Table des matières

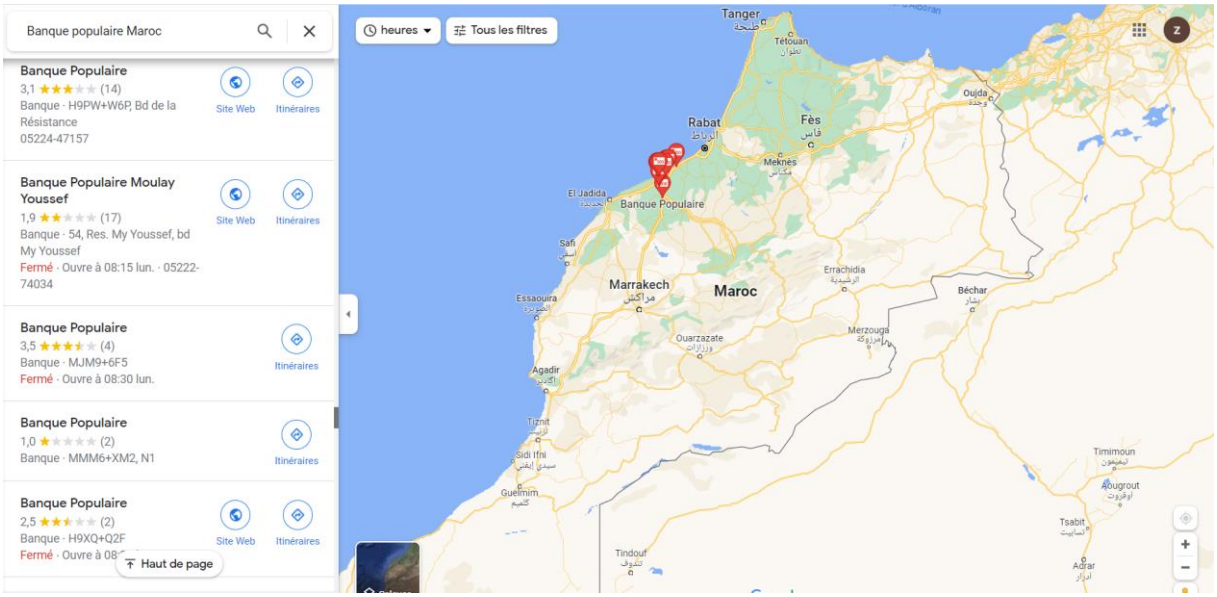
Introduction :

Ce projet a pour objectif d'analyser les commentaires des clients sur les agences de la Banque Populaire du Maroc. Pour ce faire, j'ai utilisé la technique de web scraping en utilisant Apify pour extraire les données des commentaires à partir des pages des agences dans Google Maps. Ensuite, on a appliqué différentes étapes de nettoyage de données pour préparer les commentaires en vue de leur analyse. Enfin, j'ai entraîné un modèle basé sur BERT pour détecter les sentiments des commentaires et j'ai réalisé une visualisation des données pour mieux comprendre les tendances et les patterns.

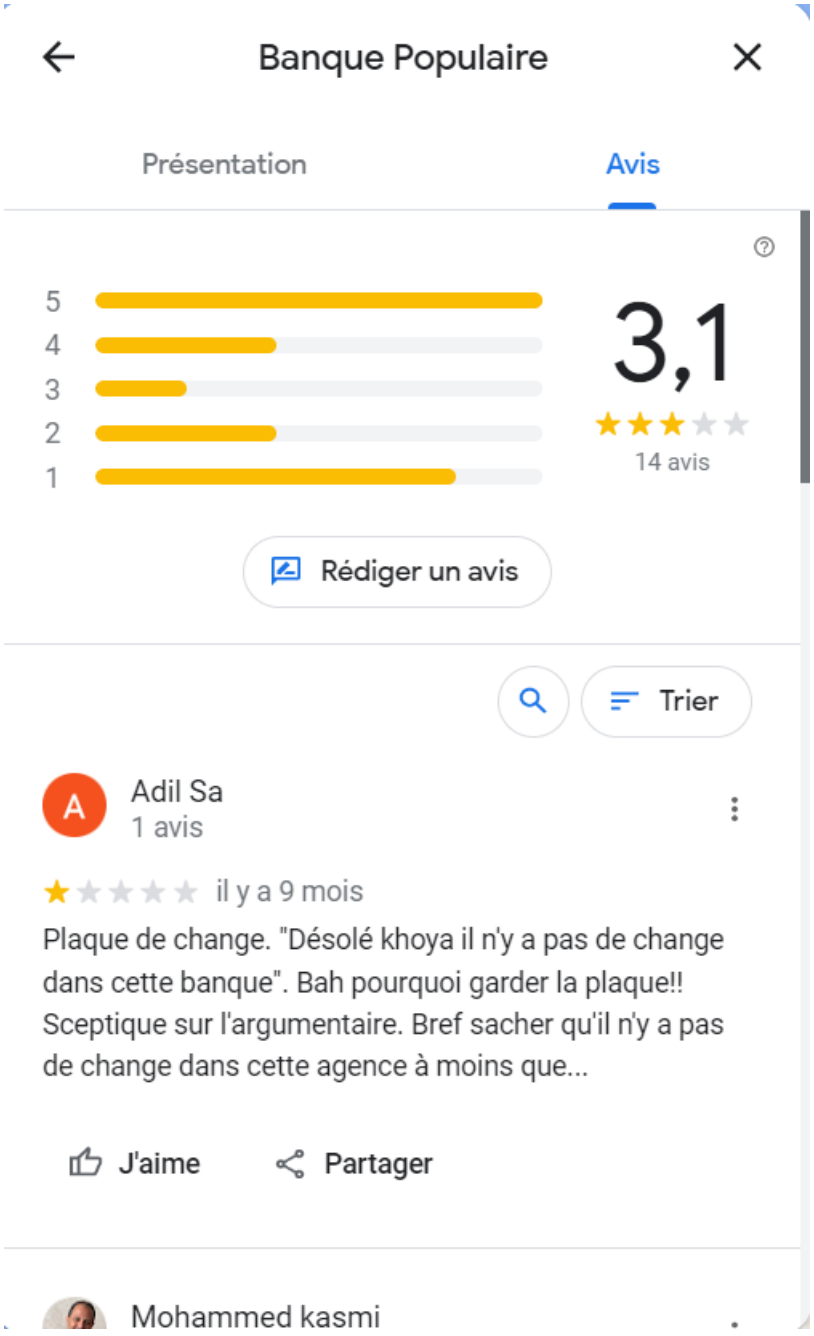
I. Collecte des données :

1. Type de données à scrapper :

On souhaite collecter les commentaires des clients des agences Banque Populaire au Maroc :



Ce qui nous intéresse particulièrement c’est le commentaires des clients, le nombre d’étoile attribué à chaque commentaire et la localisation de la banque.

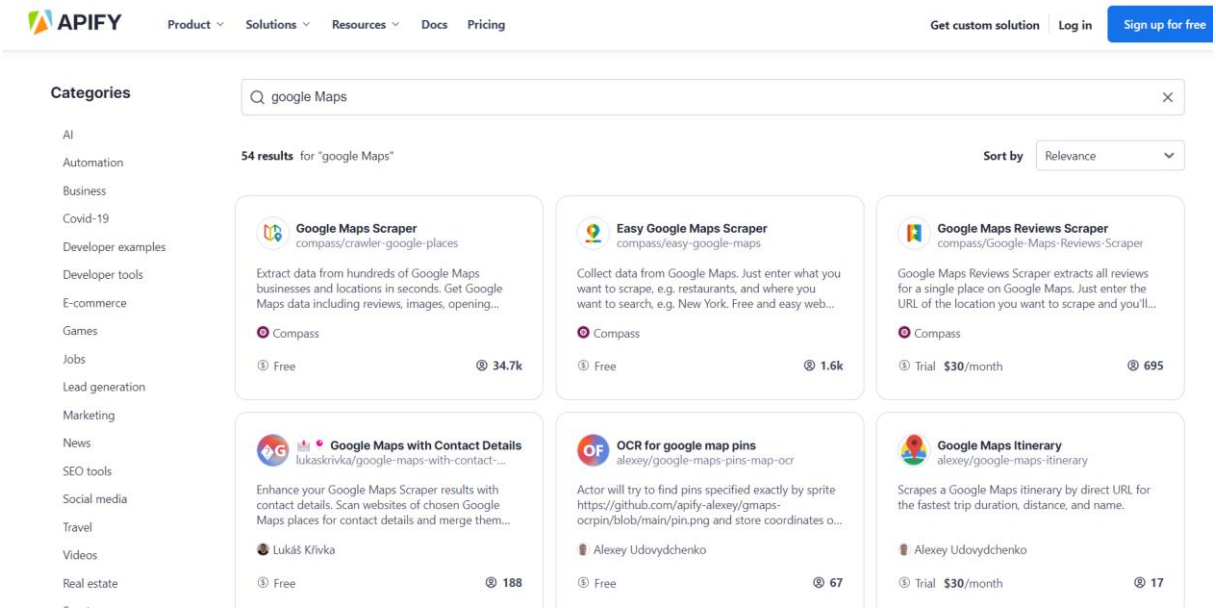


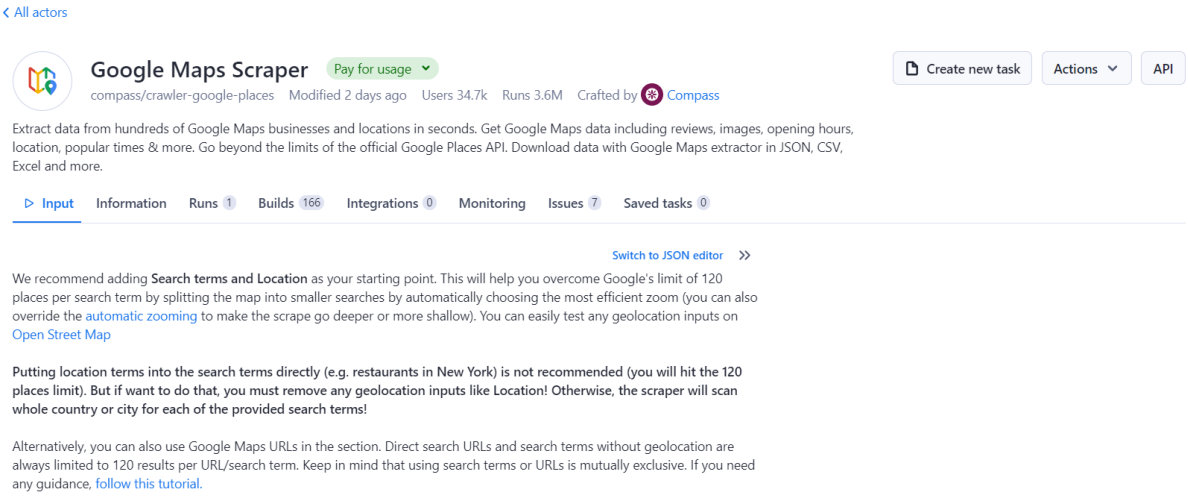
2. Scrappeur Apify :

Apify est un logiciel puissant qui simplifie le processus de grattage Web et d'extraction de données. Dans notre cas on va utiliser l'un des scrappeur Apify pour collecter les données dont on a besoin à partir de Google Maps.

a. Choix du scrappeur :

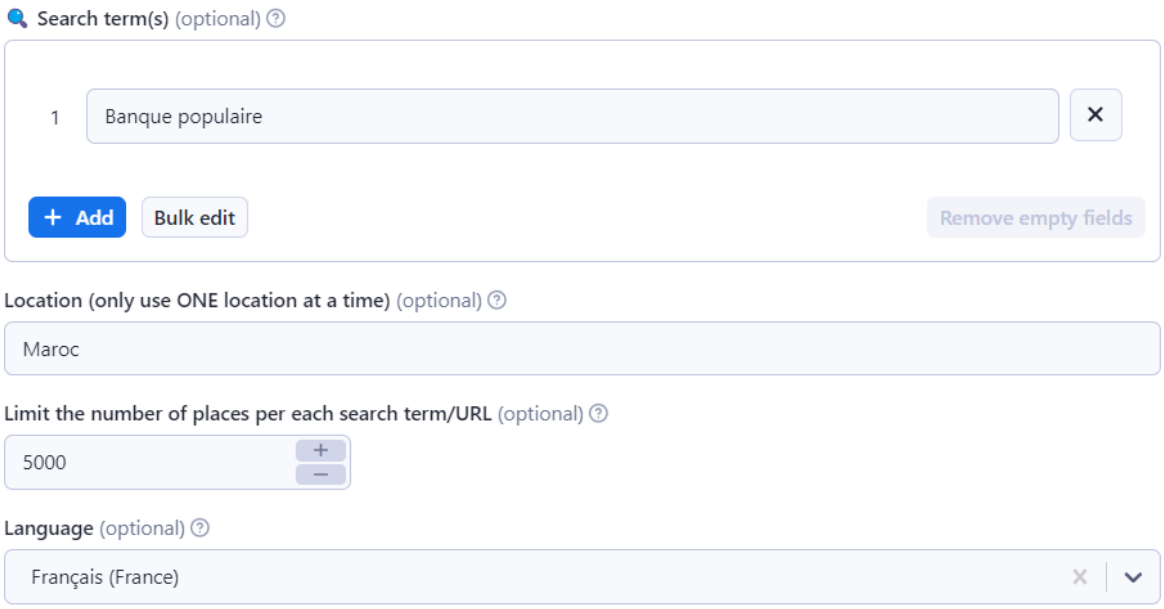
Apify nous offre une multitude de choix, pour notre cas on va choisir un scrappeur simple qui va nous fournir les données dont on a besoin.

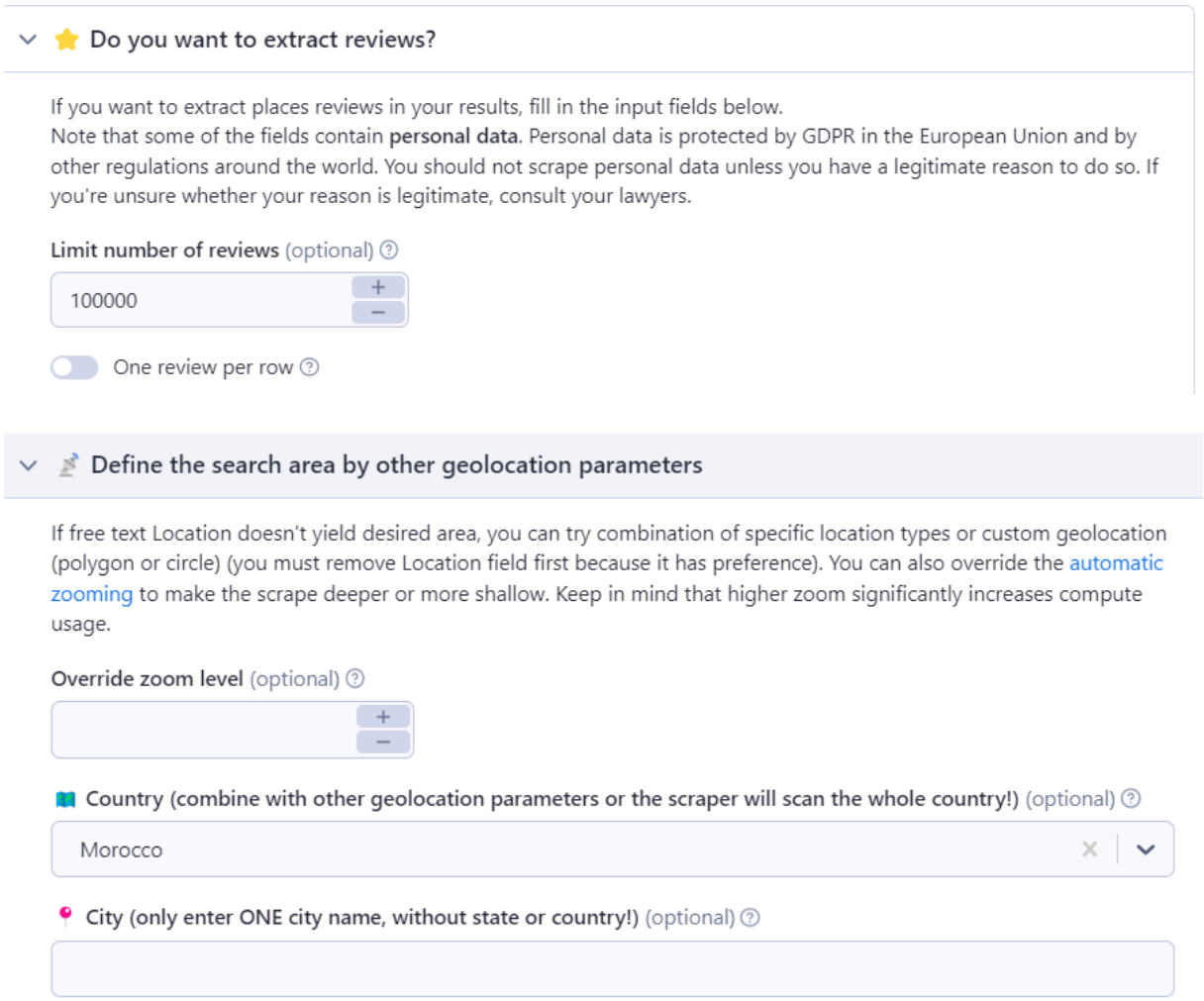




b. Programmation du scrappeur :

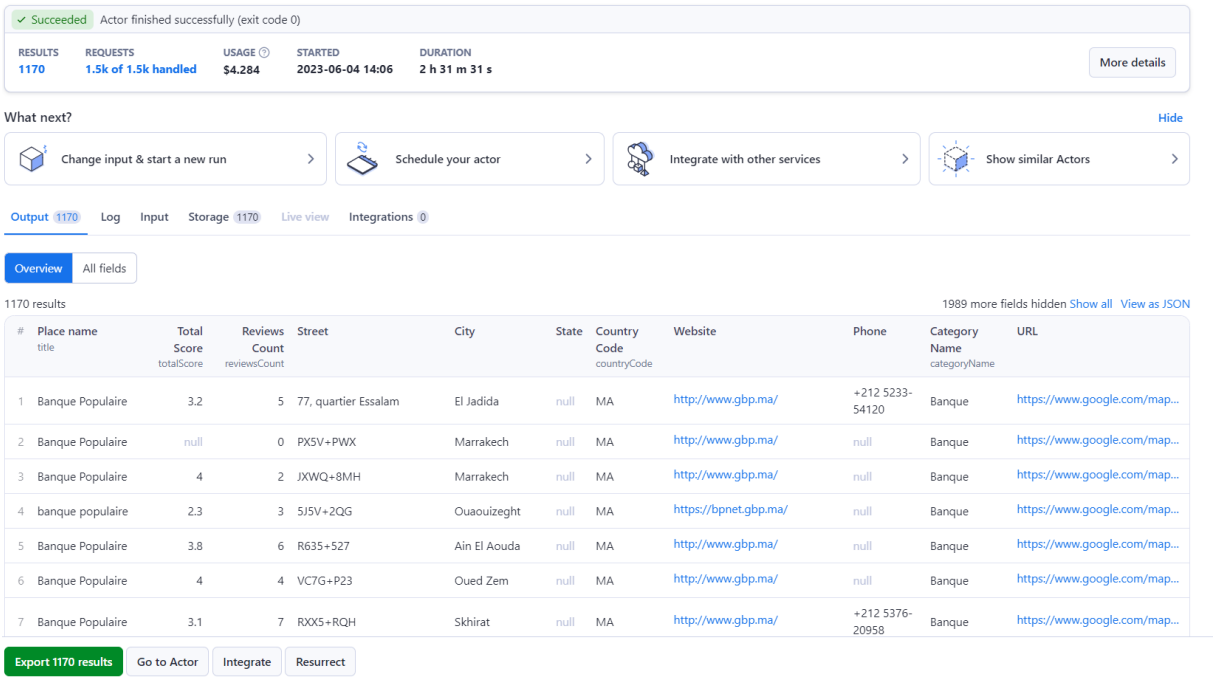
Une fois le scrappeur choisit, on doit le programmer afin qu'il cible uniquement les établissements qu'on souhaite analyser, dans notre cas c'est toutes les banques populaires du Maroc.

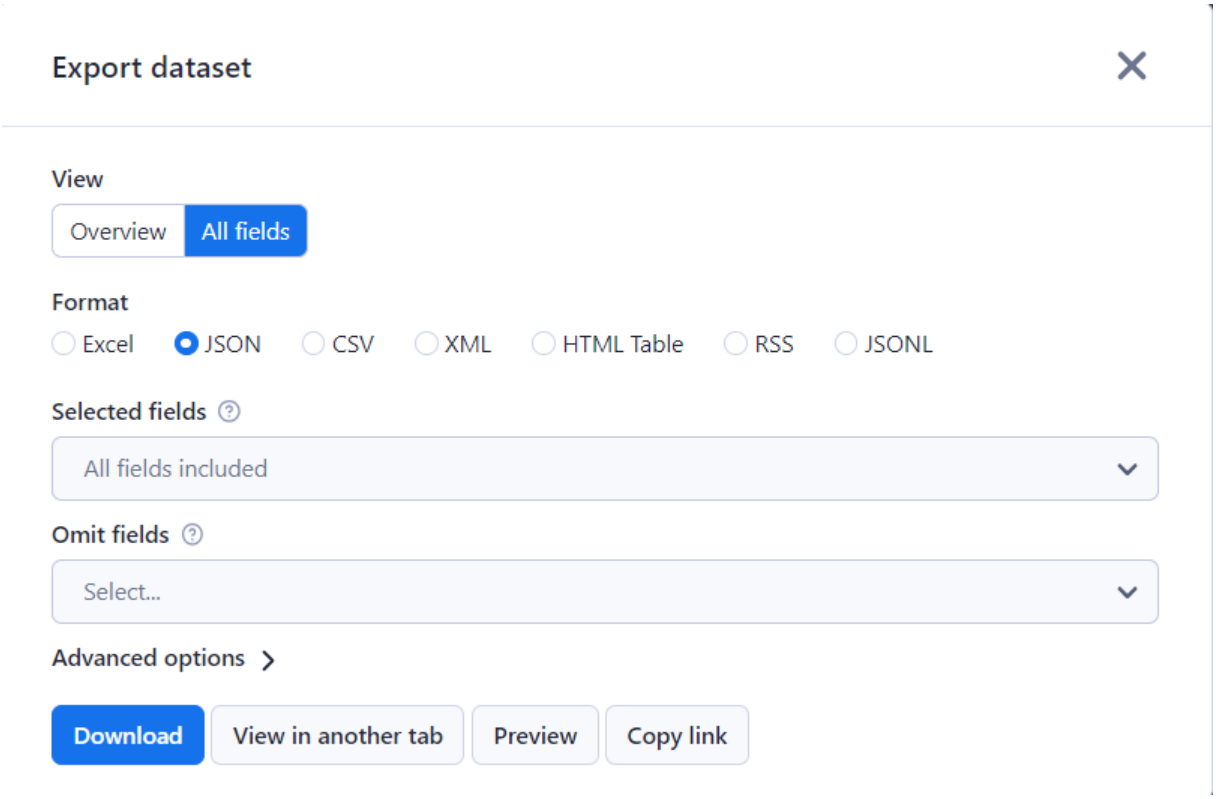




c. Résultats du scrapping :

Une fois terminer on peut exporter la data sous le format que l'on souhaite, on va l'exporter sous forme d'un fichier .csv :





Voici la forme de la base de données obtenue, elle est composée de 1170 lignes et 2000 colonnes.

	additionalInfo/Accessibilité/0/Ascenseur accessible en fauteuil roulant	additionalInfo/Accessibilité/0/Entrée accessible en fauteuil roulant	additionalInfo/Accessibilité/0/Parking accessible en fauteuil roulant	additionalInfo/Accessibilité/1/Entrée accessible en fauteuil roulant	additionalInfo/Accessibilité/1/Parking accessible en fauteuil roulant	additionalInfo/Offre/0/Avance d'argent	additionalInfo/Services/0/Convient aux enfants	[illegible] disponibles/0/Service de drive	address	categories/0	...	updatesFrom
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	77, quartier Essalam, El Jadida 24000, Maroc	Banque	...	
1	NaN	True	NaN	NaN	True	NaN	NaN	NaN	PX5V+PWX, Marrakech, Maroc	Banque	...	
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	JXWQ+8MH, Marrakech 40000, Maroc	Banque	...	
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5J5V+2QQ, Oued Zem, Maroc	Banque	...	
4	NaN	True	NaN	NaN	True	NaN	NaN	True	R635+527, Ain El Aouda, Maroc	Banque	...	
...	...	...	...	...	...	...	...	...	...	...	...	...
1165	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	53F7+667, Nador, Maroc	Banque	...	
1166	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2X6X+952, Av. Mohammed V, Al Aaroui, Maroc	Banque	...	
1167	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	113 Boulevard El Massira, Nador, Maroc	Banque	...	
1168	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	XJG4+7G3, Driouch, Maroc	Banque	...	
1169	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2GVV+V9H, Ben Taieb, Maroc	Banque	...	

1170 rows × 2000 columns

Cette base de données est incompréhensible à première vue. Mais après le traitement des données nous la rendrons plus claire.

II. Préparation de la data :

1. Simplification de la base de données :

Notre problème c'est qu'on a plein de colonnes indésirables, c'est une base de données brutes, on va donc nettoyer tout ça. Pour ce faire on va choisir les colonnes que l'on souhaite garder et on va les stocker dans une autre base de données :

```
# Liste des colonnes à conserver
columns_to_keep = ['reviews/0/text', 'reviews/0/stars', 'address', 'city','location/lat','location/lng']

# Suppression des colonnes indésirables
df = df[columns_to_keep]
```

Ensuite on va renommer les colonnes (c'est plus claire et plus simple ainsi) :

```python
# Dictionnaire des nouveaux noms de colonnes
nouvelles_colonnes = {
    'reviews/0/text': 'Commentaires',
    'reviews/0/stars': 'Nombre_étoiles',
    'address': 'Adresse',
    'city': 'Ville',
    'location/lat': 'Lattitude',
    'location/lng': 'Longetude'}

# Renommage des colonnes
df = df.rename(columns=nouvelles_colonnes)
```

Une fois toutes ces modifications effectuées, voici l'allure de notre nouvelle base de données :

	Commentaires	Nombre_étoiles	Adresse	Ville	Lattitude	Longetude
0	NaN	5.0	77, quartier Essalam, El Jadida 24000, Maroc	El Jadida	33.227927	-8.499593
1	NaN	NaN	PX5V+PWX, Marrakech, Maroc	Marrakech	31.709353	-8.005125
2	Le personnel est bien accueillant ! Et le ser...	5.0	JXWQ+8MH, Marrakech 40000, Maroc	Marrakech	31.645821	-8.010763
3	NaN	5.0	5J5V+2QG, Ouaouizeght, Maroc	Ouaouizeght	32.157562	-6.355555
4	NaN	3.0	R635+527, Ain El Aouda, Maroc	Ain El Aouda	33.802923	-6.792466
...	...	...	...	...	...	...
1165	Buen banco	4.0	53F7+667, Nador, Maroc	Nador	35.173026	-2.936949
1166	NaN	NaN	2X6X+952, Av. Mohammed V, Al Aaroui, Maroc	Al-Aroui	35.010893	-3.002108
1167	Good	3.0	113 Boulevard El Massira, Nador, Maroc	Nador	35.152482	-2.928514
1168	NaN	3.0	XJG4+7G3, Driouch, Maroc	Driouch	34.975645	-3.393693

Elle est bien plus simple à visualiser et elle sera plus pratique pour les analyses et l'entrainement des modèles.

2. Pré-traitement de la data :

a. Valeurs nulle et Word Cloud :

Dans la préparation des données on va tout d'abord commencer par supprimer les valeurs dupliquées :

```python
df.duplicated().sum()
df.drop_duplicates(inplace=True)
```

Ensuite on va s'occuper des lignes qui ne contiennent pas de commentaire :

```
df.isnull().sum()
```

```
Commentaires      534
Nombre_étoiles    109
Adresse             0
Ville               3
Lattitude           0
Longetude           0
dtype: int64
```

```
df.isnull().sum()
# Suppression des valeurs nulles
df = df.dropna()
df
df.isnull().sum()
```

```
Commentaires      0
Nombre_étoiles    0
Adresse           0
Ville             0
Lattitude         0
Longetude         0
dtype: int64
```

Maintenant on doit supprimer tous les commentaires qui ne sont pas en langue française :

```
from langdetect import detect

colonne_commentaires = 'Commentaires'

# Fonction pour détecter la langue
def detecter_langue(texte):
    try:
        langue = detect(texte)
        return langue == 'fr'
    except:
        return False

# Suppression des textes non français
df = df[df[colonne_commentaires].apply(detecter_langue)]
df
```

A la fin il ne nous reste que 317 lignes dans notre base de données :

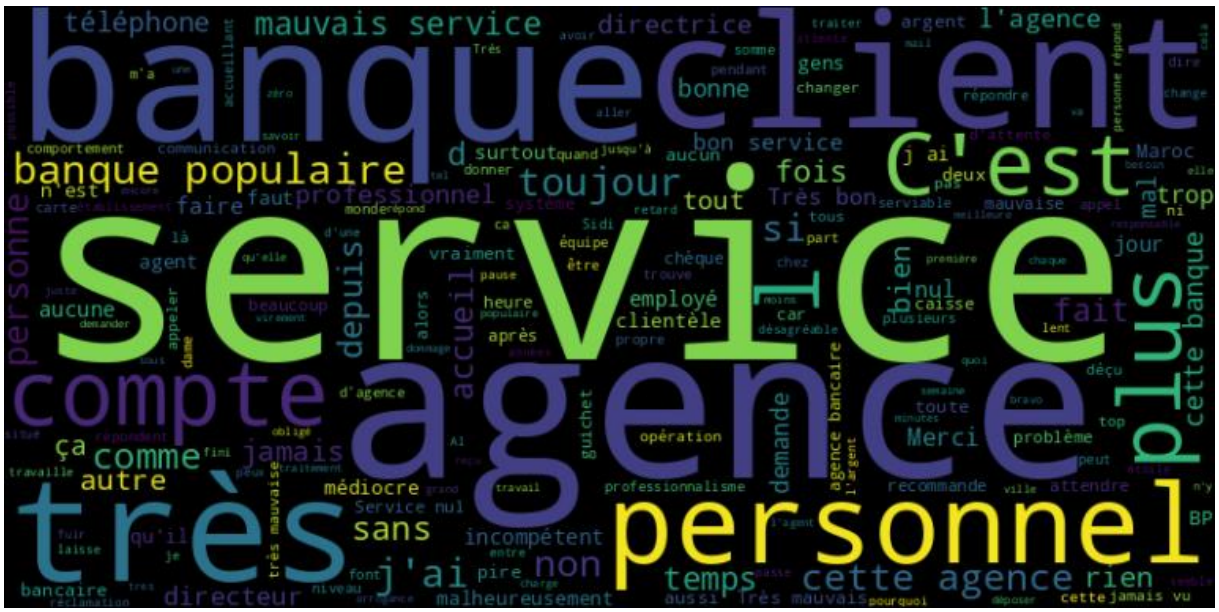
	Commentaires	Nombre_étoiles	Adresse	Ville	Lattitude	Longetude
2	Le personnel est bien accueillant ! Et le ser...	5.0	JXWQ+8MH, Marrakech 40000, Maroc	Marrakech	31.645821	-8.010763
6	Agence effroyable , personnel brute , service ...	1.0	RXX5+RQH, Skhirat, Maroc	Skhirat	33.849555	-7.040506
12	Notre agence à l'honneur de vous accueillir du...	5.0	GR45+JC6، R308, El Borouj, Maroc	El Borouj	32.506526	-7.191484
20	service médiocre aucun respect pour les clients	1.0	H835+3J4, Casablanca, Maroc	Casablanca	33.552647	-7.690906
22	C'était une agence impeccable en terme de rapi...	3.0	G8V8+28G, Casablanca, Maroc	Casablanca	33.542563	-7.684181
...	...	...	...	...	...	...
1154	Je lui donne 1 juste pour me permettre d ajout...	1.0	W494+W4J, Av. Ibn Khaldoun, Témara, Maroc	Témara	33.919835	-6.894732
1156	Une des pires succursales (toutes banques conf...	1.0	25, avenue Al Massira (C.y.m), Amal 4, Yacoub ...	Yacoub El Mansour	33.982400	-6.880680
1159	Personnel en sous effectif , numero de telepho...	1.0	X5MF+53H, Av/bd Al Haouz, El Youssoufia 10190,...	El Youssoufia	33.982932	-6.827269
1161	Personnels aimables, bonne prestation !!	4.0	JGMV+CCV, Âïn-Harrouda, Maroc	Âïn-Harrouda	33.633607	-7.456465
1164	mauvaise service	1.0	JG97+PH6, Bd Allait Ibn Asaad, Casablanca 2025...	Casablanca	33.619286	-7.486109

317 rows × 6 columns

Avant de tokeniser les textes on va d'abord afficher un word cloudet supprimer les mots vides :

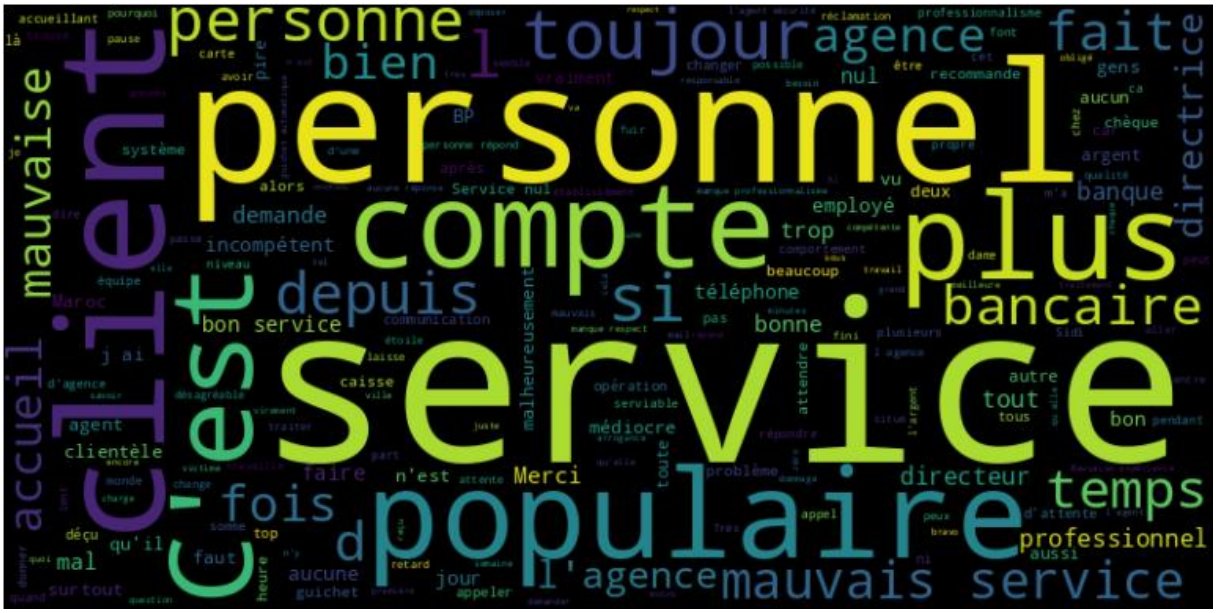
```
import nltk
from nltk.corpus import stopwords

# Téléchargement des mots vides (stop words) pour le français
nltk.download('stopwords')

# Nom de la colonne des commentaires
colonne_commentaires = 'Commentaires'

# Liste des mots vides pour le français
mots_vides = set(stopwords.words('french'))

# Fonction pour supprimer les mots vides des commentaires
def supprimer_mots_vides(texte):
    mots = texte.split()
    mots_filtres = [mot for mot in mots if mot.lower() not in mots_vides]
    return ' '.join(mots_filtres)

# Suppression des mots vides dans les commentaires
df[colonne_commentaires] = df[colonne_commentaires].apply(supprimer_mots_vides)
```

Et voici notre word cloud :



Apparemment certains mots classiques qui ne servent qu'à designer la banque comme ''banque'' ou ''agence'' sont très présents dans les textes nous devront les supprimer également pour avoir une meilleure clarté sur les mots qui désignent des sentiments.

```
# Liste de mots vides personnalisée
mots_vides_personnalises = ['banque','agence', 'client', 'très', 'comme', 'ça', 'autre', 'jamais', 'rien', 'non', 'jusqu\'à', 'sans', 'j\'ai', 'cette']
```



C'est déjà mieux comme ça, on commence à apercevoir des mots qui reflètent des sentiments.

b. Tokenisation et lemmatisation :

Maintenant qu'on a bien préparer les textes on peut les tokeniser :

```
# Tokenisation des commentaires
df[colonne_commentaires] = df[colonne_commentaires].apply(tokenizer)
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

	Commentaires	Nombre_étoiles	Adresse	Ville	Lattitude	Longetude
2	[personnel, bien, accueillant, !, service, bon]	5.0	JXWQ+8MH, Marrakech 40000, Maroc	Marrakech	31.645821	-8.010763
6	[effroyable, „ personnel, brute, „ service, ...	1.0	RXX5+RQH, Skhirat, Maroc	Skhirat	33.849555	-7.040506
12	[l'honneur, accueillir, lundi, >, vendredi, 8h...	5.0	GR45+JC6· R308, El Borouj, Maroc	El Borouj	32.506526	-7.191484
20	[service, médiocre, aucun, respect, clients]	1.0	H835+3J4, Casablanca, Maroc	Casablanca	33.552647	-7.690906
22	[C'était, impeccable, terme, rapidité, service...	3.0	G8V8+28G, Casablanca, Maroc	Casablanca	33.542563	-7.684181
...	...	...	...	...	...	...
1154	[donne, 1, juste, permettre, ajouter, commenta...	1.0	W494+W4J, Av. Ibn Khaldoun, Témara, Maroc	Témara	33.919835	-6.894732
1156	[pires, succursales, (, toutes, banques, confo...	1.0	25, avenue Al Massira (C.y.m), Amal 4, Yacoub ...	Yacoub El Mansour	33.982400	-6.880680
1159	[Personnel, sous, effectif, „ numero, telepho...	1.0	X5MF+53H, Av/bd Al Haouz, El Youssoufia 10190,...	El Youssoufia	33.982932	-6.827269
1161	[Personnels, aimables, „ bonne, prestation, !...	4.0	JGMV+CCV, Âïn-Harrouda, Maroc	Âïn-Harrouda	33.633607	-7.456465
1164	[mauvaise, service]	1.0	JG97+PH6, Bd Allait Ibn Asaad, Casablanca 2025...	Casablanca	33.619286	-7.486109

317 rows × 6 columns

Juste après on va les lemmatiser :

```
from nltk.stem import WordNetLemmatizer

# Téléchargement des ressources nécessaires pour la lemmatisation

nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')

# Fonction de lemmatisation
def lemmatiser(mots):
    lemmatizer = WordNetLemmatizer()
    mots_lemmatizes = [lemmatizer.lemmatize(mot) for mot in mots]
    return ' '.join(mots_lemmatizes)

# Lemmatisation des commentaires
df[colonne_commentaires] = df[colonne_commentaires].apply(lambda x: lemmatiser(x))
df
```

c. Vectorisation des textes :

L'une des dernières étapes avant d'entrainer un modèle d'apprentissage est de vectoriser le texte en vecteurs numériques, on va utiliser une vectorisation TF-IDF car les mots rares on une grande importance dans les commentaires et ils reflètent des sentiments :

```
import spacy
from sklearn.feature_extraction.text import TfidfVectorizer

# Création de l'objet TfidfVectorizer
vectorizer = TfidfVectorizer()

# Vectorisation des commentaires
X = vectorizer.fit_transform(df[colonne_commentaires])

# Récupération des noms des fonctionnalités (feature names)
feature_names = vectorizer.get_feature_names_out()

# Conversion en DataFrame
df_vectorise = pd.DataFrame(X.toarray(), columns=feature_names)
```

d. Encodage de la variable nombre d'étoile :

La dernière étape avant l'entrainement du modèle d'apprentissage est d'encoder la colonne Nombre_étoiles en une variable catégoriale binaire. Cette variable et notre point de jugement, elle permettra à notre modèle d'attribuer une étiquette positive ou négative aux commentaires durant l'apprentissage :

```
# Nom de la colonne "nombre_étoiles"
colonne_etoiles = 'Nombre_étoiles'

# Conversion en variable catégorielle binaire
df[colonne_etoiles] = df[colonne_etoiles].replace({1: 0, 2: 0, 3: 1, 4: 1, 5: 1})
```

Les valeurs inférieures à 3 sont transformées en 0, ils seront ainsi considérés comme des commentaires négatifs et le reste des valeurs sont transformées en 1. C'est des commentaires positifs.

	Commentaires	Nombre_étoiles	Adresse	Ville	Lattitude	Longetude	Sentiment
2	persnnelbeccuellntservceb	1.0	JXWQ+8MH, Marrakech 40000, Maroc	Marrakech	31.645821	-8.010763	positif
6	effrblepersnnelbruteser...	0.0	RXX5+RQH, Skhirat, Maroc	Skhirat	33.849555	-7.040506	négatif
12	lheurenneurccuellrlundven...	1.0	GR45+JC6· R308, El Borouj, Maroc	El Borouj	32.506526	-7.191484	positif
20	servcemédcreucunrespect...	0.0	H835+3J4, Casablanca, Maroc	Casablanca	33.552647	-7.690906	négatif
22	céttmpeccbletermerpdters...	1.0	G8V8+28G, Casablanca, Maroc	Casablanca	33.542563	-7.684181	positif
...	...	...	...	...	...	...	...
1154	dnnejustepermettrejuter...	0.0	W494+W4J, Av. Ibn Khaldoun, Témara, Maroc	Témara	33.919835	-6.894732	négatif
1156	pressuccurslestutesbnqu...	0.0	25, avenue Al Massira (C.y.m), Amal 4, Yacoub ...	Yacoub El Mansour	33.982400	-6.880680	négatif
1159	persnnelsueffectfnumert...	0.0	X5MF+53H, Av/bd Al Haouz, El Youssoufia 10190,...	El Youssoufia	33.982932	-6.827269	négatif
1161	persnnelsmblesbnnepresttn	1.0	JGMV+CCV, Âïn-Harrouda, Maroc	Âïn-Harrouda	33.633607	-7.456465	positif
1164	muvseservce	0.0	JG97+PH6, Bd Allait Ibn Asaad, Casablanca 2025...	Casablanca	33.619286	-7.486109	négatif

317 rows × 7 columns

La variable est apparemment de type float, on va donc la transformer en entier :

```
df['Nombre_étoiles'] = df['Nombre_étoiles'].astype(int).replace({1.0: 1, 0.0: 0})
```

Commentaires	Nombre_étoiles	Adresse	Ville	Lattitude	Longetude
Le personnel est bien accueillant ! Et le ser...	1	JXWQ+8MH, Marrakech 40000, Maroc	Marrakech	31.645821	-8.010763
Agence effroyable , personnel brute , service ...	0	RXX5+RQH, Skhirat, Maroc	Skhirat	33.849555	-7.040506
Notre agence à l'honneur de vous accueillir du...	1	GR45+JC6· R308, El Borouj, Maroc	El Borouj	32.506526	-7.191484
service médiocre aucun respect pour les clients	0	H835+3J4, Casablanca, Maroc	Casablanca	33.552647	-7.690906
C'était une agence impeccable en terme de rapi...	1	G8V8+28G, Casablanca, Maroc	Casablanca	33.542563	-7.684181
...	...	...	...	...	...
Je lui donne 1 juste pour me permettre d ajout...	0	W494+W4J, Av. Ibn Khaldoun, Témara, Maroc	Témara	33.919835	-6.894732
Une des pires succursales (toutes banques conf...	0	25, avenue Al Massira (C.y.m), Amal 4, Yacoub ...	Yacoub El Mansour	33.982400	-6.880680
Personnel en sous effectif , numero de telepho...	0	X5MF+53H, Av/bd Al Haouz, El Youssoufia 10190,...	El Youssoufia	33.982932	-6.827269
Personnels aimables, bonne prestation !!	1	JGMV+CCV, Âïn-Harrouda, Maroc	Âïn-Harrouda	33.633607	-7.456465
mauvaise service	0	JG97+PH6, Bd Allait Ibn Asaad, Casablanca 2025...	Casablanca	33.619286	-7.486109

Le pré-traitement des données s'achève ainsi, notre base de données est prête à être utilisée.

III. Entrainement du modèle :

1. Choix et entrainement du modèle :

On va choisir un modèle pré-entrainer pour détecter les sentiments dans les commentaires.

Le modèle BERT (Bidirectional Encoder Representations from Transformers) est un modèle de langage pré-entraîné qui a révolutionné le domaine du traitement automatique du langage

naturel (NLP). BERT est basé sur une architecture de transformer bidirectionnelle qui permet une compréhension contextuelle approfondie des mots et des phrases.

Il y a plusieurs raisons pour lesquelles on peut choisir BERT pour une tâche de classification de sentiments :

- Compréhension contextuelle : BERT prend en compte le contexte global d'un texte grâce à son architecture bidirectionnelle. Cela lui permet de capturer les nuances et les dépendances contextuelles entre les mots, ce qui peut améliorer considérablement la performance de la classification de sentiments.
- Représentations de mots riches : BERT est pré-entraîné sur de grandes quantités de données textuelles, ce qui lui permet d'apprendre des représentations de mots riches et informatives. Ces représentations peuvent aider le modèle à capturer les caractéristiques pertinentes des mots dans les commentaires lors de la classification des sentiments.
- Prise en charge du multilinguisme : BERT est disponible dans de nombreuses langues, y compris le français. Cela permet de bénéficier des avantages de BERT pour la classification des sentiments dans des textes en français.

Cependant on doit fine-tuniner des données, c’est essentiel pour adapter BERT à la tâche spécifique de classification des sentiments. Le modèle BERT est pré-entraîné sur une tâche de prédiction de mots manquants et il a une compréhension générale du langage. Cependant, le fine-tuning permet d'ajuster les paramètres du modèle pour qu'il se spécialise dans la tâche spécifique de classification des sentiments. En fine-tunant les données, le modèle peut apprendre à associer les caractéristiques des commentaires aux étiquettes de sentiment (positif ou négatif) et améliorer ainsi les performances de classification.

```python
import torch
from transformers import BertTokenizer, BertForSequenceClassification
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Diviser les données en ensembles d'entraînement et de test
X_train, X_test, y_train, y_test = train_test_split(df['Commentaires'], df['Nombre_étoiles'], test_size=0.

import torch
import torch.nn as nn
from transformers import BertTokenizer, BertForSequenceClassification
from torch.utils.data import DataLoader, TensorDataset

# Charger les données d'entraînement et de validation, et les convertir en tenseurs
train_texts = [X_train]  # Liste des textes d'entraînement
train_labels = [y_train]  # Liste des étiquettes (1 ou 0) correspondant aux textes d'entraînement

val_texts = [X_test]  # Liste des textes de validation
val_labels = [y_test]  # Liste des étiquettes (1 ou 0) correspondant aux textes de validation

tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-cased')

train_encodings = tokenizer(train_texts, truncation=True, padding=True)
val_encodings = tokenizer(val_texts, truncation=True, padding=True)

train_dataset = TensorDataset(torch.tensor(train_encodings['input_ids']),
                              torch.tensor(train_encodings['attention_mask']),
                              torch.tensor(train_labels))

val_dataset = TensorDataset(torch.tensor(val_encodings['input_ids']),
                            torch.tensor(val_encodings['attention_mask']),
                            torch.tensor(val_labels))

# Charger le modèle pré-entraîné
model = BertForSequenceClassification.from_pretrained('bert-base-multilingual-cased', num_labels=2)

# Paramètres d'entraînement
batch_size = 16
learning_rate = 2e-5
num_epochs = 5
```

```python
# Définir l'optimiseur et la fonction de perte
optimizer = torch.optim.AdamW(model.parameters(), lr=learning_rate)
loss_fn = nn.CrossEntropyLoss()

# Créer les dataloaders pour l'entraînement et la validation
train_dataloader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
val_dataloader = DataLoader(val_dataset, batch_size=batch_size)

# Entraînement du modèle
for epoch in range(num_epochs):
    model.train()
    total_loss = 0

    for batch in train_dataloader:
        optimizer.zero_grad()
        input_ids, attention_mask, labels = batch
        outputs = model(input_ids, attention_mask=attention_mask, labels=labels)
        loss = outputs.loss
        total_loss += loss.item()
        loss.backward()
        optimizer.step()

    avg_train_loss = total_loss / len(train_dataloader)

    model.eval()
    total_val_loss = 0
    total_val_correct = 0

    for batch in val_dataloader:
        input_ids, attention_mask, labels = batch
        with torch.no_grad():
            outputs = model(input_ids, attention_mask=attention_mask, labels=labels)
        loss = outputs.loss
        logits = outputs.logits
        total_val_loss += loss.item()
        _, predicted_labels = torch.max(logits, dim=1)
        total_val_correct += (predicted_labels == labels).sum().item()

    avg_val_loss = total_val_loss / len(val_dataloader)
    val_accuracy = total_val_correct / len(val_dataset)

    print(f"Epoch {epoch+1}/{num_epochs} - Train Loss: {avg_train_loss:.4f} - Val Loss: {avg_val_loss:.4f} - Val Accuracy: {val_accuracy:.4f}")
```

2. Prédiction et résultats sur l'ensemble de test :

Dans le code précèdent on avait déjà programmé les données de test et voici les résultats :

```
Epoch 5/10 - Train Loss: 0.2352 - Val Loss: 0.2874 - Val Accuracy: 0.8923
Epoch 10/10 - Train Loss: 0.2017 - Val Loss: 0.2498 - Val Accuracy: 0.9132
```

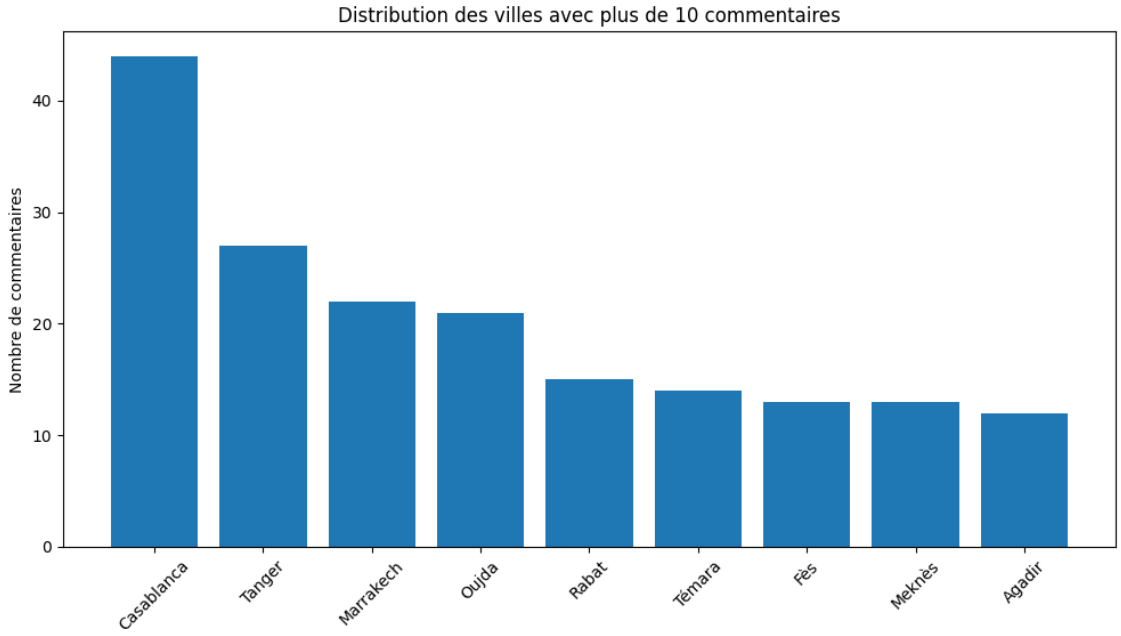
- Epoch 5/10 : indique qu'il s'agit de la 5e epoch sur un total de 10 epochs prévues pour l'entraînement du modèle.
- Train Loss = 0.2352 représente la perte (loss) moyenne sur les données d'entraînement à cette epoch. Une valeur de 0.2352 suggère une faible perte, ce qui indique une bonne adéquation (fit) du modèle aux données d'entraînement.
- Val Loss = 0.2874 est la perte moyenne sur les données de validation à cette epoch. Une valeur de 0.2874 indique une perte légèrement plus élevée que la perte d'entraînement, mais encore relativement faible, ce qui suggère une bonne capacité du modèle à généraliser sur de nouvelles données.

- Val Accuracy = 0.8923 représente la précision (accuracy) du modèle sur les données de validation à cette epoch. Une valeur de 0.8923 indique que le modèle a correctement classé environ 89,23 % des exemples de la validation, ce qui est considéré comme un bon niveau de précision.

IV. Visualisation des données :

1. Les villes les plus impliquées :

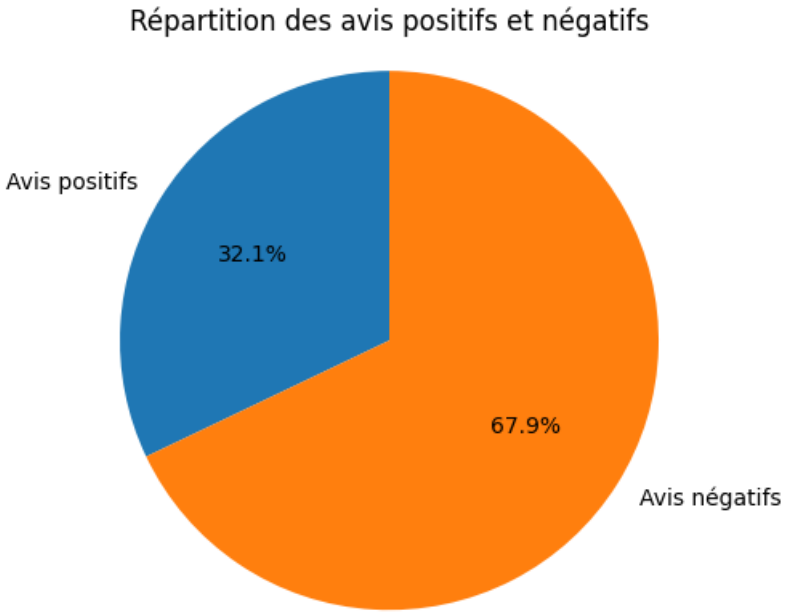
On va commencer par voir dans quel villes les citoyens interagissent le plus et donnent leurs avis sur les banques :



Dans le reste des villes ont a moins de 10 commentaires sur leurs agences.

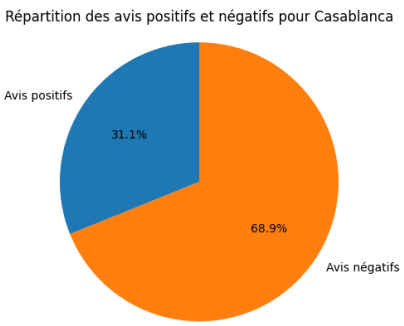
2. Taux de satisfaction des clients :

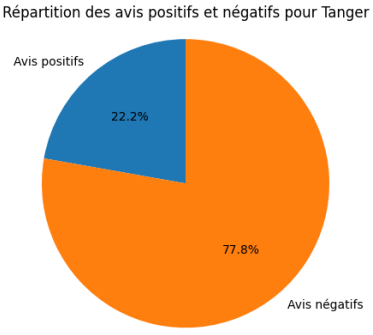
On va voir à quel point les clients sont satisfaits des agences de la Bp dans le royaume :

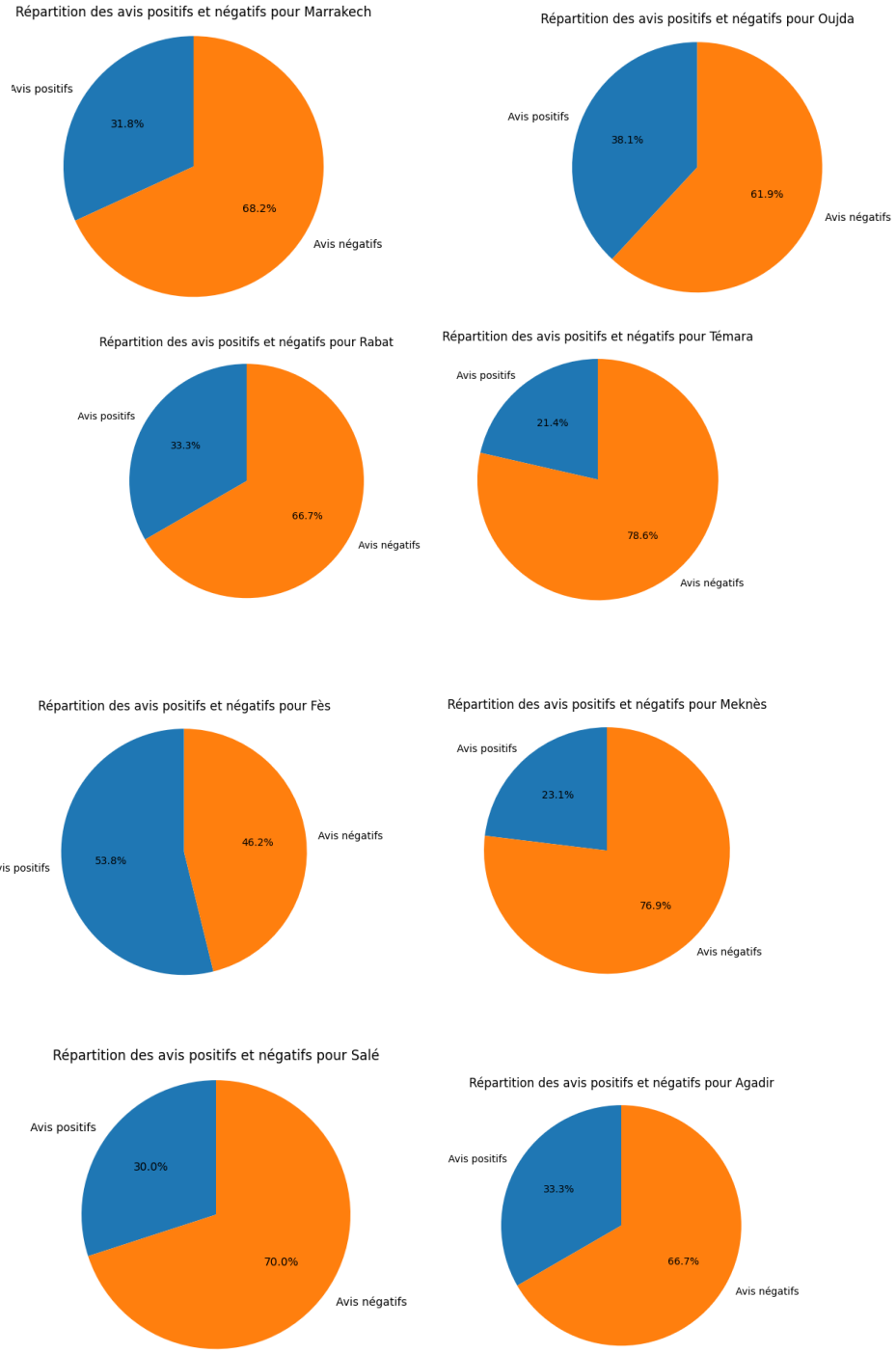


Il y'a une dominance des avis négatifs. Plus de 60% des clients ne sont pas satisfaits du service des agences Bp au Maroc.

On va voir plus en détails le taux de satisfaction des clients en l'analysant par rapport à chaque ville (on va se limiter aux 10 villes qui ont le plus de commentaires) :



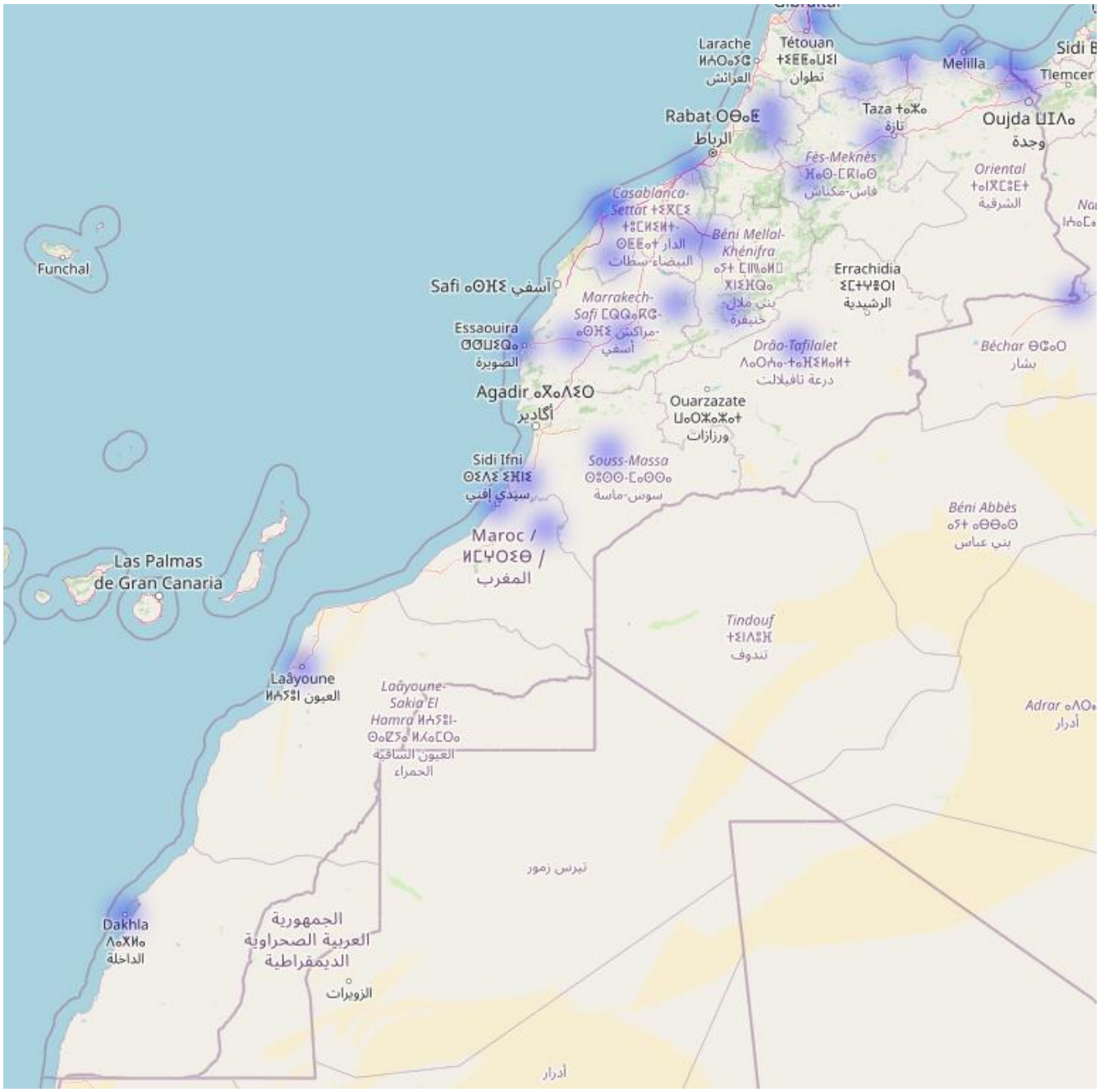




3. Heat-map :

Pour mieux illustrer tous ces graphes on va créer une heat-map du Maroc qui affiche le taux de colère des clients par région :

Les zones en bleu représentent les zones ou les clients sont en colères, plus le bleu est intense plus les clients sont en colère dans cette zone.


Larache
العرائش
Tétouan
تطوان
Melilla
Sidi B
Tlemcer
Rabat
الرباط
Taza
تازة
Oujda
وجدة
Fès-Meknès
فاس-مكناس
Oriental
الشرقية
Casablanca-Settat
الدار البيضاء-سطات
Béni Mellal-Khénifra
بني ملال خنيفرة
Errachidia
الرشيدية
Funchal
Safi
آسفي
Marrakech-Safi
مراكش-آسفي
Essaouira
الصويرة
Drâa-Tafilalet
درعة تافيلالت
Béchar
بشار
Agadir
أگادير
Ouarzazate
ورزازات
Sidi Ifni
سيدي إفني
Souss-Massa
سوس-ماسة
Béni Abbès
بني عباس
Maroc / المغرب
Las Palmas de Gran Canaria
Tindouf
تندوف
Laâyoune
العيون
Laâyoune-Sakia El Hamra
العيون الساقية الحمراء
Adrar
أدرار
تيرس زمور
Dakhla
الداخلة
الجمهورية العربية الصحراوية الديمقراطية
الزويرات
أدرار

Conclusion :

Ce projet a permis d'explorer et d'analyser les commentaires des clients sur les agences de la Banque Populaire du Maroc. En utilisant des techniques de web scraping et de nettoyage de données, on a pu extraire et préparer les données pour l'analyse ultérieure. L'entraînement du modèle basé sur BERT a permis de détecter les sentiments des commentaires, ce qui peut être précieux pour comprendre la satisfaction des clients et identifier les problèmes potentiels. Enfin, la visualisation des données a permis de présenter les résultats de manière claire et concise, offrant ainsi une meilleure compréhension des tendances et des insights.

Ce projet démontre l'importance de l'analyse des commentaires clients dans le domaine bancaire et met en évidence les avantages de l'utilisation de techniques avancées telles que le web scraping, le nettoyage de données et l'apprentissage automatique pour extraire des informations précieuses à partir de grandes quantités de données non structurées. Ces informations peuvent aider les entreprises à améliorer leurs services, à mieux comprendre les besoins des clients et à prendre des décisions éclairées pour leur satisfaction.