Semantic-driven hybrid recommender system using embeddings for Chicago Airbnb listings

ABSTRACT

This project develops a recommender system for Chicago Airbnb listings using the Inside Airbnb database. The system identifies the top k listings based on three preferences: guest count, price point, and text comments. Each recommendation includes the location and distance to the nearest subway station, ensuring enhanced accessibility. The project utilizes three datasets: the Airbnb listings dataset with 7,952 entries, the reviews dataset with 417,795 entries, and the Chicago CTA stations dataset with 145 entries. The process consists of four stages, Exploratory Analysis, Review Scoring, Listing Clustering, and Recommender System, each building on the previous stage to create an efficient, comprehensive system.

KEYWORDS: Recommender System, Sentiment Analysis, Clustering, Geospatial Map, Embedding

INTRODUCTION

Chicago is well known as one of the popular tourist destinations in the United States. Indeed, the city is notorious for its rich food culture, scenic downtown, and accessible public transportation. The diversity of the city in population, cultures, attractions, cuisine, and recreational activities has significantly impacted its economy, particularly the hospitality industry, from hostels, and hotels to even couch surfing. Over the years, the traditional hospitality industry has been taken over by a company named Airbnb, Inc. – a San Francisco-based company offering an online marketplace for short- and long-term stays and experiences. Acting as a broker between renters and property owners, Airbnb has evolved from a small-scale income source for homeowners to a mainstream platform for hospitality entrepreneurs. This project will utilize Inside Airbnb database for Chicago to develop a semantic-driven hybrid recommender system, designed to assist tourists in discovering the best Airbnb listings aligned with their preferences. The recommendation system will also factor in proximity to subway stations, ensuring visitors have optimal access to public transportation during their stay.

Several methods were applied in data preprocessing to refine the data. For example, thorough exploratory analysis of the Airbnb listings dataset was conducted using line and bar charts, histograms, and geospatial maps to describe and visualize relationships between attributes. In the reviews dataset, techniques were applied to generate and validate sentiment scores, labeling reviews as 'negative,' 'neutral,' or 'positive.' VADER was used as a benchmarking model, and word clouds visually verified the initial sentiment output. Sentiment predictions were tested and cross-validated using Logistic Regression, Linear SVC, and Naive Bayes, comparing results with VADER's output. Airbnb listings are grouped into three clusters based on names and descriptions using Hierarchical Clustering, helping identify common themes among listings. This clustering provides a nuanced view, allowing the recommender system to tailor suggestions based on content and style similarities. The spatial relationship between each listing and subway station is calculated with the geopy library to ensure accurate distance measurements. This feature is especially useful for travelers and locals relying on the CTA, enhancing the practicality of each recommendation.

The final dataset was compiled to include detailed information about each listing, such as amenities, neighborhood, accommodation type, nearest subway station and its associated details, Airbnb category derived from clustering, and sentiment-based review ratings. This comprehensive dataset served as the foundation for building a recommender system that

leverages embedding similarity between the user's input and the attributes of each listing. The system effectively provides users with tailored recommendations based on their specified preferences, enhancing the relevance and personalization of the search results. Despite these successes, there remains significant room for improvement. Additionally, the project was implemented using Google Colab with GPU and Python 3.10, which provided an accessible and flexible development environment.

LITERATURE REVIEW

Ensemble hotel recommender

Ray, Garain, and Sarkar (2021) conducted a study on building a hotel recommender system using an ensemble-based approach to sentiment analysis, combining two distinct methodologies: contextual embeddings and traditional NLP features to capture nuanced sentiment expressions in text. The contextual embeddings section utilizes variations of the BERT (Bidirectional Encoder Representations from Transformers) model in two approaches. The first approach includes two BERT models, with BERT1 performing binary classification (positive/negative) and BERT2 and BERT3 performing multi-label classification (positive/negative/neutral). The second approach uses a single BERT4 model for multi-label classification. In contrast, the traditional NLP features section achieves feature extraction through DOC2VEC for document-level representation and TF-IDF for evaluating word importance. Additionally, Word2Vec with Gensim is used for word-level feature extraction, while SentiWordNet provides polarity scores to assess sentiment. A random forest classifier then performs multi-label classification, contributing to the overall sentiment assessment. The ensemble method integrates outputs from these models to produce a final sentiment score, aiming for greater accuracy by synthesizing insights from multiple approaches. The authors also apply aspect categorization focused on nouns and use K-RMS clustering to categorize reviews based on similarity using eight dominant index terms. To validate the model, various deep learning pre-trained models, such as Recurrent Neural Network (RNN), Bidirectional LSTM (Bi-LSTM), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU), were employed. This approach of using multiple models for sentiment analysis validation and weight assessment will be considered for the project.

Tourism recommender

A study conducted by Abbasi-Moud, Vahdat-Nejad, and Sadri (2021) utilized TripAdvisor data, focusing strictly on nouns, to build a tourism recommender system based on user preferences and attraction features. The process involved three main stages. First, user preference extraction was conducted through sentiment analysis of graph-based noun clusters using SentiWordNet 3.0 and semantic similarity computation with WordNet. Semantic similarity was measured using methods such as Wu-Palmer similarity, which assesses concept depth, and Extended Gloss Overlaps, which evaluates definition overlap. In the attraction feature extraction stage, reviews were grouped by attraction and then further organized by weather conditions in a nested grouping approach, allowing the model to capture how attraction characteristics vary with the weather. Finally, in the recommender system stage, semantic similarity measures were applied to rank attractions based on similarity scores, with adjustments made using context-aware filters (such as current time, weather, and location) to increase the relevance of recommendations. This layered approach enhances recommendation accuracy and personalization. To validate sentiment analysis, the authors used a Support Vector Machine, Naive Bayes, and ground truth user reviews. Additionally, Cosine and Jaccard similarities were

used for further semantic similarity measurements. For this project, the approach of focusing on dominant words will be applied, although other parts of speech (such as verbs, adverbs, and adjectives) will also be utilized, rather than restricting the analysis to nouns only.

DATASET

Three datasets in .csv format were incorporated into this project, along with a Chicago shapefile for geospatial mapping. The Chicago Airbnb listings and reviews datasets were retrieved from Inside Airbnb. Additionally, the CTA dataset will be used in the recommender system. This dataset, provided by the Chicago Transit Authority and published by the Chicago Data Portal, contains information on train stations within the City of Chicago.

Listings Data

The original listings dataset consists of 7,952 Airbnb listings – Geospatial Map, each with 75 attributes. Given the large number of attributes, irrelevant columns were removed to focus only on those pertinent to this project. This reduction resulted in a dataset with 18 columns. Table 1 provides a summary.

Table 1: Listings Dataset Schema

ATTRIBUTE	DESCRIPTION	DATA TYPE
id	Listing ID	Numerical
accommodates	Maximum capacity of the listing	
availability_365	Availability x days in the future as determined by the calendar	
bathrooms	Number of bathroom(s)	
beds	Number of bed(s)	
bedrooms	Number of bedroom(s)	
latitude	Latitude of the listing	
longitude	Longitude of the listing	
price	Rental price	
log_price	Rental price (log)	
review_scores_accuracy	Review scores based on scale 1-5	
reviews_count	Total reviews counts applied time weight	
amenities	Available amenities	Categorical
name_description	Joint data of description and name	
neighbourhood_cleansed	Chicago neighborhoods	
room_type	Type of rental	
host_is_superhost	Whether host has super host status	Boolean
host_identity_verified	Whether host is verified	

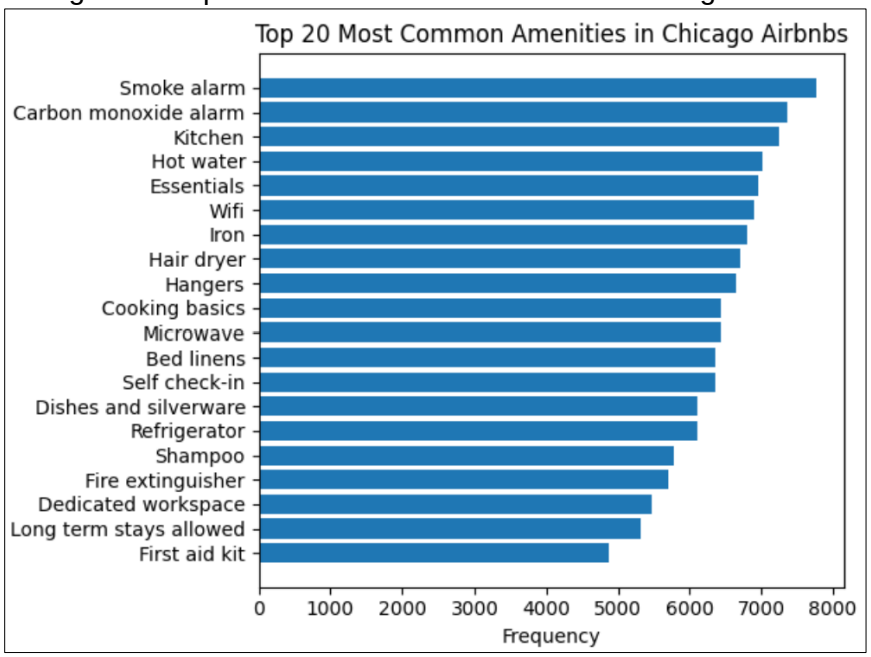
The following preliminary pre-processing steps were applied:

- Applied time-weighted decay to re-count the total number of reviews for each listing, giving higher weight to recent reviews. The decay specifications are as follows: Reviews older than 12 months: weight = 0.8 | Reviews from the last 12 months: weight = 0.9 | Reviews from the last 30 days: weight = 1.0

- Removed irrelevant attributes such as scraping-related data, URL links, and other information not required for the project scope, as well as columns with a high percentage of missing values.

There are 3,416 unique amenities available across all Chicago Airbnb listings, catering to various guest requirements and preferences. The top 20 most common amenities are focused on safety features, such as smoke alarms and fire extinguishers, followed by basic necessities like Wi-Fi, kitchen access, and hot water, as shown in Figure 1. The host with the most listings is Blueground with host ID: 107434423 and 549 listings, a well-known hospitality company and a competitor of Airbnb. Most listings are located in the Near North Side neighborhood, with 913 listings in this area. Additionally, the majority of listings can accommodate up to 8 guests, Figure 2. The most common room type in the dataset is Entire Home/Apartment, accounting for 6,165 listings (77.5% of all Airbnb listings), followed by Private Room, while Shared and Hotel Rooms represent only a minimal share of the listings.

Figure 1. Top 20 most common amenities in Chicago Airbnb



In terms of pricing, the average price generally increases with accommodation capacity, suggesting a high demand for larger accommodations that likely appeal to groups or families, Figure 3. For Entire Home/Apartment listings, this upward trend in price aligns with an average rental price of $254.64, reflecting the demand for spacious rentals. However, there are some anomalies, particularly in shared and private rooms, where certain listings with higher accommodation capacities have unexpectedly low prices. In contrast, hotel room prices exhibit greater variability as capacity increases, which could be due to additional factors such as luxury features or amenities influencing pricing, as indicated by their higher average price of $535.16. For private and shared rooms, the price trend fluctuates significantly, implying that factors such as location, property quality, or host ratings may impact prices more than accommodation capacity alone, given their much lower average prices of $97.53 and $59.44, respectively.

Figure 2. Distribution of accommodates

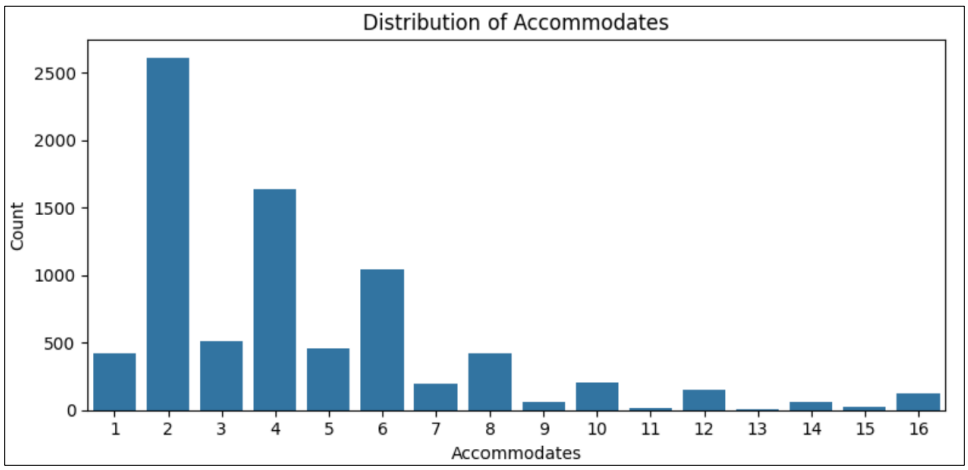
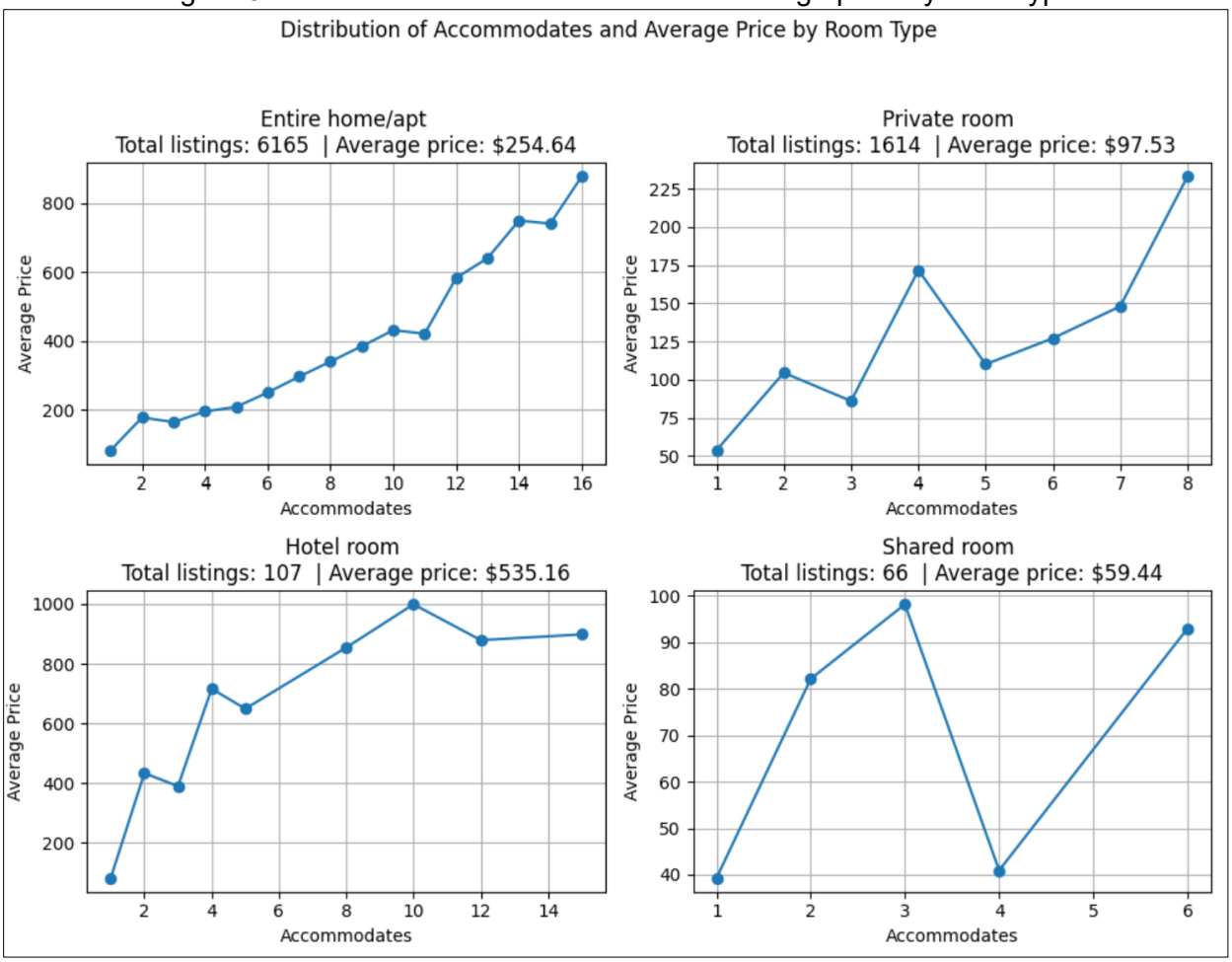


Figure 3. Distribution of accommodates and average price by room type



The original price distribution is heavily skewed to the right, indicating that most Airbnb listings are priced at the lower end of the range (less than $500), with a few outliers at very high prices. Listings with extremely high prices (over $2000) likely represent high-end or premium properties. After applying a log transformation, the price distribution approximates a normal

(bell-shaped) distribution, Figure 4, which reduces the impact of high-priced outliers and provides a more balanced view of the data for analysis and modeling.

Figure 4. Distribution of price

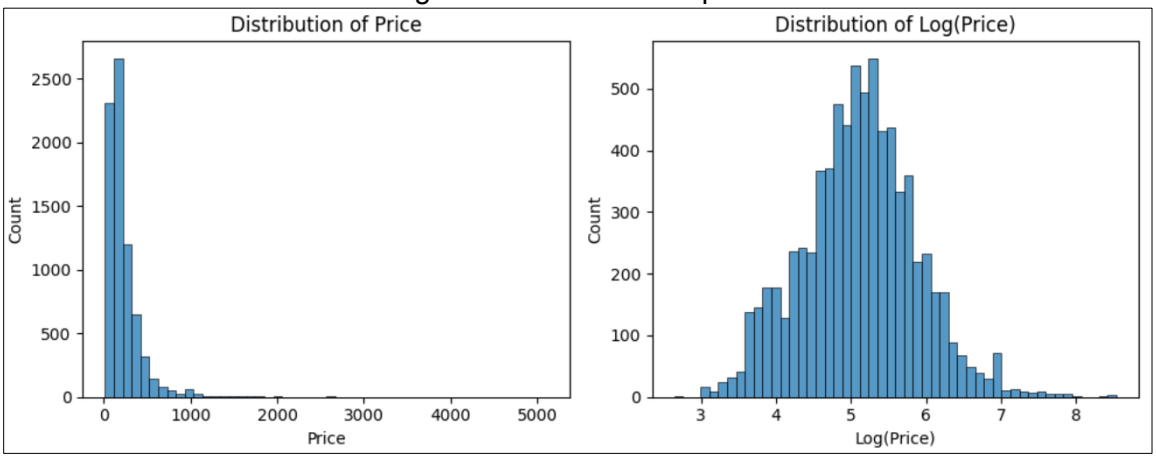
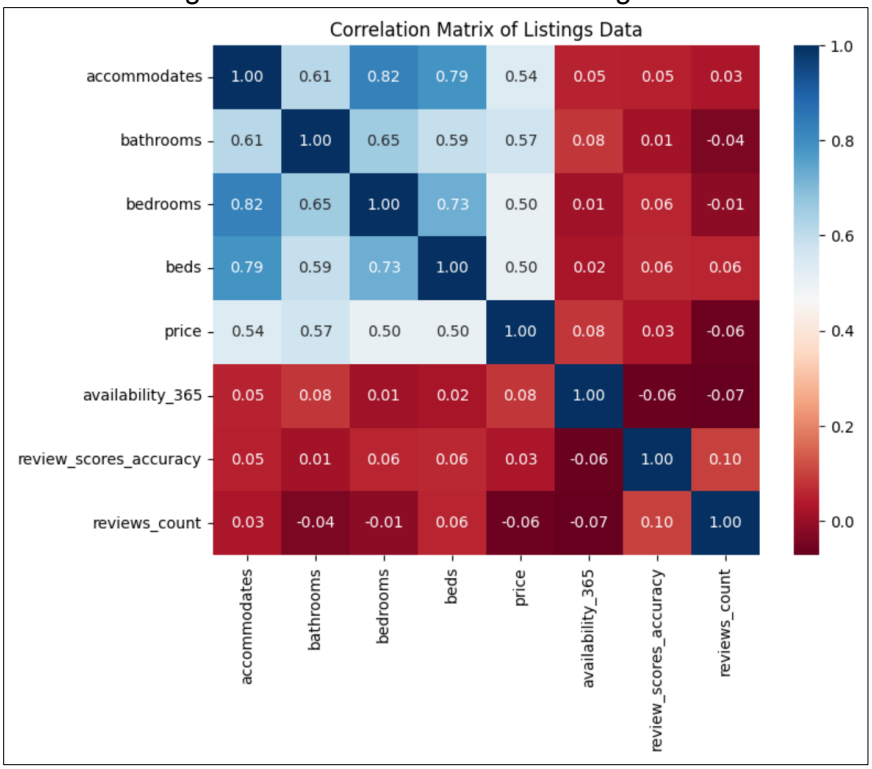


Figure 5. Correlation matrix of listings data



The correlation matrix reveals strong positive correlations between the number of people a listing can accommodate and variables like bedrooms and beds, as well as moderate positive correlations between price and the number of bedrooms and beds, as shown in Figure 5. This is expected, as accommodating more guests requires more space, which typically results in a higher price. Interestingly, the variable "Availability 365" has negligible correlations with all other variables, suggesting that the number of days a listing is available does not significantly impact

or depend on other factors. Similarly, review scores show minimal correlations, indicating that customer booking decisions may not be heavily influenced by reviews alone.

Reviews Data

The original reviews dataset consisted of 417,795 entries with 6 attributes, representing multiple reviews for each listing. The goal of this data preparation is to conduct effective sentiment analysis. Given the text-heavy nature of the data, extensive preprocessing steps were applied:

- Replaced special new line characters.
- Removed non-English comment using fast_langdetect library.
- Replaced non-Latin characters and emojis with empty string using regex.
- Filtered and/or removed short comments given condition length less than 2. Usually these are a single emoji, missing values, or incomplete text that cannot make up a word.
- Removed unnecessary columns that does not need for sentiment analysis.

Further preprocessing steps were then applied to refine the text for analysis: Non-ASCII characters were replaced with spaces, all text was converted to lowercase for consistency, punctuation and numbers were removed, finally compound words connected by hyphens or underscores were split into separate tokens. Ultimately, the dataset was reduced to 403,291 reviews with six attributes, Table 2. This refined dataset is now ready for tokenization and further processing to accurately capture the sentiment of each review.

Table 2: Reviews Dataset Schema

ATTRIBUTE	DESCRIPTION	DATA TYPE
id	Review ID	Numerical
listing_id	Listing ID	
reviews_length	Review length, number of total tokens	
comments	Single review of a listing	Categorical
comments_tokenized	Tokens of ‘comments’	
date	Date of the review	

Chicago Transit Authority (CTA) Data

Table 3: CTA Dataset Schema

ATTRIBUTE	DESCRIPTION	DATA TYPE
latitude	Latitude of the station	Numerical
longitude	Longitude of the station	
station_info	Name and color of the station	Categorical

The original CTA dataset contained 302 observations and 17 attributes, detailing various train stations, including information on train direction. The objective was to only keep each unique station and its essential information, ignoring directional details, and to map each listing to its nearest train station based on longitude and latitude. After processing, the reduced CTA dataset - Geospatial Map, now includes 145 unique stations in Chicago, each represented with three key attributes. Table 3 provides a summary. Overall, insights from the three datasets provide foundational knowledge for building the recommender system.

OBJECTIVE AND METHODOLOGY

Objective

The main objective is to create a simple recommendation system using embeddings to compare the user's query with each listing's name and description. This approach allows users to find the most suitable Airbnb based on their preferences, enhancing the relevance and quality of search results. Some preliminary filters, such as headcount, amenities, or neighborhood, will be applied to narrow down the data for each query. By filtering first, the system can reduce unnecessary computation and focus on listings that meet basic user criteria. This layered approach helps ensure that users receive accurate and personalized recommendations without overwhelming them with irrelevant options. Additionally, the use of embeddings allows the system to capture nuanced similarities in language, improving the likelihood of finding listings that truly align with what users are seeking.

Methodology

For the methodology, various libraries and modules are utilized for data preprocessing. Sentiment analysis, a computational process for determining whether a piece of writing is positive, negative, or neutral, is conducted using VADER (Valence Aware Dictionary and sEntiment Reasoner). VADER is a lexicon and rule-based sentiment analysis tool that is specifically tuned for social media expressions. It creates a scoring system for the reviews dataset. Given VADER's design for social media data and its efficient runtime, it is well-suited for calculating average sentiment scores across all reviews. VADER is trained on a large volume of data, which enhances its ability to capture language patterns and understand text.

Since the dataset does not include ground-truth accuracy, the 'review_scores_accuracy' attribute in the listings data is heavily skewed toward positive sentiment, leading to class imbalance. Consequently, VADER's sentiment output will be used as a benchmark to cross-check with sentiment analysis output from pre-trained models. Additionally, a word cloud is generated to verify that VADER's output accurately captures words representing 'positive,' 'neutral,' and 'negative' sentiments. Further sentiment analysis will be conducted using Logistic Regression, SVC (Support Vector Classification), and Naive Bayes models. SMOTE (Synthetic Minority Oversampling Technique) and class_weight='balanced' will be applied to address class imbalance. A hierarchical clustering algorithm is used to group Airbnb listings by name and description, utilizing 50-dimensional GloVe embeddings of tokenized text with SpaCy. This clustering aims to categorize listings into three groups that represent typical characteristics of guest stays. Geodesic distance metrics are used to calculate the distance in miles between listings and subway stations.

Finally, a semantic-driven hybrid recommender system uses embeddings to suggest the top k Airbnb listings along with their nearest subway stations based on user preferences and give k. This system makes recommendations by comparing all feature attribute similarities and embeddings with user references and returns the top significant items based on descending Cosine similarity scores. This approach not only personalizes recommendations but also ensures that the selected listings are contextually relevant to the user's needs. Additionally, the integration of proximity to subway stations adds a valuable layer of convenience, making it easier for users to navigate the city during their stay.

SENTIMENT ANALYSIS

Within the listings data, the only available ground-truth customer review metric is the 'review_scores_ratings' attribute. However, this data is highly skewed: out of 7,952 listings, only 167 have a rating score below 4.0 on a scale of 1 to 5. Due to this imbalance, validation of the preliminary VADER output will be supported by a word cloud analysis. Additionally, the VADER output will serve as a benchmark. To further verify sentiment predictions, models such as Logistic Regression, Linear SVC, and Naive Bayes will be trained and tested to cross-check against VADER's results, with weighted adjustments to address class imbalance. Several preprocessing steps will be undertaken to prepare the data for sentiment analysis:

- Lemmatization will be applied to reduce words to their root forms. All three validate models will be trained and tested with both lemmatized and non-lemmatized text.
- Trigrams will be generated to capture contextual relationships between words.
- Common words like 'stay,' 'place,' 'chicago,' 'host,' 'room,' and 'airbnb' will be removed.

VADER model

To handle class imbalance, VADER's compound scores (ranging from -1 to +1) are adjusted with weighted thresholds given scores ≥ 0.5 are classified as 'positive,' scores < 0 as 'negative,' and scores in between as 'neutral.' The initial VADER output shows that out of 403,291 reviews, 370,198 are labeled 'positive' (91.8%), 26,449 as 'neutral' (6.55%), and 6,644 as 'negative' (1.65%). Additionally, word clouds generated for these three classes reveal tokens that align with expectations.

Positive Reviews

Containing typical phrases like "highly recommend," "great location," "everything needed," and "really enjoy." Additionally, less frequent phrases such as "come back," "quick respond," and "felt safe" also reflect positive feedback, Figure 1.

Figure 6. Word cloud for positive reviews

Neutral Reviews

Representing 6.55% of the data, include common Airbnb descriptors like "nice," "good," "thank," and "convenient." However, they also contain some negative terms like "problem" and "complaint." The term 'good' appears more prominently than “great,” which aligns with a more moderate level of satisfaction, Figure 2.

Figure 7. Word cloud for neutral reviews

Negative Reviews

Comprising 1.65% of the data, are marked by distinctly negative words such as "issue," "broken," and "dirty." Terms like "clean," "bed," and "shower" appear as well, potentially associated with negative adjectives. Notably, the majority of negative reviews are for apartment units, which is logical given that entire home/apartment rentals are dominant in the dataset, Figure 3.

Figure 8. Word cloud for negative reviews

Logistic regression

Logistic Regression model achieves an accuracy of 85.9%, result shown in Table 4, with the original text, which is moderate compared to other models. However, it has the longest runtime at 415.34 seconds, indicating a substantial computational cost. When lemmatization is applied, the accuracy improves to 89.6&, showing that lemmatization positively impacts performance by capturing more generalized word forms. Additionally, the runtime is significantly reduced to 263.81 seconds, making it more efficient. Overall, while Logistic Regression benefits from lemmatization, it still lags in accuracy compared to other models and has higher computational demands.

Linear SVC

Linear SVC demonstrates the highest accuracy among all models with the original text, at 92.45%, Table 4. Its runtime is also relatively low at 134.29 seconds, making it an efficient and highly accurate choice among three models. With lemmatization, the accuracy slightly decreases to 91.87%, and the runtime increases to 212.76 seconds. This suggests that lemmatization may introduce slight complexity in processing while not significantly enhancing accuracy for this model. Overall, the original version of Linear SVC is the best performer, balancing high accuracy with adequate runtime.

Naive Bayes

Naive Bayes model performs well with the original text, yielded an accuracy of 89.09%, Table 4, and having the shortest runtime among all models at 20.69 seconds. This combination of good accuracy and fast processing makes it suitable for real-time applications or large datasets. However, when lemmatization is applied, the accuracy decreases substantially to 73.009%, although the runtime improves slightly to 15.78 seconds. The significant drop in accuracy with lemmatization suggests that Naive Bayes relies heavily on the original form of words and loses predictive power when generalized forms are used. Therefore, Naive Bayes with original text is preferable, but it may still fall short in accuracy compared to Linear SVC for tasks requiring high precision.

Table 4: Sentiment Analysis Results

MODEL	SMOOTHING	ACCURACY	RUN TIME (seconds)
Logistic Regression	Original	0.859003	415.398643
Logistic Regression	Lemmatized	0.896180	263.814764
Linear SVC	Original	0.924488	134.288854
Linear SVC	Lemmatized	0.918694	212.765875
Naive Bayes	Original	0.890915	20.691782
Naive Bayes	Lemmatized	0.730932	15.788062

Sentiment output selection

Overall, some observations indicate that models using lemmatized comments have longer run times compared to models using the original comments. The accuracy of these models varies depending on their characteristics, ranging from a minimum of 73% to a maximum of 92%. Based on these results, Linear SVC on the original data demonstrates the best performance,

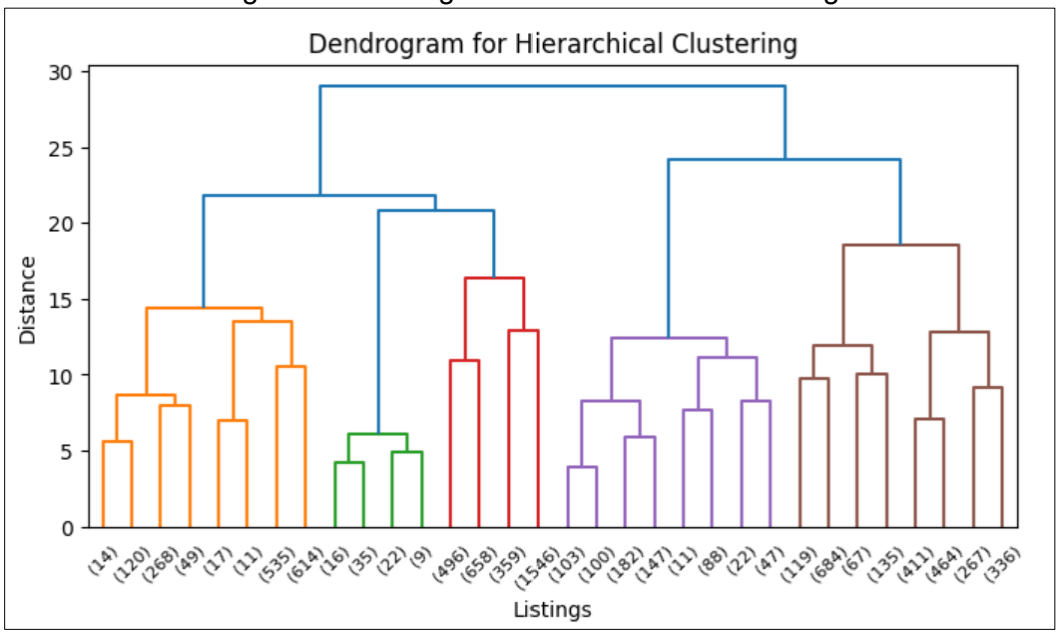
achieving 92% accuracy and appropriate runtime, making it the preferred choice for training on the entire dataset and comparison with the VADER model.

Using the entire training dataset, the output shows that out of 403,291 reviews, 378,720 are labeled as ‘positive’ (93.9%), 19,661 as ‘neutral’ (4.87%), and 4,910 as ‘negative’ (1.23%), highlighting an even more imbalanced class distribution. When comparing VADER and Linear SVC models, Linear SVC classifies the word “good” as positive sentiment, while VADER classifies it as neutral. Given the purpose of sentiment analysis and the typical meanings of these words, technically, 'great' should be classified as 'positive,' while 'good' should remain neutral. Thus, VADER’s output aligns more closely with these expectations and has a slightly better class distribution, making it the selected output for the next step.

LISTING CLUSTERING

For each listing, its name and description are combined into a single text entry. The objective of this merging is to group listings that share common words and characteristics, placing similar listings together based on their descriptions. This approach aims to create clusters that reflect the unique attributes of each listing type. Prior to clustering, several pre-processing steps were applied to the text data to ensure accuracy and meaningful results. First, each text was tokenized using SpaCy, employing its default set of stop words, with an additional filter to only retain tokens longer than two characters, this helps ensure that only full English words are considered. Next, the text was lemmatized using SpaCy to reduce each word to its root form. Common, non-informative words were removed to highlight distinguishing characteristics, such as “chicago,” “nan,” “room,” “home,” “place,” “neighborhood,” “bedroom,” “airbnb,” “locate,” “miles,” “mile,” “apartment,” “walk,” “park,” “loop,” “city,” “downtown,” “restaurant,” “minute,” “line,” “stay,” “area,” “offer,” “away,” “street,” “location,” “michigan,” “min,” “close,” and “block.” Removing these words allows the clustering process to better capture the unique features of each listing type.

Figure 9. Dendrogram for hierarchical clustering

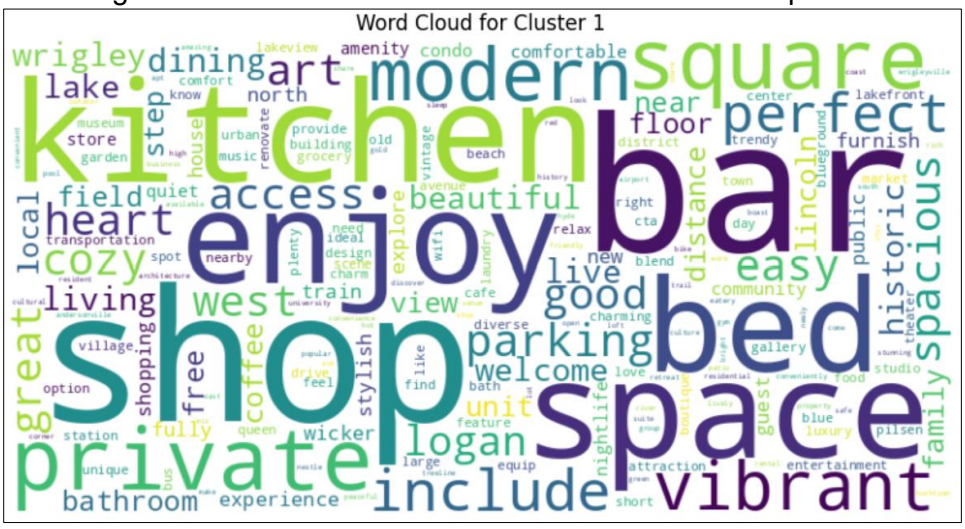


For each listing description, a 50-dimensional mean embedding was generated to capture the semantic information of the text. Clustering was then performed based on the similarity of these embeddings, using Euclidean distance as a measure of closeness between listings. The primary clustering method used was Hierarchical Clustering. While KMeans and DBSCAN (Density-Based Spatial Clustering of Applications with Noise) were also considered, these methods did not yield meaningful or interpretable results for this dataset. Figure 9 shows a dendrogram was used to visualize the distribution of clusters. The dendrogram initially suggested that 4 clusters might be optimal. However, after examining the descriptive characteristics, it was determined that 3 clusters provided a more interpretable outputs, which was further validated using word clouds. Additionally, BERTtopic was explored to identify the dominant topic within each cluster. However, it did not produce meaningful or coherent suggestions for this dataset. The final three clusters and their descriptions are as follows:

Cluster 1 - "Vibrant Social Spaces"

44,769 listings – This cluster is characterized by listings that emphasize social spaces and modern amenities, appealing to those interested in lively environments, crowded neighborhoods, and bustling areas with bars and shops, making them attractive to travelers seeking convenience and entertainment. As shown in Figure 10, The top 10 words associated with this cluster are “bar,” “shop,” “kitchen,” “space,” “bed,” “enjoy,” “private,” “square,” “modern,” and “vibrant”.

Figure 10. Word cloud for cluster 1 – “Vibrant Social Spaces”



Cluster 2 - "Cultural and Scenic Attractions"

2,483 listings – Listings in this cluster are located near cultural sites and scenic landmarks, including museums and rivers. These accommodations often emphasize their accessibility to key attractions, offering an excellent base for tourists interested in exploring Chicago. This cluster is ideal for tourists seeking proximity to attractions and opportunities for sightseeing. The top 10 words in this cluster are “museum,” “north,” “south,” “river,” “field,” “lake,” “center,” “attraction,” “space,” and “avenue,” as shown in Figure 11.

Cluster 3 - "Accessible with Public Transportation"

700 listings – Highlight convenient access to public transportation options, ideal for travelers who prioritize ease of transit. These listings often appeal to guests looking for hassle-free travel experiences, with features like nearby bus stops, train stations, and accessible parking. The top 10 words for this cluster include “train,” “private,” “public,” “parking,” “great,” “easy,” “free,” “transportation,” “bus,” and “bar”, shown in Figure 12.

Figure 11. Word cloud for cluster 2 – “Cultural and Scenic Attractions”

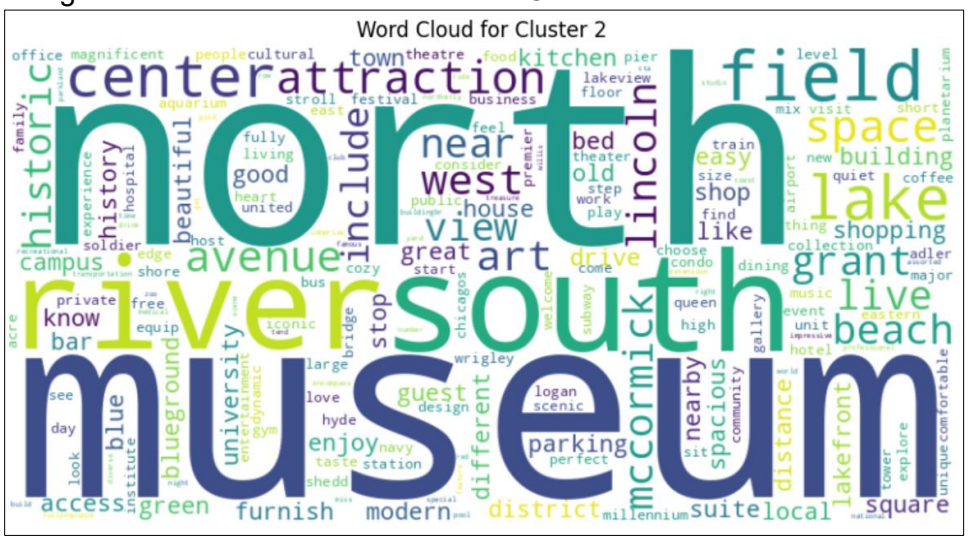
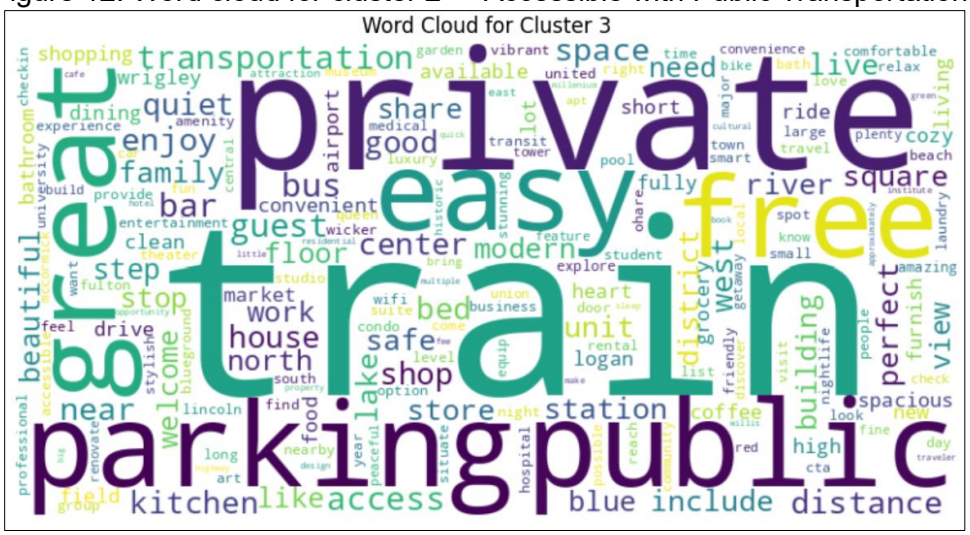


Figure 12. Word cloud for cluster 2 – “Accessible with Public Transportation”



DATASET CONSOLIDATION

To create a foundation database for the system, the final dataset integrates sentiment scores from reviews with listing data and CTA data by merging the listings, reviews, and CTA datasets. This combined dataset provides comprehensive information for each of the 7,952 Airbnb listings, including the average sentiment score, cluster group name, and nearest train station(s).

Aggregate sentiment score

The first step in constructing the dataset is to combine the listings and reviews data. Since each listing is associated with multiple reviews, it is essential to account for the effect of time on sentiment scores to ensure that recent reviews carry more weight. Therefore, the overall sentiment score for each listing is calculated using a weighted average approach, considering scores from both the VADER and Linear SVC sentiment analysis models. For each listing, every individual review's sentiment score is multiplied by a weight determined by its recency, thereby emphasizing recent feedback more strongly. Once weighted, these scores are summed and then divided by the total weight to produce a single, comprehensive sentiment score for the listing. The weights are structured to prioritize more recent reviews, as these reflect the most current experiences of guests. Specifically, reviews within the last 90 days are assigned the highest weight of 1.0, reviews from 91 to 365 days are weighted at 0.9, and reviews older than 365 days are assigned a weight of 0.8. This weighted approach allows the final sentiment score to capture recent trends and changes in guest satisfaction while still accounting for the overall review history:

- Reviews within the last 90 days: Weight = 1.0
- Reviews from 91 to 365 days: Weight = 0.9
- Reviews older than 365 days: Weight = 0.8

The formula for calculating the weighted average sentiment score is:

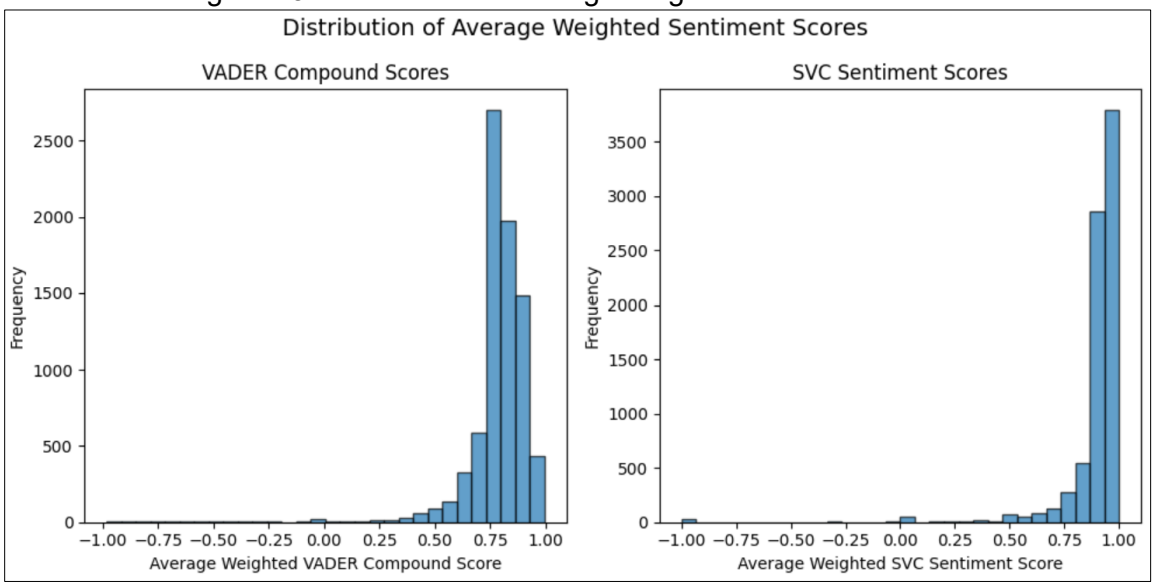
$$\text{Weighted Average Sentiment} = \frac{\sum(\text{Sentiment Score} \times \text{Weight})}{\sum(\text{Weight})} \quad (1)$$

where:

- Weighted Average Sentiment: The overall sentiment score for a listing based on its reviews.
- Sentiment Score: The sentiment score from each individual review.
- Weight: The assigned weight based on the review's recency.

To determine the optimal sentiment score between VADER and Linear SVC, the distribution of average weighted sentiment scores for each model was analyzed in depth. Figure 13 reveals that both VADER and Linear SVC distributions are significantly skewed toward positive sentiment, indicating that the majority of listings have high average scores, often close to 1. This skew toward positive scores suggests that most guest reviews express satisfaction with their stays. However, some key differences emerged between the two models. VADER scores exhibit a wider range of values, with a portion of listings receiving scores closer to zero and even slightly negative, indicating that VADER may be more sensitive to subtle variations in sentiment. This broader range can be valuable in distinguishing listings with mixed reviews or capturing occasional negative feedback. In contrast, the Linear SVC scores are tightly clustered near the high end, with a higher concentration of scores close to 1 and fewer listings scoring near zero or below. This tight clustering suggests that Linear SVC may lean toward a more positive classification, potentially classifying sentiment more conservatively or failing to capture nuanced sentiments as effectively as VADER. Given these findings, VADER's ability to capture a broader spectrum of sentiment, encompassing both positive and less favorable feedback, made it the preferred model for the recommender system. By selecting VADER, the recommender can reflect a more nuanced picture of guest satisfaction, which may be beneficial for users seeking listings with consistently high reviews.

Figure 13. Distribution of average weighted sentiment scores



Missing values

Additionally, there are 1,476 listings that do not have any reviews. To maintain consistency across the dataset, the sentiment scores for these listings will be replaced by the mean sentiment score of listings with available sentiment data. This approach ensures that listings without reviews do not unduly impact the analysis or recommendations. Other attributes with missing values in the dataset are handled with specific imputation strategies to ensure data completeness:

- 'host_is_superhost': Randomly assigned as either true or false with a 50-50 distribution, reflecting the balanced distribution in the existing data.
- 'bathrooms', 'bedrooms', 'beds': Missing values are replaced by the average, rounded down, to avoid overestimating accommodations and ensure that guests have enough space.
- 'price': Replaced with the average price to provide a reasonable baseline for listings without a specified price.

Nearest train station(s)

The second step is to combine the listings and CTA data. For each train station, the distance to each listing is calculated using the geodesic module from geopy, measuring the distance in kilometers (km). Train stations considered "nearby" are those within a radius of 0.8 km, which is approximately a 10-minute walk. This allows users to easily identify listings that offer convenient access to public transportation. The nearest train station data for each listing is stored as a list, where each element is a tuple containing two nested tuples. The first nested tuple includes the train station name and its corresponding line color(s), and the second nested tuple provides the train station's geographic coordinates (longitude and latitude).
This structure ensures that each listing can reference multiple nearby train stations if available, offering a more complete view of transit options. The purpose of this step is to enhance the recommender system by providing geospatial mapping that highlights each suggested listing along with its corresponding nearest train stations. This feature is valuable for users who prioritize proximity to public transportation. It is important to note, however, that some listings in

the dataset do not have any train stations within the 0.8 km radius and, therefore, will not display nearby transit options.

After joining the three datasets, the final dataset now includes 7,952 Airbnb listings in Chicago, each represented with 19 carefully selected attributes, Table 5. These key attributes were meticulously chosen through various approaches and processes to ensure that each listing is accurately and comprehensively represented. The attributes encompass basic filtering information, such as guest capacity and essential amenities, as well as outputs from previous steps, including the overall sentiment score and the nearest train station(s) location. Additionally, embeddings of each listing's name and description have been created to facilitate cross-referencing with user queries. This robust feature set not only provides essential listing details but also offers enhanced insights into customer satisfaction and accessibility to transit. Overall, the dataset enables a highly personalized and user-centric recommendation experience for travelers exploring Chicago's Airbnb options.

Table 5: Final Dataset Schema

ATTRIBUTE	DESCRIPTION	DATA TYPE
id	Listing ID	Numerical
accommodates	Maximum capacity of the listing	
avgWeighted_VADER	Overall VADER sentiment score	
bathrooms	Number of bathroom(s)	
beds	Number of bed(s)	
bedrooms	Number of bedroom(s)	
embedding	Mean embedding of 'name_description'	
latitude	Latitude of the listing	
longitude	Longitude of the listing	
price	Rental price	
amenities	Available amenities	Categorical
cluster_name	Name of the cluster the listing belongs to	
name_description	Joint data of description and name	
neighbourhood_cleansed	Chicago neighborhoods	
tokens	List of tokens from 'name_description'	
train_station	Information of nearest train station (s)	
room_type	Type of rental	
host_is_superhost	Whether host has super host status	Boolean
host_identity_verified	Whether host is verified	

RECOMMENDER SYSTEM

The recommender system is designed to recommend Airbnb listings based on user input, such as preferences for accommodations, price range, and desired amenities. It combines filtering, sentiment analysis, and embeddings to provide relevant and personalized results. The system

integrates geographic information and allows for detailed recommendations with interactive maps. There are five main steps.

Step 1 - User input and preferences

The system begins by gathering user preferences through a guided interface to personalize the recommendations. Users are prompted to specify essential criteria such as the minimum number of guests a listing should accommodate, the maximum price they are willing to pay, and the minimum number of bathrooms, bedrooms, and beds required. These inputs ensure the system filters listings that meet the basic logistical needs of the user. Additionally, users can specify whether they prefer listings hosted by verified hosts or super hosts, adding an extra layer of trustworthiness to the recommendations. To tailor the results even further, users can list preferred neighborhoods and amenities, which allow the system to narrow down the options based on specific geographical and functional preferences.

A unique feature of the system is its ability to incorporate qualitative user input through a brief comment or description of their ideal stay, such as "fun and sunny" or "quiet and peaceful." This input is processed using a pre-trained word embedding model, such as GloVe, which converts the text into a numerical representation. These embeddings capture the contextual meaning of the user's preferences, enabling the system to align them with the descriptions of available listings. This combination of structured inputs and free-form textual preferences ensures a personalized experience, as the system matches not only the functional requirements but also the emotional or experiential expectations of the user.

Step 2 - Filtering listings

Once user preferences are collected, the system applies a series of filters to narrow down the listings to those that best match the specified criteria. The first set of filters focuses on numerical and categorical values, such as ensuring that listings meet the minimum number of bathrooms, bedrooms, and beds, as well as staying within the maximum price range. Boolean filters, like requiring verified hosts or super hosts, are also applied to include only listings that align with these trust-related preferences. Neighborhood filters further refine the results by matching listings located in the user's desired areas. Similarly, amenities filters ensure that listings provide the facilities specified by the user, such as Wi-Fi, air conditioning, or parking.

To incorporate qualitative user preferences, the system evaluates the semantic similarity between the user's comment embedding and the embeddings of listing descriptions. For each listing, the cosine similarity between these embeddings is calculated, resulting in a similarity score. This score captures how well a listing aligns with the user's personal preferences or desired experience, as expressed in their comment. The listings are then sorted by their similarity scores, ensuring that the most contextually relevant options are prioritized. This approach allows the system to balance functional filtering with a deeper understanding of the user's qualitative needs, leading to a more tailored and meaningful recommendation set.

Step 3 - Enhancing rankings

After filtering listings based on user preferences, the system ranks them to highlight the most suitable options. This ranking process combines two key metrics: the sentiment score and the similarity score. The sentiment score is derived from the average sentiment of reviews for each listing, providing an indicator of overall customer satisfaction. Listings with consistently positive

reviews receive higher sentiment scores, ensuring that user feedback plays a crucial role in the evaluation process. The similarity score, on the other hand, measures the alignment between the user's comment embedding and the listing description embedding, reflecting how well the listing matches the user's qualitative preferences. By including both scores, the system ensures that listings are not only functionally suitable but also align with the user's expectations and experiences.

To create a unified ranking, the system normalizes both scores to a common scale (0 to 1) for comparability. It then combines them using weighted averaging, where the similarity score is typically given more importance - 60%, to prioritize alignment with user preferences, while the sentiment score contributes the remaining weight - 40%, to ensure quality assurance through past reviews. This weighted combination creates a "combined score" for each listing, which determines its final rank. Listings are then sorted by this combined score in descending order, ensuring the most relevant and highly rated options are presented first. This multi-faceted ranking process provides a balanced and reliable approach to recommending listings, accounting for both user-specific needs and general listing quality.

Step 4 - Personalized recommendations

Once the listings are ranked, the system identifies the top-k recommendations tailored to the user's needs. Users can specify the number of listings they would like to see or choose to view all matching options. To enhance personalization, the system organizes the recommendations into clusters based on characteristics such as "Cultural and Scenic Attractions," "Vibrant Social Spaces," or "Accessible with Public Transportation." These clusters help categorize listings by their distinctive features, making it easier for users to find options that align with their specific preferences or interests. The clustering also adds diversity to the recommendations, ensuring that users receive a variety of relevant choices.

If users select multiple clusters, the system divides the requested number of listings among the chosen categories, distributing them as evenly as possible. It prioritizes the highest-ranked listings within each cluster but can also shuffle clusters to introduce some variability. In cases where a cluster does not have enough listings to meet the desired count, the system supplements the recommendations with top listings from the overall ranked dataset. This approach ensures that users receive a well-rounded selection of listings that not only match their criteria but also provide unique options tailored to their interests and context. The final result is a curated list of high-quality recommendations that balance user preferences with the diversity of the listings available.

Step 5 – Geospatial map visualization

To provide an interactive and user-friendly experience, the system visualizes the top-k recommendations on a geographic map. Each recommended listing is marked on the map with details such as its price, neighborhood, and nearby train stations, allowing users to explore the location and accessibility of each property. The map includes distinct icons to differentiate between listings and train stations, making it easy to understand the proximity of transportation options. Additionally, users can interact with the map by clicking on markers to view pop-ups containing detailed information about the listings. This visual representation enhances the recommendations by adding a spatial perspective, helping users make informed decisions based on the geographic context of their preferred listings.

DISCUSSION AND CONCLUSIONS

Overall, the developed Airbnb recommender system demonstrates a practical and impactful approach to tailoring travel accommodations to individual preferences. By combining semantic embeddings, sentiment analysis, and user-specific filters, the system successfully enhances the relevance and quality of recommendations. Its focus on usability, including features like neighborhood and amenity preferences and interactive geospatial maps, provides users with actionable insights into their options. The system's clustering mechanism also introduces a level of diversity, ensuring users receive recommendations across various themes and characteristics. Overall, this solution adds value to travelers seeking personalized accommodations in a popular destination like Chicago, particularly those prioritizing convenient public transportation access.

Furthermore, the innovation behind this system lies in its hybrid approach, which integrates advanced natural language processing techniques with user preferences. The use of sentiment analysis validates the feedback from previous guests, further improving the reliability of the recommendations. This dual emphasis on user input and data-driven insights highlights the system’s adaptability and user-centered design, making it a robust tool for enhancing the Airbnb experience.

Discussion

Despite its strengths, the system's performance could face challenges when scaling to larger datasets or more complex user requirements. For example, the reliance on embeddings for comment-based filtering may yield less accurate results if the user's input is vague or overly general. Similarly, the clustering mechanism, while beneficial, might need refinement to ensure an even distribution of recommendations across diverse user preferences. Another potential limitation is the system's dependency on pre-defined sentiment analysis models, which may not account for nuanced or context-specific guest feedback. Addressing these limitations could further enhance the accuracy and usability of the system, especially as user needs evolve.

Future recommendation

For future improvement, the system could benefit from incorporating dynamic weighting for the ranking process, allowing users to adjust the importance of sentiment scores and similarity measures based on their priorities. Additionally, integrating more advanced machine learning models, such as transformer-based architectures, could improve the handling of complex user comments and ensure more accurate embedding calculations. Expanding the system to include additional features, such as personalized travel itineraries or price forecasting, would also increase its value. Lastly, leveraging real-time data updates and user feedback loops could refine recommendations over time, ensuring the system remains responsive to changing market trends and user preferences.

AUTHOR CONTRIBUTIONS

The author, Mai Ngo], independently performed all tasks of this project. This included data gathering, cleaning and preparation, developing the machine learning algorithms, conducting exploratory data analysis, and implementing clustering and recommendation methodologies. Additionally, the author created visualizations, processed textual and geospatial data, and

integrated sentiment analysis into the system. All writing, including documentation and report creation, was also completed solely by the author.

REFERENCES

Abbasi-Moud, Z., Vahdat-Nejad, H., & Sadri, J. (2021). Tourism recommendation system based on semantic clustering and sentiment analysis. Expert Systems with Applications, 167, 114324.

City of Chicago. (2018, July 11.). Boundaries - neighborhoods. City of Chicago Data Portal. https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Neighborhoods/bbvz-uum9

City of Chicago. (2024, August 2). CTA system information: List of 'L' stops. City of Chicago Data Portal. https://data.cityofchicago.org/Transportation/CTA-System-Information-List-of-L-Stops/8pix-ypme/about_data

Inside Airbnb. (2024, June 21). Get the data. Inside Airbnb. https://insideairbnb.com/get-the-data/

Ngo, Mai. (2024, November 11). Chicago Airbnb listings map. GitHub Pages. https://pngo1997.github.io/Chicago-Airbnb-Listings/chicago_all_listings_map.html

Ngo, Mai. (2024, November 11). Chicago Airbnb CTA map. GitHub Pages. https://pngo1997.github.io/Chicago-Airbnb-CTA/chicago_train_stations_map.html

Ray, B., Garain, A., & Sarkar, R. (2021). An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews. Applied Soft Computing, 98, 106935