

ETUDE DE CAS



Travail réalisé par:
Cherif Meriam



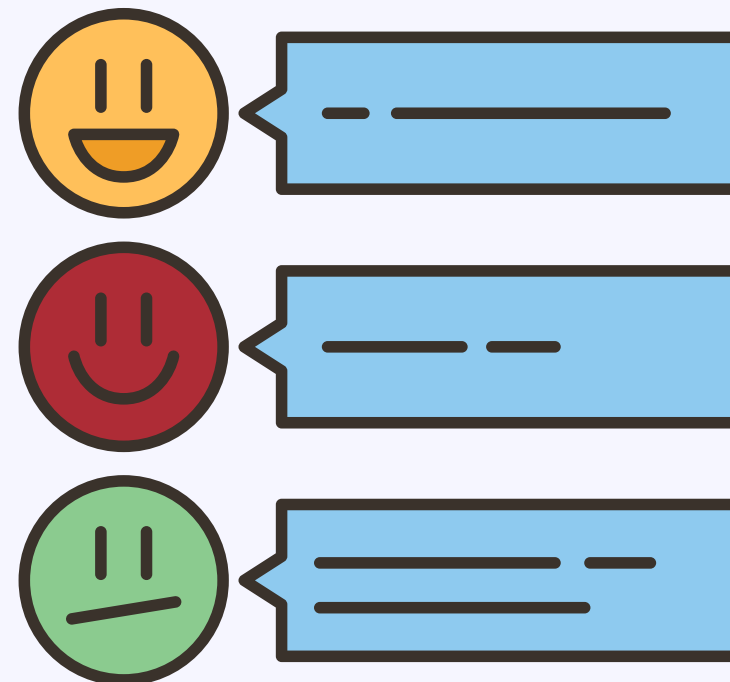
Plan :

- 1.Introduction
- 2.Démarche de travail
- 3.Librairies testées et utilisées
- 4.KPI monitorés
- 5.Application avec Interface
Utilisateur
- 6.Perspectives d'amélioration

INTRODUCTION

Pourquoi créer un algorithme intelligent de réponse aux avis?

- **Le pouvoir important des avis clients dans l'ère numérique**
- **Faciliter et accélérer les réponses aux avis**



DÉMARCHE DE TRAVAIL :



1. Collecte de données
2. Extraction de la polarité
3. Extraction de la catégorie
4. Extraction du sentiment

1. COLLECTE DE DONNÉES



On a choisi un jeu de données:

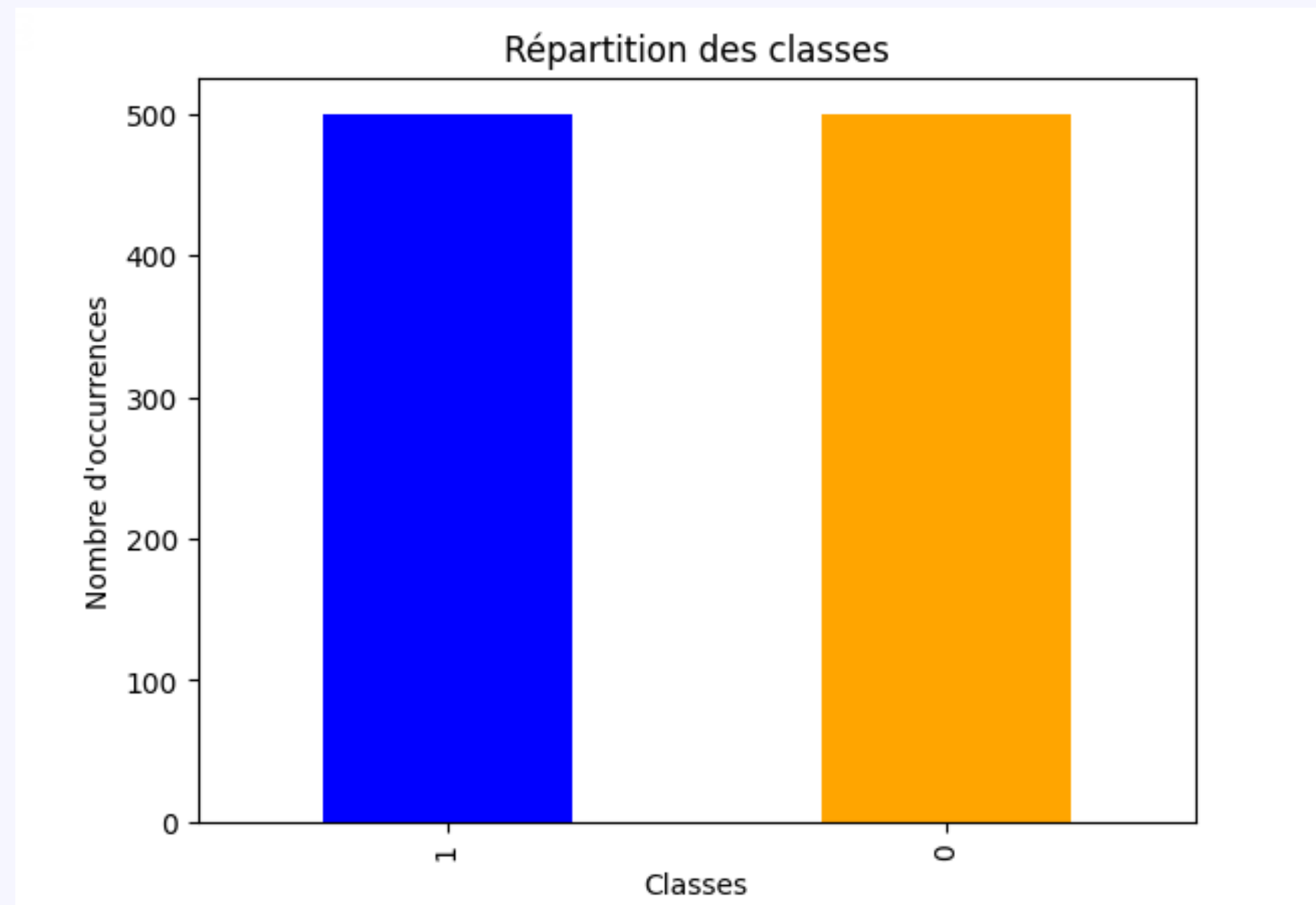
- **en anglais**
- **disponible sur Kaggle, accessible via le lien suivant:**

<https://www.kaggle.com/datasets/mawro73/restaurant-reviews-for-nlp/>

- **comptant 1000 lignes.**

Voilà un aperçu des données:
Ce sont des avis de restaurants
comprenant deux colonnes :
'Review', qui contient l'avis, et
'Liked', avec des valeurs de 0 si la
polarité de l'avis est négative et 1
sinon.

	Review	Liked
0	Wow... Loved this place.	1
1	Crust is not good.	0
2	Not tasty and the texture was just nasty.	0
3	Stopped by during the late May bank holiday of...	1
4	The selection on the menu was great and so wer...	1



La répartition des classes dans notre dataset est équilibrée : 500 positifs et 500 négatifs.

2. EXTRACTION DE LA POLARITÉ

- **Déterminer si un avis est positif ou négatif**
- **Classification avec Naive Bayes: Modèle de classification supervisée**
- **Segmentation des avis (Evaluation granulaire)**




```
[67] review="the food was good. the waiters wer polite."
total_polarity,negative_sentences,positive_sentences = predict_polarity_per_sentence(review)
print(f"Polarité totale de l'avis: {total_polarity}")
```

```
1/1 [=====] - 0s 15ms/step
0
the food was good.
1/1 [=====] - 0s 17ms/step
0
the waiters wer polite.
Polarité totale de l'avis: negative
```

Logique de polarité:

- Si toutes les phrases de l'avis sont positives, l'avis est considéré comme positif.
- Si l'une des phrases de l'avis est négative, tout l'avis est considéré comme négatif.

3. EXTRACTION DE LA CATÉGORIE

La liste des catégories qu'un avis peut aborder:

- Food
- Service
- Ambiance
- Price/Quality
- Overall Experience
- Cleanliness



- Création de la Liste des Mots-Clés par Catégorie
- Extraction des mots clés de l'avis
- Calcul de la similarité entre les mots-clés de chaque catégorie et les mots-clés extraits de l'avis client.
(coefficient de similarité Jaccard)
- Association de l'avis à la catégorie dont les mots-clés correspondent le mieux

```
[51] category = find_category("everything was bad")  
      print(f"The category for the review is: {category}")
```

```
The category for the review is: Overall Experience
```

L'extraction de la catégorie est réalisée par la fonction `find_category`

4. EXTRACTION DU SENTIMENT

L'extraction du sentiment est réalisée par le biais des Transformers.

```
# a function to determine the feeling
from transformers import pipeline
def feeling(review):
    classifier = pipeline("text-classification", model="j-hartmann/emotion-english-distilroberta-base", return_all_scores=True)
    result=classifier(review)
    max_score_label = max(result[0], key=lambda x: x['score'])['label']
    max_score = max(result[0], key=lambda x: x['score'])['score']
    return (max_score_label)
```

```
sentiment=feeling("i am disgusted with the food eu3")
print(sentiment)
```

```
disgust
/usr/local/lib/python3.10/dist-packages/transformers/pipeli
warnings.warn(
```

Le modèle utilisé dans le pipeline de classification d'émotions est conçu pour prédire six émotions différentes. Ces émotions sont généralement les suivantes :

- 1. Joy (Joie)**
- 2. Sadness (Tristesse)**
- 3. Anger (Colère)**
- 4. Surprise (Surprise)**
- 5. Fear (Peur)**
- 6. Disgust (Dégoût)**

LIBRAIRIES TESTÉES ET UTILISÉES

- **spaCy** : Utilisé pour le prétraitement des textes
- **nltk**: Utilisé pour la segmentation en phrases des avis clients
- **scikit-learn** : Utilisé pour les modèles de machine learning, notamment pour le classificateur Naive Bayes.
- **XGBoost** : Utilisé comme alternative pour le classificateur
- **pandas** : Utilisé pour manipuler et analyser les données, notamment pour le chargement du dataset.

- **numpy** : Utilisé pour les opérations sur les tableaux et les matrices.
- **Flask** : Utilisé pour créer une interface web.
- **TensorFlow/Keras** : Utilisé pour la création et l'entraînement d'un modèle de deep learning.

KPI MONITORÉS

- **Accuracy (dans la partie de l'extraction de la polarité)**
On a testé avec MultinomialNB et on a obtenu un score 76.5%
On a également testé avec XGBClassifier et on a obtenu un score 72.0%
- **Temps de réponse moyen: on a obtenu 2.215280294418335 secondes (test sur 5 exemples)**

APPLICATION AVEC INTERFACE UTILISATEUR

Review Analysis App

Enter your review:

food was very bad. the waiters were so impolite

Get Response

Response:

Greetings! We are sorry to hear about your experience with our Food and Service. Your feedback is valuable, and we'd like to address any concerns you have. Please let us know how we can enhance your experience next time.

PERSPECTIVES D'AMÉLIORATION

- Utilisation du Prompting et OpenAI
- Accepter des avis de différentes langues (Traduction)
- Préparation d'un dataset Client-Spécifique
- Développer la liste des mots associés à chaque catégorie
- Améliorer l'Accuracy de la détermination de la polarité

**MERCI POUR VOTRE
ATTENTION**