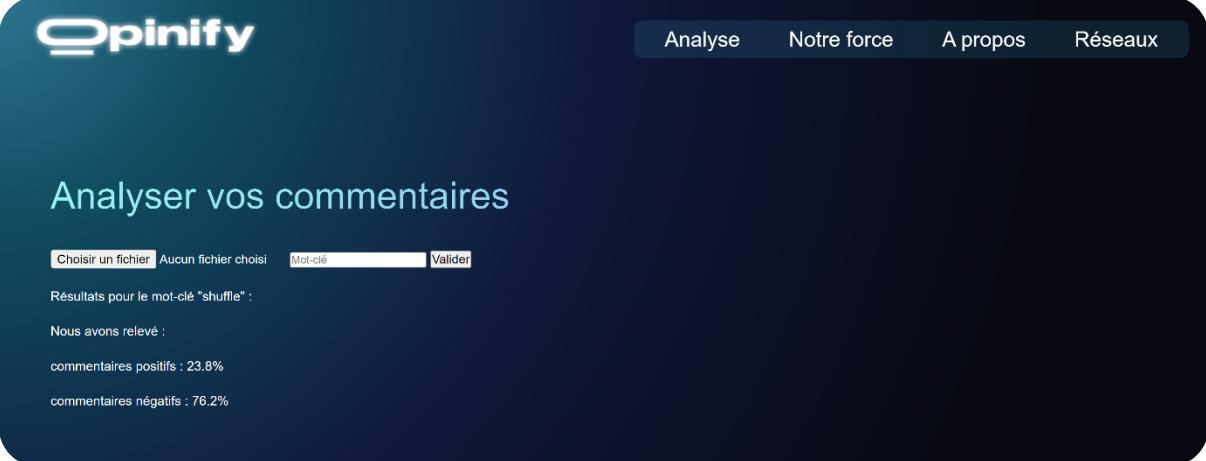
La première fonctionnalité présente sur la page d'accueil de notre site est l'analyse d'un data set. Cette fonctionnalité permet à l'utilisateur de sélectionner un fichier CSV unique à partir de son ordinateur.



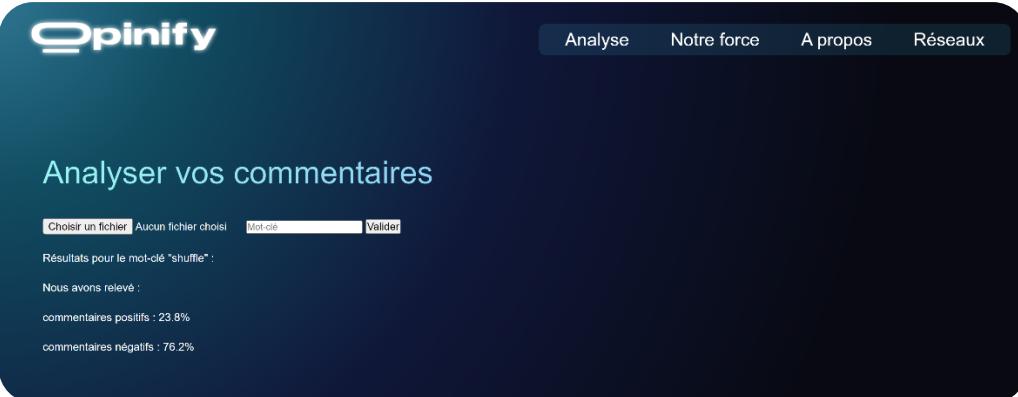
Une fois le fichier sélectionné, l'utilisateur a la possibilité d'insérer un mot clé afin d'obtenir une analyse précise des commentaires contenant ce mot clé.



Si l'utilisateur choisit de ne pas insérer de mot clé, l'analyse sera alors générale. Lorsque l'utilisateur appuie sur le bouton "Valider", l'analyse est lancée.



The screenshot shows the 'Analyse' (Analysis) section of the Opinify platform. At the top, there are four navigation tabs: 'Analyse', 'Notre force', 'A propos', and 'Réseaux'. Below the tabs, the title 'Analyser vos commentaires' (Analyze your comments) is displayed. There are three input fields: 'Choisir un fichier' (Select a file), 'Aucun fichier choisi' (No file selected), 'Mot-clé' (Key word), and a 'Valider' (Validate) button. A message below the fields says 'Résultats pour le mot-clé "shuffle"' (Results for the keyword "shuffle"). It then states 'Nous avons relevé:' (We have found) followed by two lines of text: 'commentaires positifs : 23.8%' and 'commentaires négatifs : 76.2%'.

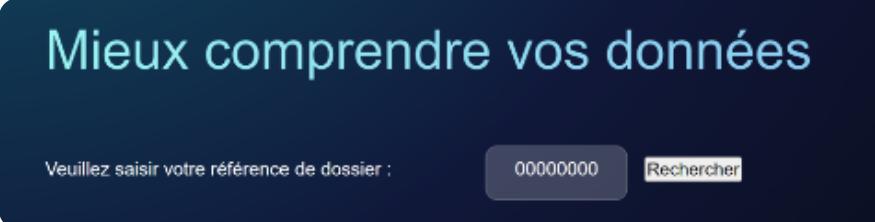


This screenshot is identical to the one above, showing the 'Analyse' section of the Opinify platform. It displays the same interface, including the title 'Analyser vos commentaires', the analysis form with 'Choisir un fichier' and 'Mot-clé' fields, and the results for the keyword 'shuffle' showing 23.8% positive comments and 76.2% negative comments.

L'analyse consiste à examiner le contenu du fichier CSV et à évaluer le pourcentage de commentaires positifs et négatifs à l'aide de notre modèle entraîné. Le résultat de l'analyse est ensuite affiché à l'utilisateur, indiquant le pourcentage de commentaires positifs et négatifs présents dans le dataset.

Accès à l'analyse exploratoire :

L'analyse exploratoire ne pouvant pas être automatisée car elle nécessite une analyse personnalisée selon chaque dataset, nous avons implémenté une fonctionnalité qui permet à l'utilisateur de récupérer un fichier PDF. Ce dernier contient une analyse exploratoire des données approfondies du dataset.



The screenshot shows the 'Mieux comprendre vos données' (Better understand your data) section of the Opinify platform. At the top, there is a search bar with the placeholder 'Veuillez saisir votre référence de dossier :'. To the right of the search bar is a text input field containing '00000000' and a 'Rechercher' (Search) button.

Cette deuxième fonctionnalité permet à l'utilisateur d'accéder à son dossier en utilisant un numéro de dossier spécifique. Sur la page, l'utilisateur peut entrer le numéro de dossier dans un champ dédié, puis appuyer sur le bouton "Rechercher".

Mieux comprendre vos données

Veuillez saisir votre référence de dossier :

zafaqq5

Rechercher

Votre dossier n'existe pas !

Lorsque l'utilisateur entre un numéro de dossier inconnu, un message s'affiche pour l'avertir que ce numéro est inconnu dans notre système. Cela permet d'éviter toute confusion ou erreur lors de la recherche des dossiers.

Mieux comprendre vos données

Veuillez saisir votre référence de dossier :

CAX1oh89

Rechercher

Télécharger le PDF

En revanche, si le numéro de dossier est reconnu, l'utilisateur aura la possibilité de télécharger un fichier PDF associé à ce dossier directement à partir du site :



Annexes

5 / 7 | - 100% + | ☰

En conclusion, l'analyse des mots les plus fréquents dans les commentaires des utilisateurs de Spotify met en évidence les aspects positifs et négatifs liés à l'expérience des utilisateurs. Les mots tels que "songs", "play", "like" et "cant" reviennent régulièrement, indiquant des préoccupations récurrentes des utilisateurs. Les commentaires positifs sont souvent liés aux termes tels que "great", "good", "love" et "best", tandis que les commentaires négatifs se rapportent davantage aux difficultés rencontrées avec la fonctionnalité de lecture et à l'accès réservé aux utilisateurs premium. Cette analyse permet de mieux comprendre les attentes et les opinions des utilisateurs de Spotify, ce qui peut contribuer à l'amélioration continue de l'application et des services proposés.

Annexes



Le fichier PDF téléchargeable contiendra alors l'analyse détaillé mener sur le fichier CSV en contenant des éléments graphiques facilement interprétable.

Cette fonctionnalité facilite l'accès rapide et pratique aux dossiers des utilisateurs. Elle garantit que les utilisateurs autorisés peuvent retrouver et télécharger facilement leurs fichiers PDF sans avoir à contacter notre service client. Cela permet d'améliorer l'efficacité et la convivialité de notre service.

c) A propos :

Opinity

Analyse Notre force A propos Réseaux

Notre entreprise

Votre partenaire pour transformer vos commentaires en opportunités. Chez Opinity, notre mission est de vous aider à comprendre et tirer parti des commentaires de vos clients. Nous sommes nés de la volonté de répondre à un besoin crucial : celui de transformer les précieuses insights contenues dans vos commentaires en actions concrètes et stratégiques. Notre équipe d'experts est là pour vous offrir une vision claire et précise de la perception de votre entreprise, vous permettant ainsi de prendre des décisions éclairées et de renforcer votre position sur le marché.

Description de l'entreprise :

Notre entreprise

Votre partenaire pour transformer vos commentaires en opportunités. Chez Opinity, notre mission est de vous aider à comprendre et tirer parti des commentaires de vos clients. Nous sommes nés de la volonté de répondre à un besoin crucial : celui de transformer les précieuses insights contenues dans vos commentaires en actions concrètes et stratégiques. Notre équipe d'experts est là pour vous offrir une vision claire et précise de la perception de votre entreprise, vous permettant ainsi de prendre des décisions éclairées et de renforcer votre position sur le marché.

Sur la page "À propos" de notre site Web, nous offrons une description complète de notre entreprise. Nous mettons en avant notre mission, notre vision et nos objectifs en soulignant notre engagement envers nos clients et l'excellence dans nos services. Nous expliquons également le secteur d'activité dans lequel nous opérons, en mettant en avant nos points forts et notre positionnement sur le marché.

Dans cette description, nous pouvons également mentionner notre historique, nos réalisations clés et notre réputation dans l'industrie. Nous soulignons nos valeurs fondamentales et notre culture d'entreprise, notamment notre engagement envers l'innovation, la qualité, la satisfaction client et la responsabilité sociale.

Présentation de l'équipe :

L'équipe



Marion NGUYEN

Data analyst &
web développeur

Edwin SAVORY

Développeur
data

Morgan SENECHAL

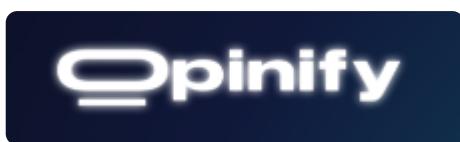
Développeur
dataLionel PIMENTA
SILVA

Chef de projet

La page "À propos" présente également une présentation de notre équipe. Nous mettons en avant les membres clés de notre équipe en présentant leur rôle au sein de l'entreprise. Il est important de souligner les compétences et l'expérience de notre équipe, car cela renforce la crédibilité de notre entreprise et inspire confiance aux visiteurs de notre site.

d) Le header :

Le header de notre site Web constitue une composante essentielle présente sur toutes les pages. Cette partie supérieure fixe de notre site offre une navigation intuitive et permet aux utilisateurs d'accéder facilement à différentes sections du site. Voici les éléments présents dans notre header :

Nom de l'entreprise avec lien vers la page d'accueil :

Le nom de notre entreprise est affiché dans le header et sert de lien cliquable qui redirige les utilisateurs vers la page d'accueil. Cela permet aux visiteurs de revenir rapidement à la page principale du site à tout moment en un seul clic.

Barre de navigation :

La barre de navigation est un élément clé du header, offrant une navigation claire et conviviale vers les différentes sections du site. Voici les options de navigation présentes dans notre barre :

Analyse : Cette option permet aux utilisateurs d'accéder directement à la page d'analyse de données, où ils peuvent sélectionner un fichier CSV, insérer un mot clé et lancer l'analyse pour obtenir les résultats souhaités.

Notre force : Cette option dirige les utilisateurs vers la section "Pourquoi nous choisir" de notre page d'accueil. Cela leur permet de découvrir les raisons convaincantes pour lesquelles ils devraient choisir notre service.

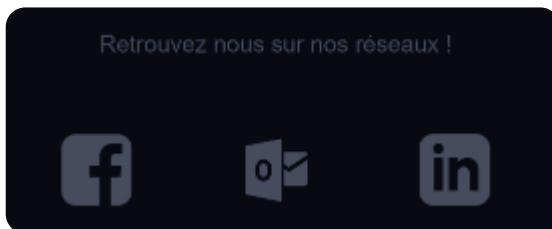
À propos : Cette option amène les utilisateurs à la page "À propos" où ils peuvent en apprendre davantage sur notre entreprise, sa mission, son équipe et ses valeurs.

e) Le footer :

Le footer, présent sur toutes les pages de notre site Web, est une section importante qui fournit des informations complémentaires et des liens utiles aux utilisateurs. Voici les éléments présents dans notre footer :

Logo de l'entreprise :

Le logo de notre entreprise est affiché dans le footer, permettant aux utilisateurs de l'identifier rapidement et de renforcer la reconnaissance de notre marque. Cela crée également une cohérence visuelle avec le reste du site.

Accès aux réseaux sociaux :

Le footer comprend des icônes de différents réseaux sociaux, tels que Facebook, Outlook et LinkedIn. Ces icônes servent de liens cliquables qui redirigent les utilisateurs vers ces réseaux.

Présence sur toutes les pages :

Le footer est présent sur toutes les pages de notre site Web, en bas de chaque page. Cela permet aux utilisateurs de trouver facilement les informations supplémentaires, les liens utiles et de naviguer rapidement vers d'autres parties du site.

Le choix du design :

Dans le cadre de notre projet, le choix du design de notre site web a été soigneusement considéré pour répondre à plusieurs critères importants. Voici les principaux aspects pris en compte lors de la conception du design :

Thème moderne et innovant :

Nous avons opté pour un thème moderne et innovant qui reflète l'esprit de notre entreprise et de nos services. Le design de notre site web est conçu de manière à être à la pointe des dernières tendances en matière de conception web, avec une esthétique contemporaine et engageante.

Couleur :

Nous avons choisi une palette de couleurs foncée avec un gradient de bleu. Cette sélection de couleurs apporte plusieurs avantages : elle crée une atmosphère agréable et professionnelle, elle est également associée à l'économie d'énergie et offre une expérience visuelle agréable aux utilisateurs. De plus, le contraste entre les éléments du site et le fond foncé améliore la lisibilité et met en valeur les informations clés.

Site interactif, dynamique, responsive et minimaliste :

Nous avons mis l'accent sur l'interactivité et la dynamique de notre site web pour offrir une expérience utilisateur immersive. Les éléments interactifs, tels que les boutons cliquables et les animations subtiles, permettent aux utilisateurs d'interagir facilement avec le contenu du site. De plus, notre site web est conçu pour être responsive, c'est-à-dire qu'il s'adapte de manière optimale à différents appareils et tailles d'écran, offrant ainsi une expérience cohérente et conviviale sur les ordinateurs de bureau, les tablettes et les smartphones.

Nous avons également adopté une approche minimalisté dans la conception, en évitant les éléments superflus et en mettant l'accent sur la clarté et la simplicité. Cela permet aux utilisateurs de naviguer intuitivement et de trouver facilement les informations dont ils ont besoin.

Interface intuitive :

L'interface de notre site web a été conçue de manière à être intuitive, c'est-à-dire que les utilisateurs peuvent naviguer facilement et trouver rapidement les informations recherchées. Les éléments de navigation, tels que les menus déroulants et les icônes explicites, sont placés de manière stratégique pour faciliter la découverte du contenu et l'accès aux différentes sections du site.

Les fonctionnalités :

Dans le cadre de notre projet, notre site web propose deux fonctionnalités principales qui visent à offrir une expérience enrichissante aux utilisateurs. Voici les fonctionnalités clés de notre site web :

Analyse automatique du data set :

L'une des fonctionnalités principales de notre site web est l'analyse automatique du data set. Les utilisateurs ont la possibilité de télécharger un fichier de type CSV à partir de leur ordinateur. Une

fois le fichier sélectionné, notre système se charge de l'analyser automatiquement, en extrayant des informations pertinentes et en fournissant des résultats précis.

Cette fonctionnalité permet aux utilisateurs d'obtenir une analyse approfondie de leur data set sans avoir à effectuer des calculs ou des manipulations complexes. Elle facilite le processus d'exploration des données et offre des informations utiles aux utilisateurs pour prendre des décisions éclairées.

Accès à l'analyse exploratoire :

En plus de l'analyse automatique du data set, notre site web permet également aux utilisateurs d'accéder à une fonctionnalité d'analyse exploratoire. Cette fonctionnalité offre une interface conviviale où les utilisateurs peuvent explorer et visualiser leurs données de manière interactive.

Grâce à cette fonctionnalité, les utilisateurs peuvent effectuer des analyses plus approfondies en utilisant des outils de visualisation avancés tels que des graphiques, des tableaux et des diagrammes. Ils peuvent filtrer, trier et regrouper les données selon leurs besoins, ce qui leur permet de découvrir des tendances, des corrélations et des insights importants.

En combinant l'analyse automatique du data set avec l'analyse exploratoire, notre site web offre aux utilisateurs une gamme complète d'outils d'analyse de données, leur permettant d'obtenir des informations précieuses et de mieux comprendre leurs données.

II. Impact économique sur nos potentielles clients

Pouvoir analyser proprement et rapidement l'avis de ses clients permet aussi de faire des économies non négligeables et ce de plusieurs manières. Premièrement la réputation de l'entreprise s'en trouvera grandement améliorée, en effet si les entreprises montrent qu'elles sont à l'écoute de leurs clients, elle sera en mesure de prendre des décisions de modification et/ou d'ajouts de nouvelles technologies beaucoup plus rapidement. Cela peut engendrer une augmentation des ventes et des revenus.

L'IA de filtrage permet de plus de faire, évidemment une économie au niveau des coûts opérationnels. Les IA permettent d'automatiser le processus de tri et donc de diminuer les couts de main-d'œuvre.

1. La concurrence

Notre modèle de filtrage des avis utilise des algorithmes avancés d'apprentissage automatique pour analyser et comprendre le contenu des avis des utilisateurs. Cela nous permet de fournir des résultats précis et pertinents lorsqu'un utilisateur recherche un mot spécifique, tel qu'une fonctionnalité du secteur d'activité de l'entreprise. Notre modèle est continuellement entraîné et amélioré pour garantir la meilleure qualité de filtrage et des résultats fiables.

Notre modèle utilise des techniques d'analyse sémantique avancée pour comprendre le sens et le contexte des avis des utilisateurs. Il ne se contente pas de rechercher simplement des mots-clés, mais il est capable de comprendre les nuances, les expressions et les opinions exprimées dans les avis. Cela permet d'obtenir une vision plus complète et détaillée de ce que les utilisateurs pensent réellement d'une fonctionnalité spécifique.

Nous avons mis un point d'honneur à concevoir une interface utilisateur conviviale et intuitive. Notre projet offre une expérience utilisateur fluide, avec une facilité de navigation et une simplicité d'utilisation. Les utilisateurs peuvent entrer leurs requêtes de manière rapide et facile, et les résultats sont présentés de manière claire et compréhensible.

2. Nos objectifs futurs

Nos objectifs futurs sont de rajouter un algorithme de scraping HTML qui nous permettra de créer un fichier CSV uniquement à partir d'un lien web.

Nous voudrions aussi améliorer la gestion des ambiguïtés et des subtilités. Ainsi les avis des utilisateurs qui peuvent parfois contenir des ambiguïtés ou des subtilités qui rendent leur interprétation plus complexe seront mieux traités. L'amélioration de notre algorithme peut impliquer l'intégration de techniques de compréhension contextuelle et de sémantique plus avancées pour traiter ces cas particuliers et garantir des résultats plus précis.

Conclusion

En conclusion, le projet OPINIFY a apporté de nombreux avantages à notre équipe en tant qu'étudiants en ingénierie informatique et numérique. Tout d'abord, il nous a permis d'appliquer concrètement les connaissances acquises au cours de notre formation, en mettant en pratique des concepts avancés tels que le traitement du langage naturel et l'apprentissage automatique.

En travaillant sur ce projet, nous avons également pu développer nos compétences en matière de collaboration et de gestion de projet. Travailler en équipe de quatre personnes nous a permis de mettre en œuvre une répartition efficace des tâches, de communiquer de manière cohérente et de tirer parti des compétences et des idées de chacun. Cette expérience a renforcé notre capacité à travailler en équipe dans un environnement professionnel.

De plus, en réalisant ce projet, nous avons également amélioré notre compréhension des défis et des enjeux auxquels font face les entreprises dans le domaine de l'analyse des avis clients. Cette connaissance pratique nous sera précieuse dans nos futures carrières, en nous permettant d'apporter des solutions innovantes et adaptées aux besoins des entreprises.

Enfin, le projet OPINIFY nous a donné l'occasion de continuer à développer notre portfolio professionnel en créant une solution concrète et fonctionnelle. Cela nous permettra de présenter notre travail lors de futurs entretiens d'embauche et de démontrer notre capacité à relever des défis techniques.

Nous sommes fiers des résultats obtenus et confiants dans le fait que cette expérience nous servira de base solide pour notre future carrière dans le domaine de l'informatique et du numérique, et sommes prêts à relever de nouveau défis.