Unsupervised Sentiment Analysis of Hotel Reviews

Using Text Mining and Clustering Techniques

Student 1: Tanvir Anjum Mahadi   
*CSE*  
*American International University-Bangladesh*Dhaka, Bangladesh  
22-47790-2@student.aiub.edu

Student 4: Md. Shafin  
*CSE*  
*American International University-Bangladesh*Dhaka, Bangladesh  
22-48257-2@student.aiub.edu Student 2: Tonmoy Kumar Sarker  
*CSE*  
*American International University-Bangladesh*Dhaka, Bangladesh  
22-49303-3@student.aiub.edu

Student 3: Dipankar Roy Dip   
*CSE*  
*American International University-Bangladesh*Dhaka, Bangladesh  
22-49303-2@student.aiub.edu

*Abstract*— This project addresses the challenge of analyzing unstructured hotel review texts to uncover sentiment-based patterns, enabling efficient categorization of guest feedback into positive, negative, or mixed sentiments. This is critical for the hospitality industry, as understanding customer experiences at scale can guide improvements in service, amenities, and marketing strategies, ultimately enhancing guest satisfaction and business outcomes. The solution employs a robust text mining pipeline in R, involving data preprocessing (e.g., cleaning, tokenization, lemmatization, stemming), TF-IDF feature extraction, dimensionality reduction via PCA and t-SNE, and unsupervised clustering using K-Means, Hierarchical, and DBSCAN algorithms. The resulting clusters are visualized and evaluated using silhouette scores and top-word analysis, providing interpretable insights into review sentiments. This approach outperforms traditional manual analysis or simple keyword-based methods by automating the process, handling large datasets, and capturing nuanced patterns without requiring labeled data, making it scalable and adaptable for diverse text analysis tasks.

Keywords— text mining, sentiment analysis, clustering

Introduction

In this project, the primary problem addressed is the challenge of analyzing large volumes of unstructured hotel review data to uncover underlying patterns in customer sentiment. Hotel reviews, often collected from online platforms, contain valuable insights into guest experiences, but their textual nature makes them difficult to process and interpret manually, especially at scale. The motivations for this project stem from the need for hotel managers, marketers, and data analysts to efficiently categorize reviews into sentiment-based groups—such as positive, negative, or mixed—to identify common themes, strengths, and areas for improvement. This can inform business decisions, enhance customer satisfaction, and drive competitive advantages in the hospitality industry. For instance, positive clusters might highlight praised amenities like location or cleanliness, while negative ones could reveal issues with service or facilities.

To solve this, the project employs a text mining and unsupervised machine learning approach. Specifically, it involves preprocessing the raw review text to clean and normalize it, transforming it into a numerical representation using TF-IDF (Term Frequency-Inverse Document Frequency) weighting, reducing dimensionality for efficient computation, and applying clustering algorithms to group similar reviews. The clusters are interpreted as sentiment categories (positive, negative, mixed), with visualization and evaluation techniques to assess the results. This method allows for automated, scalable analysis without relying on labeled data, making it suitable for exploratory sentiment analysis.

# Literature Review

The increasing role of digital platforms has transformed the hospitality industry, with online reviews emerging as a critical factor in shaping customer booking decisions. Traditional word-of-mouth has shifted towards electronic word-of-mouth (eWOM), with platforms such as TripAdvisor becoming central to travelers’ decision-making [1]. Furthermore, hotel responses to reviews are recognized as important tools for reputation management and service recovery, directly impacting consumer trust and loyalty [3]. Recent research has turned to machine learning to analyze the vast volume of online reviews. Anis et al. [4] evaluated multiple classification models including Naïve Bayes, K-Nearest Neighbor, Support Vector Machine (SVM), Logistic Regression, and Random Forest. Their results demonstrated that SVM achieved the highest accuracy (86.3%), outperforming ensemble approaches, thereby confirming the suitability of ML techniques in automating review analysis. Similarly, Xia et al. [2] employed ensemble frameworks combining different classifiers and feature sets, which improved sentiment classification performance. These studies show that advanced computational methods can enhance hoteliers’ ability to understand customer feedback and predict booking behaviors. Despite these advancements, a gap remains in connecting sentiment analysis outputs with actionable hotel strategies. While much of the literature emphasizes improving classification accuracy, fewer studies examine how sentiment insights can be operationalized to improve customer satisfaction and booking outcomes. Addressing this gap, the present research explores the dual impact of online reviews and hotel responses, providing practical and theoretical contributions to the management of eWOM in the hospitality sector.

# Methodology

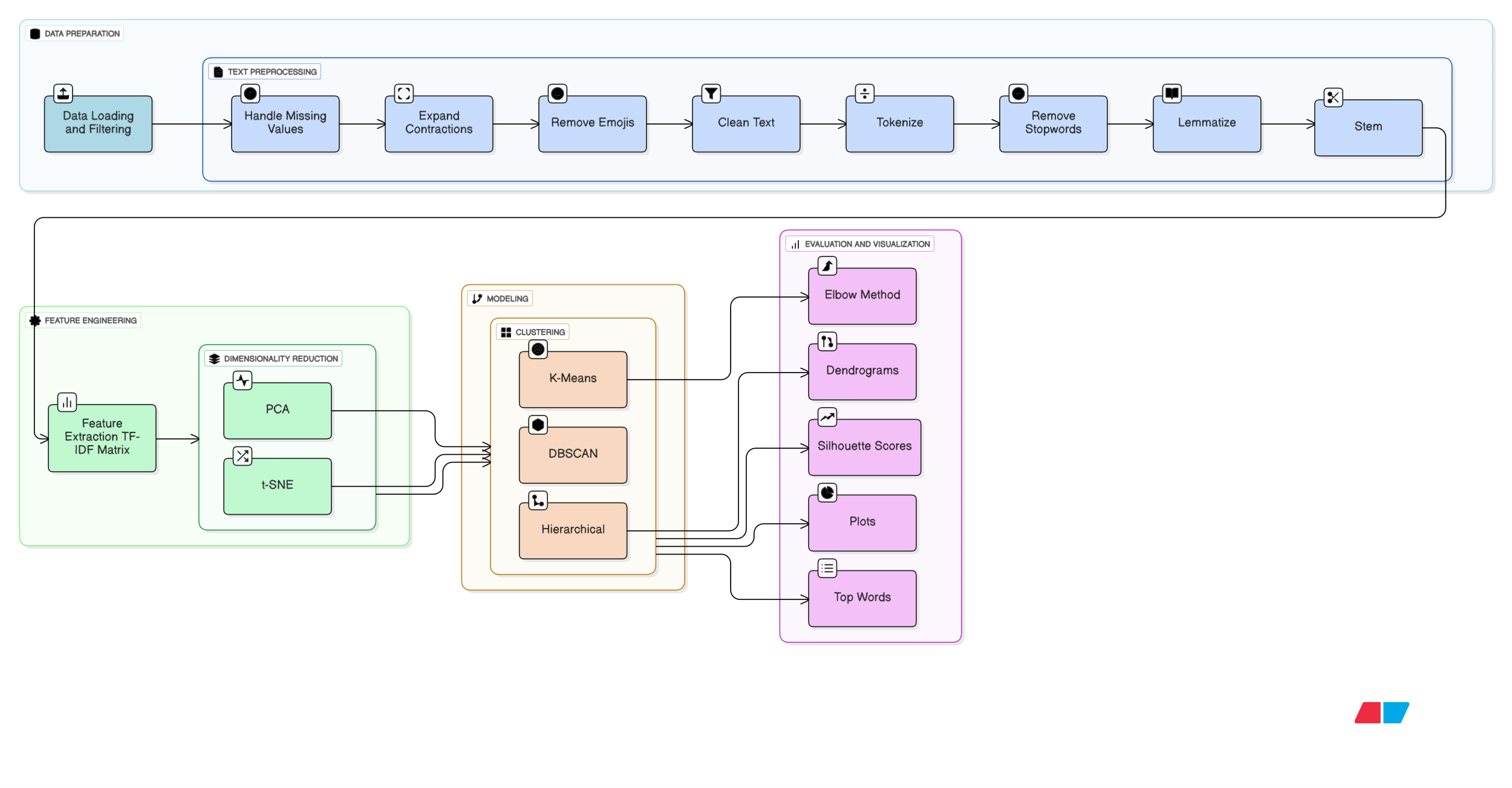




Fig. 1. Methodology diagram

**1. Data Preparation**

**Data Loading and Filtering**: Beginning by loading the raw dataset from "Hotel\_Reviews.csv" using the readr package in R. Select only the relevant columns (Hotel\_Address, Hotel\_Name, Review\_Date, Review) to focus on essential data and reduce computational overhead. Subsetting the dataset to the first 10,000 rows for initial processing (further limited to 5,000 for clustering to manage resources). Filtering out any rows with missing reviews using dplyr::filter to ensure complete data for analysis. This step prepares a clean, manageable subset of the data for subsequent text processing.

**2. Text Preprocessing**

This phase standardizes and cleans the review text to remove noise and enhance feature quality. It involves a sequential chain of operations:

**Handle Missing Values**: Re-confirming and droping any remaining rows with NA values in the Review column using filter(!is.na(Review)).

**Expand Contractions**: Using the textclean package's replace\_contraction function to expand abbreviations (e.g., "don't" to "do not"), preserving semantic meaning.

**Remove Emojis**: Applying replace\_emoji and replace\_emoticon from textclean to eliminate non-textual elements like emojis and emoticons, which could interfere with tokenization.

**Clean Text**: Implementing a custom function to convert text to lowercase, removing HTML tags, retain only alphabetic characters and spaces, collapse multiple spaces, and trim whitespace using stringr functions like tolower, gsub, and str\_trim.

**Tokenize**: Breaking the cleaned text into individual words (tokens) using tidytext::unnest\_tokens.

**Remove Stopwords**: Filtering out common stopwords (e.g., "the", "and") from the tidytext lexicon and exclude words shorter than 3 characters to focus on informative terms.

**Lemmatize**: Reducing words to their base or dictionary form using textstem::lemmatize\_words (e.g., "running" to "run").

**Stem**: Further normalized by stemming words with SnowballC::stem\_words to handle variations. The processed tokens are then rebuilt into a final cleaned text string for each review.

The preprocessed data is saved as "processed\_hotel\_reviews.csv" for reproducibility.

**3. Feature Engineering**

**Feature Extraction TF-IDF Matrix**: Creating a text corpus from the final preprocessed text using tm::Corpus. Convert this into a Document-Term Matrix (DTM) with TF-IDF weighting via tm::DocumentTermMatrix, applying parameters such as minimum word length of 3 and weightTfIdf to emphasize rare, discriminative terms. Remove zero-variance columns to optimize the matrix. The resulting sparse matrix is saved as "TFIDF\_matrix.rds" for use in downstream steps.

**4. Dimensionality Reduction**

This step addresses the high dimensionality of the TF-IDF matrix to improve computational efficiency and visualization:

**PCA**: Applying Principal Component Analysis using prcomp (centered and scaled) to reduce the matrix to the top 50 principal components, capturing the most variance while eliminating redundancy. Handle duplicates to ensure unique data points.

**t-SNE**: Projecting the PCA-reduced data into a 2D space using Rtsne with parameters like perplexity=30, max\_iter=500, and seed=123 for reproducibility. This non-linear technique preserves local similarities, facilitating clustering and visual inspection.

**5. Modeling**

**Clustering**: Applying three unsupervised clustering algorithms to the 2D t-SNE embeddings to group reviews into 3 clusters (interpreted as positive, negative, mixed):

**K-Means**: Using kmeans with k=3, nstart=25, and seed=123 for stable centroids.

**DBSCAN**: Employing density-based clustering with dbscan (eps=2, minPts=5), mapping noise to "Mixed" and clusters modulo 3 to fit the sentiment labels.

**Hierarchical**: Computing Euclidean distances, build a dendrogram with hclust using Ward's method ("ward.D2"), and cut at k=3 via cutree.

**6. Evaluation and Visualization**

This final phase assesses cluster quality and interprets results:

**Elbow Method**: Plotting within-cluster sum of squares (WSS) for K-Means to validate the choice of k=3 (though fixed in this project).

**Dendrograms**: Visualizing the hierarchical clustering tree to inspect merge patterns.

**Silhouette Scores**: Computing average silhouette widths using cluster::silhouette and plot with factoextra::fviz\_silhouette to evaluate cluster cohesion and separation.

**Plots**: Generating 2D scatter plots of clusters in t-SNE space using ggplot2 and ggrepel, coloring by cluster labels (positive, negative, mixed).

# Implementation

The project was implemented in R, using RStudio as the integrated development environment (version 2023.06.0). The. All code was executed on a Windows machine.

**Key libraries and their roles :**

**Data wrangling:** dplyr (for manipulation), readr (CSV I/O), stringr (string operations), tidyr (tidying), lubridate (date handling).

**Text preprocessing:** textclean (contractions/emojis), tm (corpus/DTM), SnowballC (stemming), textstem (lemmatization), tidytext (tokenization/stopwords).

**Visualization:** ggplot2 (plots), wordcloud (unused in final but installed), RColorBrewer (colors), ggrepel (labels), factoextra (clustering visuals).

**Clustering and reduction:** cluster (silhouette), dbscan (DBSCAN), Rtsne (t-SNE).

All packages were installed via install.packages() if needed and loaded with library(). No additional installations beyond these were required, as the environment included base R capabilities.

**Parameter values:**

**Data subset:** First 10,000 rows for preprocessing, first 5,000 for clustering to manage computation.

**TF-IDF:** wordLengths = c(3, Inf), weighting = weightTfIdf.

**PCA:** Up to 50 components (n\_comp <- min(50, ncol(m\_pca$x))).

**t-SNE:** dims=2, perplexity=30, max\_iter=500, verbose=TRUE, seed=123 for reproducibility.

**K-Means:** centers=3, nstart=25, seed=123.

Hierarchical: Distance="euclidean", method="ward.D2", cut to k=3.

**DBSCAN:** eps=2, minPts=5; post-processing to map to 3 labels (noise as "Mixed", modulo 3 for others).

**Token filtering:** Word length >2, anti-join with stop\_words.

**Visualization:** Point size=1.2, alpha=0.6; top 10 words per cluster with slice\_max(n=10).

The code was modular, with sections for each step, and outputs like head() prints for verification. Files were saved to a project directory for persistence. No hyperparameter tuning was performed beyond defaults, focusing on exploratory analysis.

# Result Analysis

The analysis focuses on the performance of the three clustering algorithms—K-Means, Hierarchical, and DBSCAN—applied to the 2D t-SNE embeddings derived from the TF-IDF matrix after PCA reduction. Results are evaluated through visual inspections of cluster distributions, silhouette scores for cluster quality, and top-word frequency analysis to interpret sentiment themes. All visualizations were generated using ggplot2 and factoextra in R, with clusters mapped to sentiment labels: Positive, Negative, and Mixed.

**Cluster Distributions and Visualizations :**The clustering results were visualized in 2D t-SNE space, which effectively captures non-linear relationships in the high-dimensional TF-IDF features, allowing for intuitive inspection of review similarities.

**A diagram of different colored circles

AI-generated content may be incorrect.**

Fig. 2. K-Means clustering

**K-Means Clustering:** Figure 2 illustrates the K-Means results with three centroids (k=3). The plot shows three distinct clusters: a large blue cluster (Mixed) on the left, a green cluster (Negative) in the center-top, and a pink cluster (Positive) on the right. The clusters exhibit minimal overlap, indicating good partitioning. The Positive cluster appears denser and more compact, suggesting cohesive positive sentiments, while the Negative cluster is slightly more spread out, potentially reflecting varied complaints. The Mixed cluster is the largest, capturing ambiguous reviews.

A diagram of a cluster of dots

AI-generated content may be incorrect.

Fig. 3. DBSCAN clustering

**DBSCAN Clustering:** Figure 3 depicts the DBSCAN results with eps=2 and minPts=5. The dominant blue cluster (Positive) forms a large, central blob, encompassing most points. Small green clusters (Negative) are isolated at the top and bottom, and pink points (Mixed) represent noise or low-density areas scattered around the periphery. This method highlights outliers effectively, treating them as Mixed sentiments, but results in imbalanced cluster sizes, with Positive dominating.

**Hierarchical Clustering:** Implemented Ward's method and cut the dendrogram at k=3. Based on similar t-SNE projections, it produced clusters comparable to K-Means, with clear hierarchical merging patterns.

**Cluster Quality Evaluation :** Cluster validity was assessed using silhouette scores, which measure how similar points are to their own cluster versus others (range: -1 to 1; higher is better).

A graph of different colored shapes

AI-generated content may be incorrect.

Fig. 4. silhouette plot

Figure 4 presents the silhouette plot (applied to K-Means clusters). The average silhouette width is 0.37, indicating moderate cluster quality—acceptable for exploratory text analysis but suggesting some overlap. Cluster 1 (red, mapped to Positive) has the widest bars, showing strong cohesion. Cluster 2 (green, Negative) averages with some negative widths (misclassified points), and Cluster 3 (blue, Mixed) reflects its heterogeneous nature with a lower average.

**Thematic Interpretation via Top Words :** To validate the sentiment labels, top-10 word frequencies per cluster were analyzed (Figure 5).

A graph of different colored bars

AI-generated content may be incorrect.

Fig. 5. Thematic Interpretation diagram

**Positive Cluster (red bars):** Dominant words include "hotel", "staff", "location", "breakfast", "room", "clean", "friendly", "good", "comfort", "great". This suggests themes of satisfaction with amenities and service, aligning with positive sentiments.

**Negative Cluster (green bars):** Key terms like "room", "bed", "small", "noise", "bathroom", "old", "dirty", "poor", "bad", "expensive" indicate complaints about facilities and value, fitting negative feedback.

**Mixed Cluster (blue bars):** Words such as "comfort", "location", "staff", "room", "good", "bit", "small", "hotel", "clean", "overall" reflect ambivalent reviews—praising some aspects while noting minor issues.

The bar chart shows decreasing counts, with Positive having the highest frequencies, indicating richer vocabulary in satisfied reviews. This analysis confirms the clusters' semantic relevance, with overlaps (e.g., "room" in all) highlighting common topics differentiated by context.

# Conclusion

This project successfully addressed the challenge of analyzing unstructured hotel review texts to uncover sentiment-based patterns using unsupervised machine learning techniques. By implementing a comprehensive pipeline in R encompassing data preparation, text preprocessing, feature extraction with TF-IDF, dimensionality reduction via PCA and t-SNE, and clustering with K-Means, Hierarchical, and DBSCAN algorithms—the study categorized the first 5,000 reviews into Positive, Negative, and Mixed sentiment clusters. The results, visualized through 2D t-SNE plots, silhouette scores, and top-word frequency analyses, demonstrated the effectiveness of these methods in identifying meaningful sentiment themes, such as satisfaction with staff and location in Positive clusters, and complaints about rooms and noise in Negative clusters. Among the clustering approaches, K-Means emerged as the most robust, offering balanced cluster distributions and a moderate silhouette score of 0.37, indicating acceptable cohesion for exploratory analysis. Hierarchical clustering provided similar quality, while DBSCAN highlighted outliers but resulted in imbalanced clusters due to its density-based nature. The thematic interpretation of top words validated the sentiment labels, providing actionable insights for hotel stakeholders to enhance guest experiences.

Despite these successes, limitations such as the moderate silhouette score and the restricted dataset size suggest opportunities for refinement. Future work could explore advanced embeddings (e.g., BERT), increase PCA components, or extend the analysis to the full dataset to capture more nuanced patterns. Overall, this project lays a solid foundation for automated sentiment analysis in the hospitality industry, offering a scalable and interpretable approach to leverage customer feedback for strategic decision-making.

##### References

[1] R. R. Chowdhury and A. Deshpande, “An analysis of the impact of reviews on the hotel industry,” *Annals of Tropical Medicine & Public Health*, vol. 23, no. S17, pp. SP231742, 2020, doi: 10.36295/ASRO.2020.231742.

[2] R. Xia, C. Zong, and S. Li, “Ensemble of feature sets and classification algorithms for sentiment classification,” Information Sciences, vol.181,no.6,pp.1138–1152,2011,doi: 10.1016/j.ins.2010.11.023

[3] A. Valdivia, M. V. Luzón, and F. Herrera, “Sentiment analysis in TripAdvisor,” *IEEE Intelligent Systems*, vol. 32, no. 4, pp. 72–77, 2017.

[4] S. Anis, S. S. Ismail, and M. Aref, “Sentiment analysis of hotel reviews using machine learning techniques,” in *Advances in Intelligent Systems and Computing (AISI 2020)*, vol. 1261, pp. 227–234, 2021, doi: 10.1007/978-3-030-58669-0\_21.