

# Test technique

Ce rapport présente une analyse des avis récupérés pour la franchise Tim Hortons située à 1257 Rue Peel, Montréal, QC H3B 5L4. L'analyse a été effectuée à l'aide de l'API SerpApi pour récupérer les avis de Google Maps et du module NLTK pour l'analyse de sentiment.

Voici les étapes effectuées lors de cette analyse:

1. **Collecte des données** : Nous avons utilisé l'API SerpApi avec la clé API appropriée et le **data\_id** spécifique à la franchise Tim Hortons pour récupérer les avis publics disponibles sur Google Maps.
2. **Analyse de sentiment** : Chaque avis a été analysé en utilisant l'outil VADER (Valence Aware Dictionary and SEntiment Reasoner) de NLTK, qui attribue un score de sentiment composé (compound score) à chaque avis. Les avis ont été classés comme positifs, négatifs ou neutres en fonction de ce score.
3. **Visualisation des données** : Les résultats ont été présentés sous forme d'histogramme pour illustrer la répartition des avis par sentiment et le score de sentiment.

## 1. Collecte des données

Pour la collecte des données, nous avons utilisé l'API SerpApi pour récupérer les avis Google de la franchise Tim Hortons basée sur la rue Peel à Montréal.

```
In [8]: import os
from serpapi import search
from dotenv import load_dotenv
import json
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import matplotlib.pyplot as plt

# Charger les variables d'environnement à partir du fichier .env
load_dotenv()

# Obtenir la clé API à partir des variables d'environnement
api_key = os.getenv('API_KEY')

# Initialisation de l'analyseur de sentiment VADER
nltk.download('vader_lexicon')
sid = SentimentIntensityAnalyzer()

def get_google_reviews(api_key, place_id, max_reviews=100):
    reviews = []
    num_reviews_per_page = 10
    start = 0
    while len(reviews) < max_reviews:
        params = {
            "engine": "google_maps_reviews",
            "data_id": place_id,
            "api_key": api_key,
            "hl": "fr", # Langue des avis
            "start": start
        }
        results = search(params)

        if "reviews" in results:
            reviews.extend(results["reviews"])
            start += num_reviews_per_page
        else:
            break # Sortir de la boucle si aucun avis n'est retourné

    return reviews[:max_reviews]
```

- **SerpApi** : Utilisé pour récupérer les avis Google car il permet de contourner certaines limitations des autres APIs.
- **NLTK** : Utilisé pour l'analyse de sentiments grâce à la simplicité et à l'efficacité de la librairie VADER incluse.
- **Matplotlib** : Utilisé pour visualiser les résultats de manière claire et compréhensible.

Nous avons aussi pu observer les avis sous format json en utilisant la fonction `get_google_reviews()` avec les paramètres appropriés.

```
# Clé API SerpApi
place_id = '0x4cc91a43c80dabfb:0x9e34b0920201835f'

# Récupérer les avis
reviews = get_google_reviews(api_key, place_id, max_reviews=329)

# Vérifiez si les avis ont été récupérés
if not reviews:
    print("No reviews found.")
else:
    # Affichage des avis en format json
    print(json.dumps(reviews, indent=4, ensure_ascii=False))
```

```
{
  "link": "https://www.google.com/maps/reviews/data=!4m8!14m7!1m6!2m5!1sChZDSUhNMG9nS0VJQ0FnSUREOVpITGRREAE!2m1!1s0x0:0x9e34b0920201835f!3m1!1s2@1:CIHM0ogKEICAgIDD9ZHLdQ%7CCgwIILP1sAYQIMHwtgE%7C?hl=fr",
  "rating": 1.0,
  "date": "il y a 2 mois",
  "iso_date": "2024-04-12T15:53:07Z",
  "iso_date_of_last_edit": "2024-04-12T15:56:04Z",
  "images": [
    "https://lh5.googleusercontent.com/p/AF1QipPO5UUXUDCb61YcmQxL7ucW9FouyxLkw4ipsJze=w150-h150-k-no-p"
  ],
  "source": "Google",
  "review_id": "ChZDSUhNMG9nS0VJQ0FnSUREOVpITGRREAE",
  "user": {
    "name": "Nikki T",
    "link": "https://www.google.com/maps/contrib/114980966510455434128?hl=fr",
    "contributor_id": "114980966510455434128",
    "thumbnail": "https://lh3.googleusercontent.com/a/ACg8ocIMSTSwmNedH5pml7H9B09e8X6GzhvXF6xPrx6P5PrI-kFHTCw=s120-c-rp-mo-br100",
    "reviews": 2,
    "avatar": 1
  }
}
```

## 2. Analyse de sentiment

Nous avons utilisé NLTK pour analyser les sentiments des avis et les classer en trois catégories (positifs, négatifs et neutres).

```
In [9]: def analyze_reviews(reviews):
        positive_reviews = []
        negative_reviews = []
        neutral_reviews = []
        compound_scores = []

        for review in reviews:
            text = review['snippet']
            scores = sid.polarity_scores(text)
            compound_scores.append(scores['compound'])
            if scores['compound'] >= 0.05:
                positive_reviews.append(text)
            elif scores['compound'] <= -0.05:
                negative_reviews.append(text)
            else:
                neutral_reviews.append(text)

        return positive_reviews, negative_reviews, neutral_reviews, compound_scores

positive_reviews, negative_reviews, neutral_reviews, compound_scores = analyze_reviews(reviews)

# Affichage des résultats
print(f"Nombre total d'avis : {len(reviews)}")
print(f"Nombre d'avis positifs : {len(positive_reviews)}")
print(f"Nombre d'avis négatifs : {len(negative_reviews)}")
print(f"Nombre d'avis neutres : {len(neutral_reviews)}")

# Calcul de la moyenne des scores de sentiment
if compound_scores:
    average_score = sum(compound_scores) / len(compound_scores)
    print(f"\nScore moyen de sentiment : {average_score:.2f}")
else:
    print("Pas de scores de sentiment disponibles pour calculer la moyenne.")

Nombre total d'avis : 329
Nombre d'avis positifs : 82
Nombre d'avis négatifs : 247
Nombre d'avis neutres : 0

Score moyen de sentiment : -0.35
```

L'analyse sentimentale montre que cet franchise admet 329 avis au total réparties ainsi :

- Nombre d'avis positifs : 82
- Nombre d'avis négatifs : 247
- Nombre d'avis neutres : 0

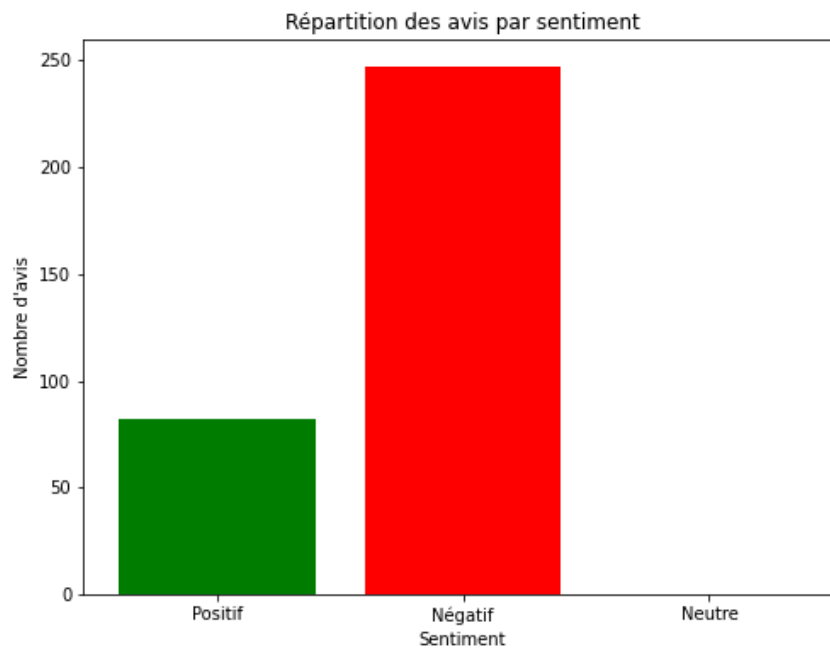
Et nous avons obtenu un score moyen de -0.35 .

### 3. Visualisation des données

Pour la visualisation des données, nous avons mis en place des graphiques et le calcul de la moyenne des sentiments pour bien illustrer nos résultats.

```
In [10]: # Création de l'histogramme
sentiments = ['Positif', 'Négatif', 'Neutre']
counts = [len(positive_reviews), len(negative_reviews), len(neutral_reviews)]

plt.figure(figsize=(8, 6))
plt.bar(sentiments, counts, color=['green', 'red', 'blue'])
plt.xlabel('Sentiment')
plt.ylabel('Nombre d\'avis')
plt.title('Répartition des avis par sentiment')
plt.show()
```



**Conclusion :** L'analyse des avis de la franchise Tim Hortons montre une répartition déséquilibrée des avis, avec une majorité d'avis négatifs et un score moyen négatif. Cela indique une non satisfaction générale des clients, bien que des avis positifs soient également présents. Ces résultats peuvent être utilisés pour identifier les points faibles et les domaines d'amélioration potentiels de l'établissement.

Sur la base de cette analyse, je recommande :

- D'investir davantage dans la qualité du service à la clientèle pour maintenir et renforcer les avis positifs.
- D'examiner les avis négatifs pour identifier les problèmes spécifiques et mettre en œuvre des mesures correctives appropriées.