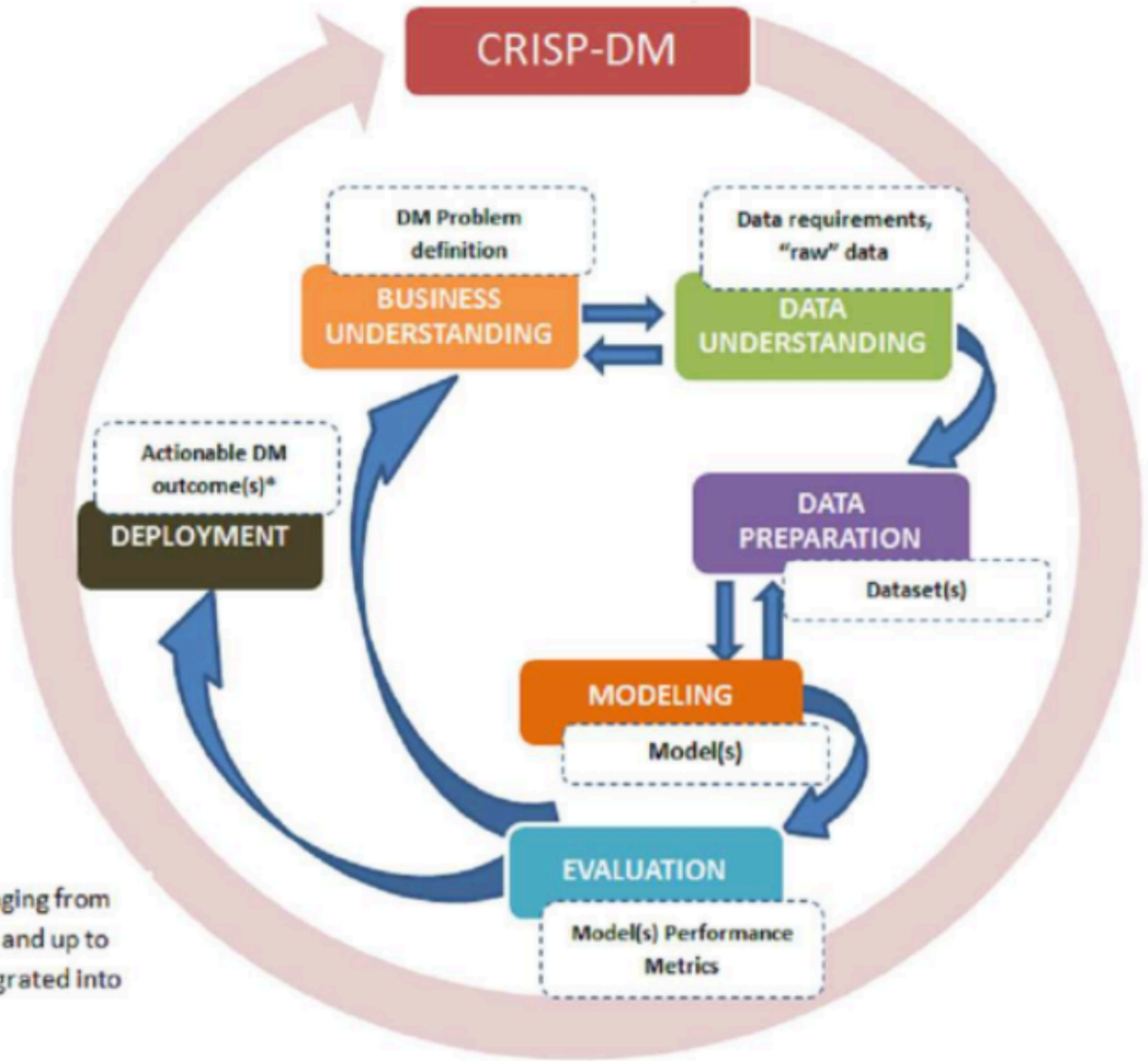
Customers Reviews on Businus Hotel, Tunis Analysis

Presented By:

Ben Amor Hanine
Cheriaa Nermine
Logtari Mariem

Décortiquons cet analyse en se basant sur le modèle CRISP



* various complexity levels ranging from one-time or repeatable report and up to complex DM components integrated into enterprise IS systems

Compréhension du metier

Contexte

- Entité: Le "Business Hotel" (4*, Tunis – Montplaisir), ciblant une clientèle d'affaires.
- Enjeu: Dans le marché concurrentiel de l'hôtellerie en tuninse, l’hotel doir exploiter l’analyse des retours clients en ligne pour évaluer sa réputation

Problème Business

- Comprendre la répartition globale des sentiments (Positif vs Négatif) pour prioriser les investissements.

Objectifs et critères de succès :

- Approche: Appliquer CRISP-DM pour Extraire les données (craping) des plateformes d'avis (Booking, Google Maps). et Analyser les sentiments (Positif, Négatif, Neutre).
- Résultat Atendu: Un modèle de classification fiable (F1-score acceptable) permettant de mesurer la satisfaction globale des clients de l’hotel, les points forts et faibles .

Aquisition et compréhen — sion des Donées

1. Stratégie de Collecte (Web Scraping)

2. Description du Dataset Final

Web Scraping

→ Choix des sources:

Sélection principale de Google Maps (208 avis identifiés)
car cette plateforme offre le volume d'avis le plus pertinent contrairement à Momondo (données agrégées) ou Booking.TripAdvisor (accès restreint).

→ Défis & Solutions :

Les sites utilisant du contenu dynamique (JavaScript) ont limité l'efficacité de Beautiful Soup. L'approche a donc évolué vers l'utilisation de Selenium (l'extraction de JSON cachés) pour garantir une collecte complète.

D’ailleurs la non -disponibilité des données(restrictions ou sécurisations) n’a pas affecté la variété de nos données car les avis dans Google Maps sont dérivés de plusieurs sites de booking ou d’avis.

Description du Dataset Final :

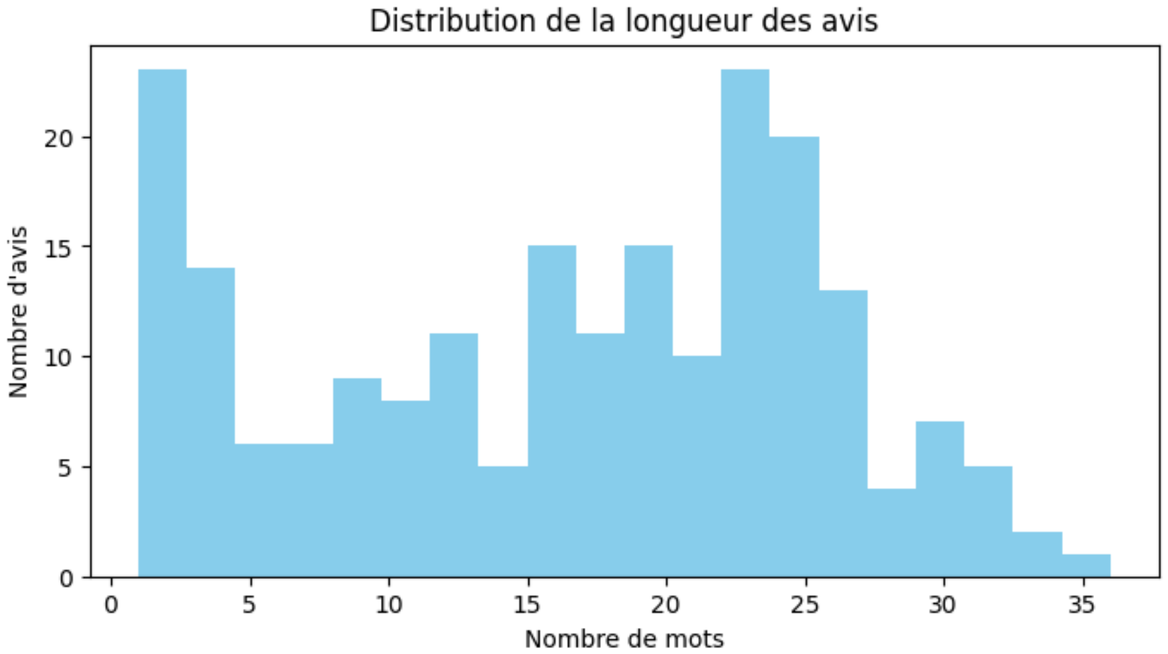
Volume: 201 Avis aprés nettoyage

Attributs collectés: nom client, Date, Texte aprés nettoyage

Observations:

- Le jeu de Données manque la note (le rating)==> on a des données non étiquetées. ce qui nous pousse à analyser les sentiments des textes pour pouvoir ajouter ce feature.

Exploration du Texte (EDA)



La distribution est fortement asymétrique à gauche.

- **La grande majorité des clients laissent des avis très courts (< 30 mots).**
- **Une minorité prend le temps de rédiger des paragraphes détaillé**
-

Cette disparité de longueur confirme la nécessité d'une normalisation (TF-IDF) lors de l'étape suivante,

Préparation des Données (Data Preparation)

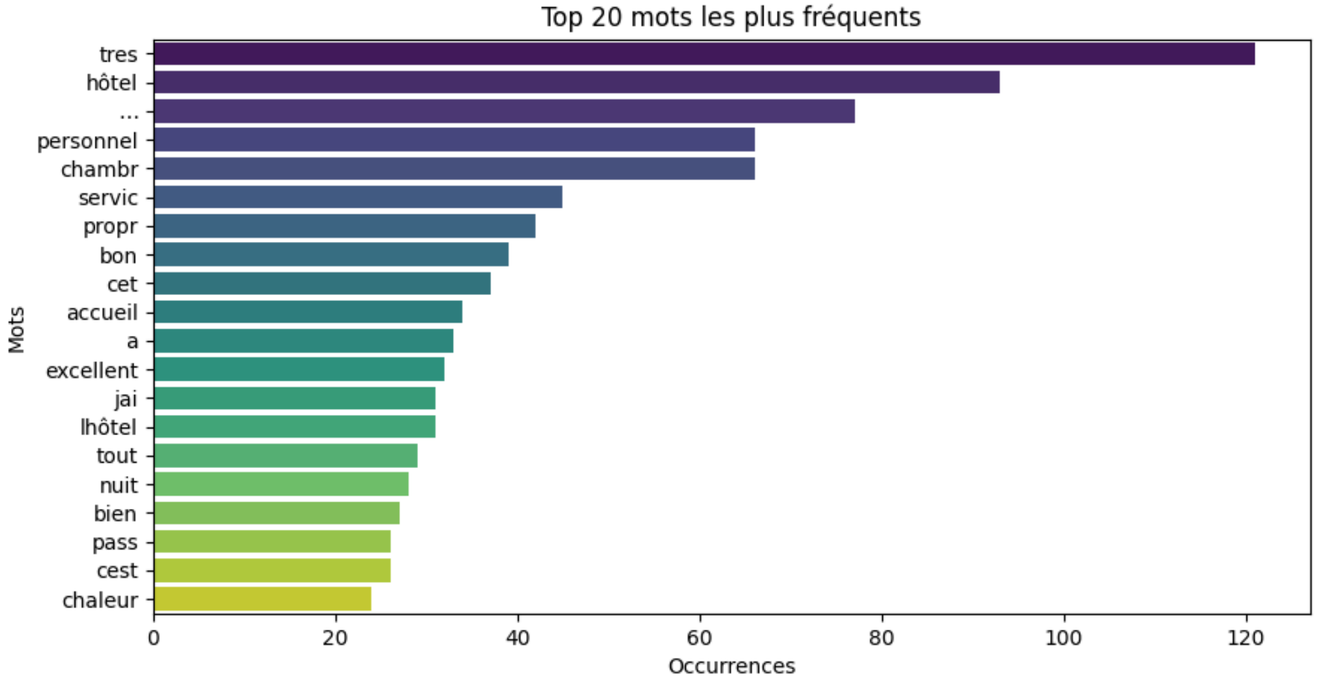
1. **Pipeline de Nettoyage NLP (Text Cleaning)**
2. **Création de la Variable Cible (Labeling)**
3. **Vectorisation (Feature Extraction)**

Pipeline de Nettoyage NLP (Text Cleaning)

➔ objectif: Transformer les commentaires bruts et bruyants en texte propre exploitable.

Traitements appliqués :

- Normalisation : Conversion en minuscules.
- Nettoyage : Suppression de la ponctuation, caractères spéciaux et chiffres.
- Filtrage : Retrait des Stopwords via NLTK.
- Racinisation (Stemming) : Réduction des mots à leur racine avec SnowballStemmer



- Validation du Nettoyage : L'élimination des stopwords et le stemming font émerger les thèmes clés de l'hôtellerie ("Chambre", "Personnel", "service”, “acceuil”).

==>Le bruit a été filtré avec succès.

- On voit des mots de sentiments: trés, bon excellent, propore

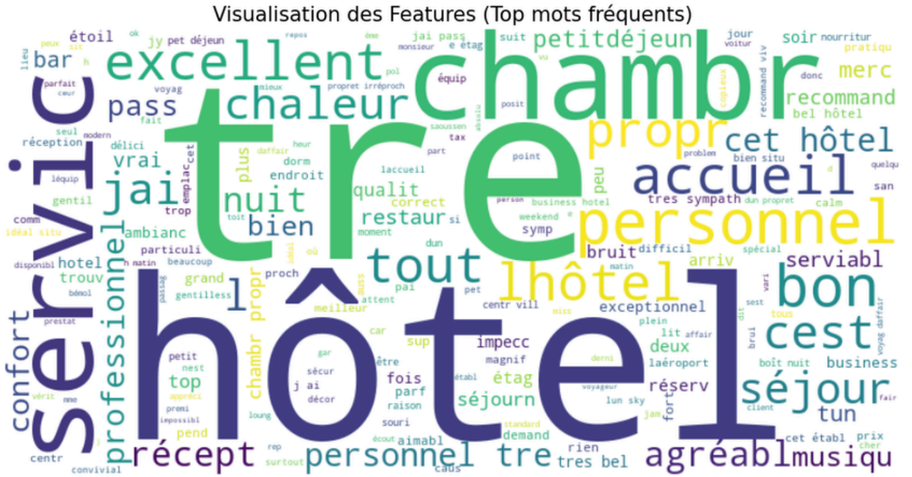
Vectorisation (Feature Extraction)

1. Construction du Vocabulaire (Bag-of-Words):

Matrice résultante: de dimension 201 x 1005:

- 201 lignes (Avis clients).
- 1005 colonnes (Le vocabulaire unique appris).

C’est notre espace de travail



Nuage de Mots:
En analysant la fréquence brute (Bag-of-Words), on voit que des termes génériques écrasent tout le reste.
Mais ne donnent aucune indication sur la satisfaction du client
C'est pour corriger ce défaut visuel que nous passons ensuite au TF-IDF

→ 2. Vectorisation

Méthode : Utilisation de TF-IDF (Term Frequency-Inverse Document Frequency).

Objectif: Pondérer selon la rarité statistique.

Résultat: réduire le poids des mots trop fréquents et de valoriser les termes spécifiques qui portent du sens.
->éliminer le bruit fréquentiel

→ Matrice finale : dimension de 201*1005 pndérée

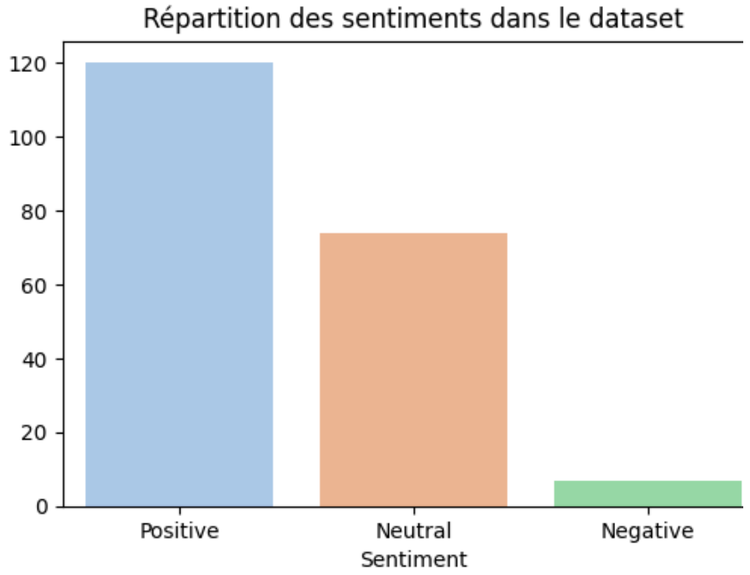
Génération des Labels & Analyse de Sentiment

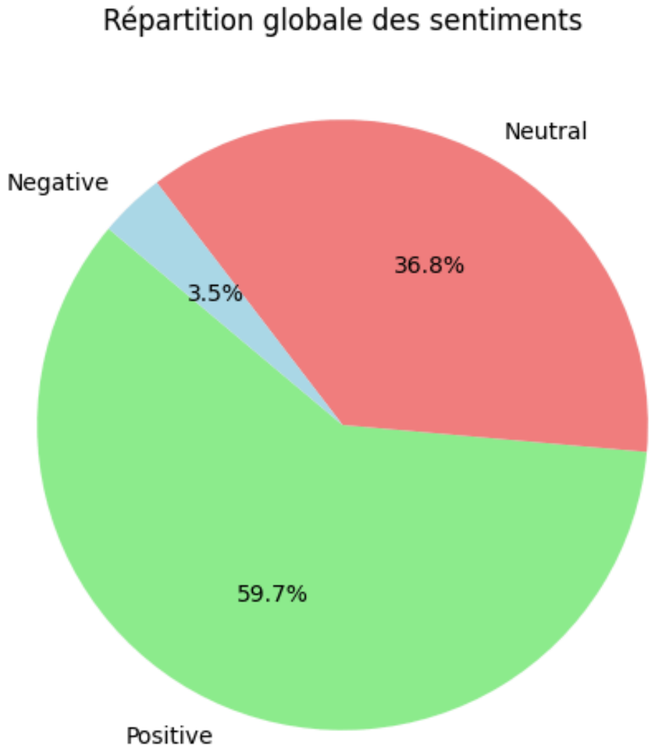
➔ Ground Truth

- Les données scrapées contiennent le texte mais pas de note (Rating) exploitable.
- Conséquence : Nous sommes dans un cas non supervisé, mais nous avons besoin de classes pour entraîner un classifieur.

➔ Solution : Labellisation Automatique

- Outil : Utilisation de TextBlob-FR (Approche Lexicale)
- Pas robuste mais suffisante pour un mini projet).
- Méthode : Calcul d'un score de polarité pour chaque avis (basé sur un dictionnaire de mots positifs/négatifs).
- Règle de décision :
  - Polarité >0.1 –> Positif
  - Polarité <0.1 –> Négatif
  - Polarité = Neutre





Cette distribution reflète une bonne satisfaction client (Business), mais crée un déséquilibre extrême (Data Science) la phase du training qui rendra la détection des avis négatifs très difficile

Environnement & Stratégie de Modélisation

Objectif
Construire un modèle de classification supervisée permettant de prédire le sentiment d'un avis client (Positive / Neutral / Negative).

Environnement & Outils

- Python 3.10 – langage principal
- Conda – gestion des environnements
- Jupyter Notebook – expérimentation

Libraries / Packages

- scikit-learn: vectorisation + algorithmes
- imbalanced-learn: oversampling
- pandas/ numpy – manipulation des données
- textblob/NLTK: Prétraitement NLP.
- joblib – sauvegarde du modèle

Sélection des Algorithmes

Nous avons mis en compétition trois algorithmes:

- Multinomial Naive Bayes : La "baseline" pour les données textuelles et petites, rapide et probabiliste.
- Logistic Regression : Simple, interprétable, robuste pour la classification binaire.
- SVM Linéaire: efficace sur données haute dimension (TF-IDF), souvent meilleur sur texte

Méthode d'Entraînement

Cross-Validation (Validation Croisée) :
Pour éviter le hasard d'une seule découpe (Train/Test) ->nous validons la robustesse du modèle sur 5 plis (K-Fold = 5)

```
=== Logistic Regression ===
Accuracy scores: [0.63414634 0.7        0.675      0.675      0.65       ]
Mean accuracy: 0.6668292682926829

=== Multinomial Naive Bayes ===
Accuracy scores: [0.58536585 0.65       0.65       0.625      0.6        ]
Mean accuracy: 0.6220731707317073

=== SVM (Linear) ===
Accuracy scores: [0.70731707 0.725      0.65       0.65       0.675      ]
Mean accuracy: 0.6814634146341463

 Meilleur modèle : SVM (Linear) avec accuracy moyenne = 0.6815
```

Tableau comparatif des performances (Cross-Validation 5-fold) accuracy ssur chaque Fold

Pourquoi le SVM a gagné ?
Le SVM est mathématiquement le plus performant pour gérer les espaces de haute dimension (nos 1005 features TF-IDF) même avec peu de données.

Évaluation de la Performance & Résultats

Le Piège de l'Accuracy: Avec 97% de données positives/neutres, l'Accuracy est un indicateur trompeur.

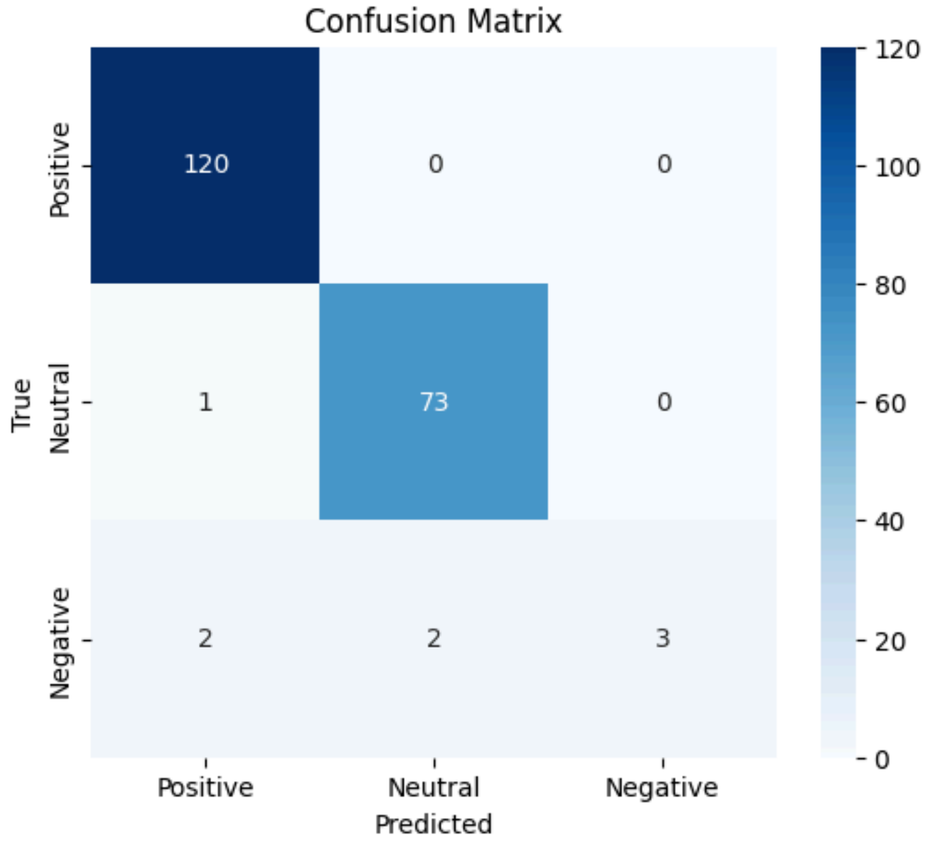
Notre Choix : F1-Score Pondéré (Weighted): C'est la moyenne harmonique de la **Précision** (qualité) et du **Rappel** (quantité).

```
f1 = f1_score(y, y_pred, average='weighted')
#weighted F1 pour prendre en compte déséquilibre éventuel
```

➔ **Le modèle a parfaitement assimilé les données d'entraînement (Accuracy > 97%).**

La pondération (Weighted F1) confirme une bonne gestion globale des classes.

Note critique : Cette performance très élevée s'explique par la taille réduite du corpus (201 avis) et la forte prédominance des avis positifs. Le modèle est "expert" sur ce dataset spécifique.

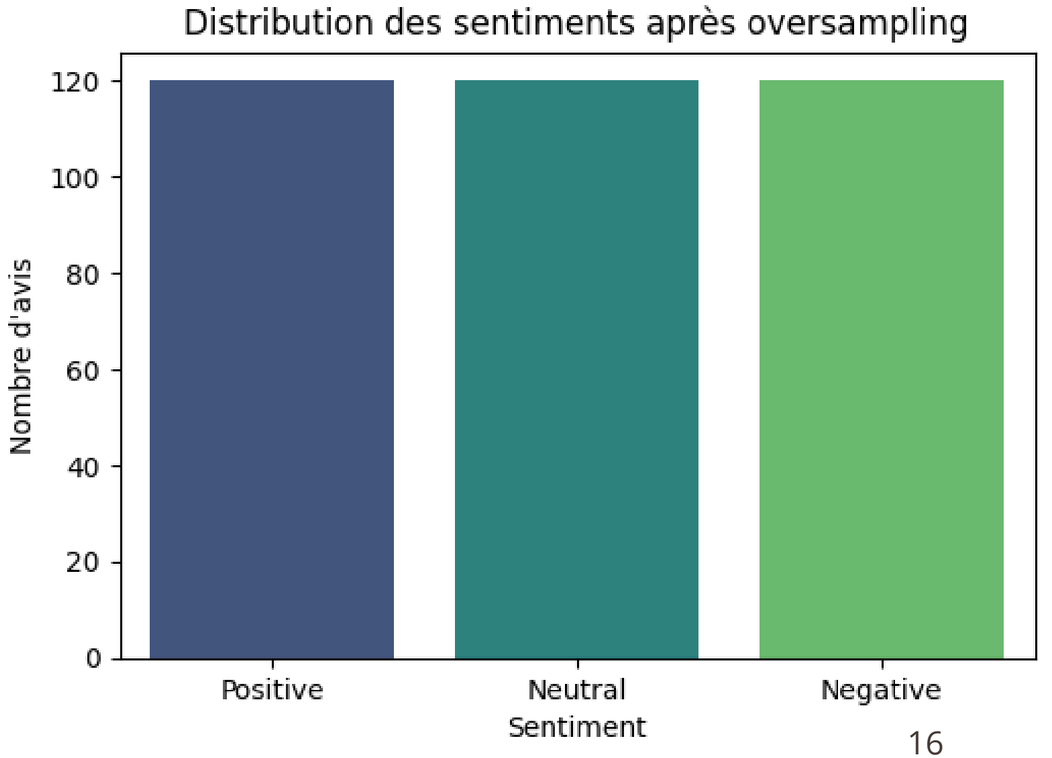


Accuracy (tout dataset): 0.9751
F1-score (weighted): 0.9713

Problèmes et améliorations

→ Le Diagnostic : Déséquilibre Critique

- **Réalité des Données :** Une disproportion majeure entre la satisfaction (120 avis positifs) et le mécontentement (7 avis négatifs).
- **Le Piège de l'Accuracy :** Avec les données brutes, un modèle "naïf" atteint 97% d'exactitude... simplement en prédisant "Positif" tout le temps.Le modèle naif maximise l'Accuracy globale au détriment du Rappel (Recall) de la classe minoritaire.
- **Conséquence :** Incapacité totale à détecter la classe "Négative" (Recall ~ 0).
- **Impact Métier :** Rater une plainte critique (Faux Négatif) coûte beaucoup plus cher à la réputation de l'hôtel que de rater un compliment: pour l'hôtel, détecter un client mécontent est l'objectif prioritaire.



Stratégie d'Amélioration: SMOTE (Oversampling)
SMOTE= Synthetic Minority Over-sampling Technique

- Principe : L'algorithme crée des exemples synthétiques (artificiels) en mélangeant les caractéristiques des avis négatifs existants via la librairie imbalanced-learn.
- Action : Il gonfle le nombre d'avis négatifs dans le jeu d'entraînement jusqu'à ce qu'il y en ait autant que les positifs

→ Le problème de biais est techniquement résolu.

Impact du Rééquilibrage (Résultats Post-SMOTE)

→ **Le SVM est encore gagant**

Performance:

Métrique : Moyenne sur 5 tests (K-Fold).
Score SVM : 90.00% (0.9000).
Évolution : On passe de ~68% (avant SMOTE) à 90%.
Le modèle est devenu très robuste.

Performance sur le Dataset Complet

- **Accuracy :** 99.44%
- **F1-Score (Weighted) :** 99.44%
- Focus Classe "Négative" :
  - **Précision :** 1.00
  - **Rappel (Recall) :** 1.00
- **Interprétation :** Le modèle ne rate plus aucun avis négatif dans ce jeu de données.

Deploiement

Test en Temps Réel

Nous avons soumis au modèle des avis inédits pour valider sa compréhension des concepts clés.

```
Review: Un personnel excellent et très accueillant
Predicted sentiment: Positive

Review: il est situé au centre ville de Tunis.
Predicted sentiment: Neutral

Review: cher, chambre bruitée , tres deconseillée.
Predicted sentiment: Negative
```

- Détection correcte des termes valorisants
- Identificationdu contenu factuel et non-émotionnel
- Reaction aux mots clés avec succès

Note: Limites & Améliorations Futures

Les Freins Actuels

Données : Volume critique (201 avis) + Bruit dans les labels (TextBlob).

Technique : Le SVM (TF-IDF) ignore le contexte (ironie, double négation)

Les Solutions Futures

Industrialisation : Scraper 2000+ avis pour combler les trous de vocabulaire.

Intelligence : Passer au Deep Learning (CamemBERTpar exemple) pour comprendre les nuances et faire du NLP robuste.

Topic Modeling : Clustering Non-Supervisé

Utiliser des algorithmes (type LDA ou K-Means) pour grouper automatiquement les avis par sujets dominants : Propreté, Service, Restauration, Isolation.==> Identifier précisément les causes de satisfaction ou de plainte, au-delà du simple score positif/négatif.

MERCI !