GRADUATE CERTIFICATE
INTELLIGENT REASONING SYSTEM (IRS)

Project Report

InnJoy

A chatbot-driven system for real-time personalized BnB recommendations

GROUP 18:

Tao Xu (A0285941U)
Wei Chuanjie (A0285709N)
Yan Zihan (A0285706W)
Zhang Yaoxi (A0285851U)

29-Oct-2023

Table of Content

1 Executive Summary

In today's era, the applications of deep learning and large language models are not merely confined to theoretical studies. Their practical applications have been widely demonstrated across various industries. Our team is dedicated to exploring these advanced technologies' real-world applications in the online BnB booking domain. As an indispensable service in daily life, users are increasingly demanding a more personalized and efficient recommendation system for online BnB platforms. Addressing this need, after extensive research and experimentation, our team developed "InnJoy—A chatbot-driven system for real-time personalized BnB recommendations."

Unlike traditional booking platforms, InnJoy strives to seamlessly integrate a chatbot powered by large language models with advanced neural collaborative filtering recommendation algorithms. This combination offers real-time and highly accurate BnB recommendations. Users no longer have to invest extensive time in intricate searches and filters. Instead, they interact with the chatbot, allowing InnJoy's system to swiftly and comprehensively understand users' immediate needs and emotional states, subsequently delivering the most suitable BnB choices. The system's core comprises two main components: firstly, a chatbot capable of instantaneously recognizing and deeply understanding users' needs and emotions; secondly, an efficient BnB recommendation engine based on neural collaborative filtering technology. Through real-time dialogues with users, the chatbot poses a series of precise questions to accurately capture users' preferences and requirements. In the background, based on the user's feedback, the recommendation system curates a personalized list of BnB suggestions. These recommendations are instantaneously displayed on the user's front-end interface, facilitating users to select their desired BnB and providing direct links to individual BnB websites or other major booking platforms.

Product Features:

1. Efficient Interaction: Through our front-end web-based chatbot, we effectively engage with users, capturing their needs and emotional states through a series of precise questions.
2. Real-time Recommendations: In the backend, the recommendation algorithm processes the information provided by users instantly, refreshing the list of recommended BnBs, including their names, prices, and booking links.
3. Highly Personalized: With the support of a large language model, our chatbot delves deep into users' intentions, offering more tailored questions and recommendations.
4. Clear Business Model: We do not directly facilitate BnB bookings. Instead, we provide redirection links to major booking platforms, earning commissions from property listings, and strategically positioning ourselves to compete differently with major platforms.

To provide you with a comprehensive understanding of InnJoy's functionality and user experience, this report includes two videos: one detailing the front-end user interaction process and another elucidating the backend algorithm. The written portion delves into the technical details and model training processes.
We sincerely hope you'll take a deeper look at InnJoy, confident in its potential to revolutionize the online BnB booking landscape.

2 Business Justification

1. The Significance of the Industry

The BnB and online accommodation booking industry represents a substantial and ever-growing market, characterized by an increasing number of global travelers seeking unique and personalized lodging experiences. Delving deep into this industry not only provides access to a lucrative market segment but also positions our product to cater to the evolving needs of tech-savvy consumers. The rising demand for swift, personalized, and efficient booking systems underscores the value proposition and potential success of InnJoy in this vast industry.

2. In-depth Explanation of Business Model

Our primary revenue model hinges on affiliate marketing. By serving as a gateway to major booking platforms, every redirection from InnJoy to these platforms that results in a booking generates commission for us. This approach ensures we remain impartial in our recommendations, focusing solely on user preferences and not on direct sales. Furthermore, as our platform grows and garners a loyal user base, there are potential opportunities to collaborate with BnB providers for sponsored listings or premium placements, thereby diversifying our revenue streams.

3. Extending to Broader E-commerce Domains

The core technology underpinning InnJoy is not exclusive to the BnB industry. The real-time, chatbot-driven recommendation system can be adapted and scaled to cater to various other e-commerce domains. For instance, it can be integrated into online retail for personalized product suggestions, travel platforms for tailored trip itineraries, or even food delivery services for real-time meal recommendations based on a user's current craving.

4. Multi-modal Input Expansion

The chatbot functionality in InnJoy isn’t restricted to text-based interactions. In the future, leveraging advancements in Natural Language Processing (NLP) and Computer Vision, the system can be enhanced to accept voice and video inputs. By doing so, InnJoy could seamlessly transform into a feature within voice assistants or home management robots, providing real-time recommendations not just through text but also through engaging auditory or visual interactions.

5. Future Expansion and Potential

As data collection and machine learning models become more sophisticated, InnJoy has the potential to evolve into a comprehensive lifestyle assistant. By integrating with smart home systems, wearable tech, and other IoT devices, it could anticipate users' needs and provide proactive recommendations across various domains, from entertainment and dining to fitness and leisure activities.

By grounding our approach in cutting-edge technology and continuously expanding to encompass the diverse needs of the modern consumer, InnJoy is poised not just to revolutionize the BnB booking industry but also to become an integral part of the future digital ecosystem.

3 Project Team

3.1 Project Objective

The main objectives of ours system is to develop a real-time personalized BnB recommendation system. Via using a chatbot to perform a interesting conversation with users, the system can capture the users preference and give a accurate recommendation.

3.2 Team Members

Full Name	**Work Items (Who Did What)**
Tao Xu	1. Feature selection: Participated in the definition and selection of features to ensure that our model could accurately interpret and cater to the users' needs and preferences. 2. Language model research: Responsible for language model research and deployment. 3. Chatbot development: Development of the chatbot' was a major part of my work, which included question identification and natural language processing (NLP)
Wei Chuanjie	1. Conceptualization and Ideation: Played a pivotal role in formulating and envisioning the overarching concepts and direction for the entire project. 2. Model Development and Recommendation Strategies: Spearheaded the development of the Neural Collaborative Filtering model, focusing on the integration and optimization of three core recommendation methodologies. 3. Team management and Project Coordination: Managed teams, ensuring alignment with project objectives, and was instrumental in setting and adhering to project timelines and schedules.
Yan Zihan	1. Data Acquisition & Processing: Responsible for the whole process from data acquisition to data processing, including data cleaning, target major data, process text information and new features generation. 2. Recommendation Strategy and Score Generation: Focusing on the general recommendation, mainly new users' recommendation. Use random forest to identify features and generate scores. 3. Project Management & Design and Troubleshooting: Help design the user interface. Communicate with team members and coordinate project progress. Final product testing and troubleshooting, summarize project results and make videos.

Zhang Yaoxi	1. User Interaction Design and Frontend Development: Conceptualized user interaction strategies and designed interfaces using HTML, CSS, and JavaScript. 2. Backend Development, System Integration, and Database Management: Developed backend services for seamless communication, optimized database structures, and implemented data processing techniques to enhance recommendation accuracy. 3. Project Management and Coordination: Managed team efforts, ensured project alignment, and coordinated timelines for effective collaboration among team members.

4 Project Solution

4.1 Project Deliverables

4.1.1 Application Features

Upon acquiring our data through Airbnb, our team conducted internal consultations and algorithm evaluations to filter out some information we deemed unimportant, such as transaction time, transaction IDs, among others. Although the transaction time could reflect when the transaction occurred, it didn't provide much valuable information in the context and objectives of our project. We were more interested in other aspects of the data, like user reviews, listing information, and pricing, etc. Hence, we decided to exclude the transaction time data point from our analysis.

Host IDs primarily serve to identify the uniqueness of host, yet they don't directly aid in analyzing the value of listings or user satisfaction in our context. Instead, we utilized the listing IDs as unique identifiers.

After this filtration process, we are prepared to train our recommendation algorithm using the following features. From the interpretation of these features, it is apparent that this dataset is more focused on describing the information of the listings (items), while lacking significantly in user information, which is considerably challenging to acquire.The refinement of our dataset by excluding certain irrelevant data points is an essential step towards creating a more accurate and efficient recommendation algorithm. The more precise representation of listings' information, as opposed to transaction-related details, aligns better with our project's objectives, focusing more on the listings and less on the transaction or user aspects. This refined dataset is expected to significantly improve the performance and accuracy of our recommendation system, by focusing on the most relevant features related to the listings and user reviews, which are crucial for achieving a higher level of user satisfaction and better recommendations.

	VARIABLE NAME	DESCRIPTION
INPUT	listing_id	BNB ID
	reviewer_id	User ID
	price	Price of BNB
	number_of_reviews	Number of reviews for the BNB
	review_scores_rating	Total rating of the BNB
	host_is_superhost	Whether the host is a superhost
	geographical_location	Location of the BNB
	purpose	Purpose of the trip
	Number_of_people	Number of people in the user's group
	surroundings	Surrounding environment
	transportation	Convenience of transportation
	room_type	Type of room
OUTPUT	scores	User's rating for the BNB

Recommendation Function 1: For returning users with IDs, provide more personalized recommendations based on the new input features from the user (new input features can also be left blank)

This recommendation feature is designed for recommending listings to returning customers. It accepts user ID, new input features, a pre-trained deep learning model, the original dataset, and the number of top recommendations to return as parameters. Below is a detailed explanation of this recommendation feature:

1. Retrieving User Historical Data:

- Initially, by using the encoded user ID (`encoded_user_id`), historical data of the user is filtered out from the original dataset. This historical data includes the listings that the user has previously reviewed.

2. Checking User Existence:

- Check if any historical data related to the given user is found. If no historical data is found, it indicates that the user has not left any reviews in the system, hence personalized recommendations cannot be provided.

3. Retrieving Unreviewed Listings:

- Filter out the listings from the original dataset that the user has not reviewed yet to provide new recommendation options. These unreviewed listings are potential targets for recommendations.

4. Setting New Features:

- Apply the new features (`new_features`) provided by the user to these unreviewed listings to simulate the user's interest in these listings. This way, we can predict the possible ratings they might give to these listings based on the new features provided.

5. Creating Model Input:

- Create input data for the deep learning model, including user ID, listing IDs, and features related to the listings. These data will be used for model predictions.

6. Making Predictions Using the Model:

- Utilize the pre-trained deep learning model to predict the ratings the user might give to the unreviewed listings. The model will consider the user's historical ratings, new features, and features of the listings to generate these ratings.

7. Sorting Predicted Scores:

- Sort the predicted scores in descending order to find the listings with the highest scores. These listings are considered to be the most suitable recommendations for the user.

8. Converting Encoded IDs to Original IDs:

- Finally, convert the encoded listing IDs back to the original listing IDs to return the final recommendation list to the user.

Through this feature, the system can provide personalized listing recommendations to returning customers based on their historical ratings, new input features, and the pre-trained deep learning model, thereby enhancing user satisfaction and experience.

Recommendation Function 2: For new users without IDs, find the most similar users through new features, and recommend listings that similar users have booked or might book (based on user collaborative filtering)

This recommendation feature is an algorithm aimed at executing recommendation actions for new users. It leverages the feature information provided by new users, employing collaborative filtering methods to identify the most similar existing users to the new user, and based on the historical behavior and model predictions of these similar users, it provides listing recommendations to the new user. Below is a detailed explanation of this recommendation feature:

1. Create New User Data:

- Initially, create a DataFrame with the new features (`new_features`) provided by the new user for subsequent recommendation actions.

2. Preprocess New User Data:

- Preprocess the new user data, including standardizing continuous features, one-hot encoding categorical features, and ensuring the data format is consistent with the original dataset for compatibility with the model.

3. Compute Similarity Between New User and Existing Users:

- Utilize Cosine Similarity to compute the similarity between the new user and existing users. This similarity measures the closeness between the new user and existing users in the feature space.

4. Identify the Most Similar User:

- From the similarity matrix, identify the most similar existing user to the new user. This most similar user will serve as the basis for recommendations.

5. Retrieve Historical Behavior of Most Similar User:

- Filter out the historical data of the most similar user from the original dataset, including the listings that this user has previously booked. This historical data will be used to recommend listings that the most similar user has booked.

6. Identify Listings Likely Preferred by Most Similar User:

- Filter out listings that the new user has not seen yet, which will be potential targets for recommendations. These listings will be scored based on the historical behavior and model predictions for the most similar user.

7. Model Prediction:

- Use the pre-trained deep learning model to predict scores for listings that the new user hasn't seen yet. The model will consider the historical behavior of the most similar user and the new user's features to generate these scores.

8. Sort and Recommend:

- Sort the listings with high predicted scores in descending order to find the listings that might be the most suitable for the new user. These listings are considered as the targets for recommendations.

Through this feature, the system can provide personalized listing recommendations to new users based on their provided feature information, similarity with the most similar existing user, historical behavior of the most similar user, and model predictions, thereby enhancing the experience and satisfaction of new users.

Recommendation Function 3: Based on the new feature information provided by users, recommend listings that are generally more popular and well-liked by a majority of people.

This recommendation feature is engineered to propose listings to users based on the new feature information they provide, with a focus on popular listings that have been well-received by a broad spectrum of people. This ensures a level of general appeal in the recommendations, which is particularly useful for users who are new to the platform or unsure of their preferences. Below is a detailed elucidation of this recommendation feature:

1. Collection of New Feature Information:

- Initially, collect and organize the new feature information (`new_features`) provided by the user into a structured format, such as a DataFrame, which will be utilized in the forthcoming steps of the recommendation process.

2. Preprocessing of New Feature Information:

- Conduct preprocessing on the new feature information to ensure consistency and compatibility with the dataset and the model. This may include tasks such as standardization of continuous features, one-hot encoding of categorical features, and handling of missing values if any.

3. Identification of Popular Listings:

- Identify listings that have been highly rated and frequently booked by a wide range of users in the original dataset. These listings are considered to be popular and are expected to have a broad appeal.

4. Model Prediction:

- Employ a NCF model to predict the likelihood that the user will appreciate the popular listings based on the provided new feature information and the features of the popular listings.

7. Sorting and Recommendation:

- Sort the popular listings based on the prediction scores in descending order to identify the listings with the highest likelihood of being appreciated by the user. Present these listings as recommendations to the user.

8. Presentation of Recommendations:

- Finally, present the sorted list of recommended popular listings to the user, providing them with options that are not only tailored to their provided preferences but also have a general appeal to a wide audience.

Through this feature, the system efficiently combines user-specific feature information with general popularity metrics to provide recommendations that cater to individual user preferences while also ensuring a broader appeal. This approach is anticipated to enhance user satisfaction, especially for those who are relatively new to the platform or have broad or undefined preferences, thereby promoting a more enjoyable user experience.

Summary of Recommendation Algorithms:

Function 1: Personalized Recommendations based on New Input Features from Users

- Use of Recommendation Technique: This feature employs a hybrid recommendation technique, blending collaborative filtering with content-based recommendations. It utilizes a collaborative filtering model to predict user ratings while also considering the new features provided by the user, hence amalgamating the advantages of different recommendation techniques.

- Rationality: This feature rationally incorporates the new features provided by users into the recommendation process to offer personalized suggestions. The information provided by users can influence the recommendation outcomes, aligning them more with the users' interests and needs.

- Coverage: Feature 1 performs well in terms of coverage, catering to both returning and new users. It can handle the scenario of new users, providing recommendations even if they do not provide new features.

Function 2: Recommendations Based on User Similarity

- Use of Recommendation Technique: This feature employs collaborative filtering technique, specifically user-based collaborative filtering. It recommends by computing the similarity between new users and existing users, thus adopting the collaborative filtering method.

- Rationality: Feature 2 rationally considers the similarity between new users and existing users, recommending listings to new users that are akin to those liked by similar users, which aligns with the fundamental notion of collaborative filtering.

- Coverage: The coverage of Feature 2 is lower, primarily catering to new users as its main objective is to provide personalized recommendations for new users, and it is not quite applicable to returning users.

Function 3: Recommendations Based on Weighted Popularity

This recommendation feature is an algorithm suitable for everyone. It is based on various characteristics of lodgings and utilizes a random forest to calculate the weight of each feature. Through iterative selection and calculation, it identifies the most representative features of lodgings, which are then weighted to obtain the final lodging popularity score. Based on this score, it provides reasonable recommendations for anyone seeking more popular lodgings. Here is a detailed explanation of this recommendation feature:

- **Acquiring BnB Features:**

Initially, the numerous features of the lodgings are analyzed, excluding features with a significant amount of missing data and extracting the features for processing.

- **Weight Calculation using Random Forest:**

The selected features undergo weight calculation through a random forest. After adjusting various hyperparameters, the most effective set is chosen as the weights, while features with no weight (i.e., those not affecting the final score) are eliminated.

- **Reanalysis of Features:**

The features with assigned weights undergo a secondary analysis. Subjective features, like price, are removed. Although the model assigns a relatively large weight to price, as the price does not reflect an objective feature of the lodging itself, it cannot simply determine the inherent attributes of a lodging based on price alone. Thus, it is removed.

- **Calculating the Final Score:**

The obtained weight information is combined with the feature set to compute the final popularity score, creating a complete list of popular lodgings after sorting.

- **BnB Recommendations:**

A subset of highly ranked lodgings from different room types is selected, ensuring a diverse recommendation. A proportionate selection is made from each type, offering randomized recommendations to ensure both reliability and a varied selection for users.

This approach is anticipated to enhance user satisfaction, especially for those who are relatively new to the platform or have broad or undefined preferences, thereby promoting a more enjoyable user experience.

4.1.2 Application Business Flow

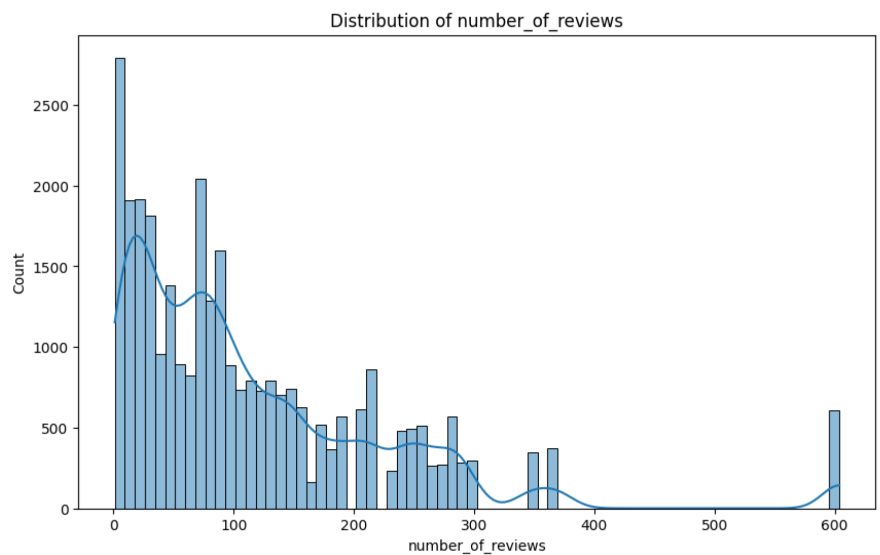
The InnJoy Vacation Rentals Recommendation System comprises a front-end and back-end, enhancing user interaction and personalized recommendations. Initially, it verifies user login credentials, redirecting unlogged users to the login page. Post login, user dialogues are sent to the backend for processing. The backend, built with Flask, handles user authentication and processes user input via custom chatbot algorithms to extract intents, entities, and features from user messages, aiding in understanding user needs. For new users, initial recommendations are based on overall popularity scores of rentals, while returning users get personalized recommendations analyzing their historical order features. During chatbot interaction, user responses are processed in real-time, with new features dynamically incorporated into the recommendation algorithm. If needed, the system prompts users for more input. Key data including recommendations and chatbot responses are sent to the front-end for display, ensuring a user-centric, engaging experience through seamless front-end and back-end coordination.

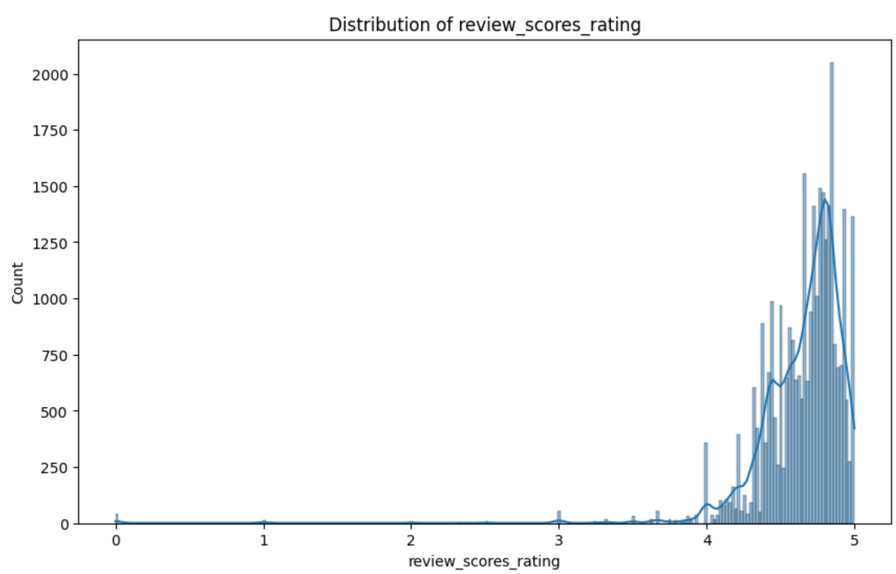
4.2 Knowledge Representation

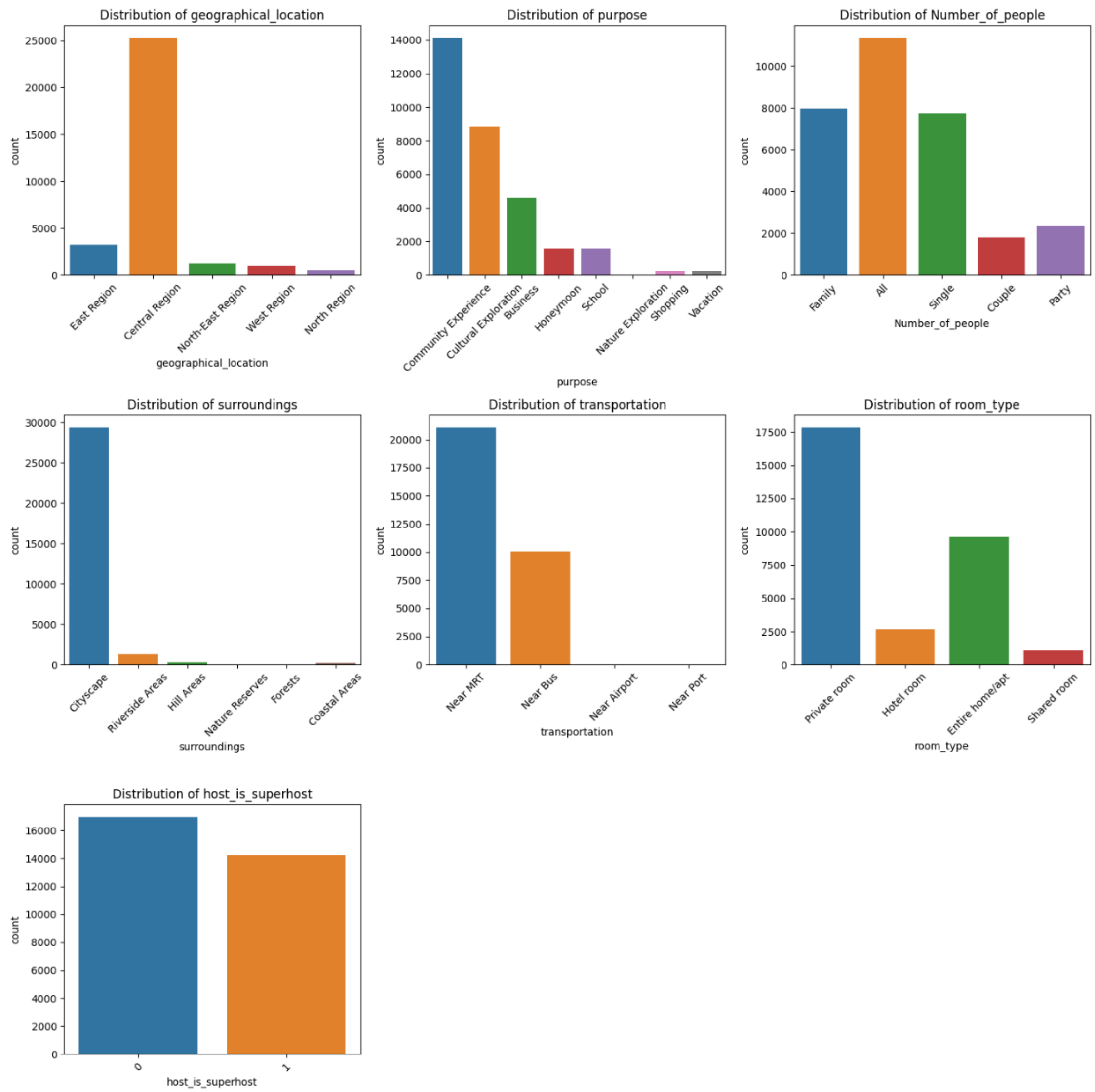
4.2.1 Database Structure

During the data analysis process, we conducted a visual analysis of the dataset to gain a more accurate and intuitive understanding of the distribution of accommodations and the corresponding user preferences. In the handling process, we categorized the data into the following types: continuous features ('price', 'number_of_reviews', 'review_scores_rating') and categorical features ('geographical_location', 'purpose', 'Number_of_people', 'surroundings', 'transportation', 'room_type', 'host_is_superhost').

Visualization is a crucial aspect of data analysis as it provides a visual representation of the data, making it easier to identify trends, patterns, and outliers. By categorizing the features into continuous and categorical, we can apply appropriate visualization techniques for each type. For instance:



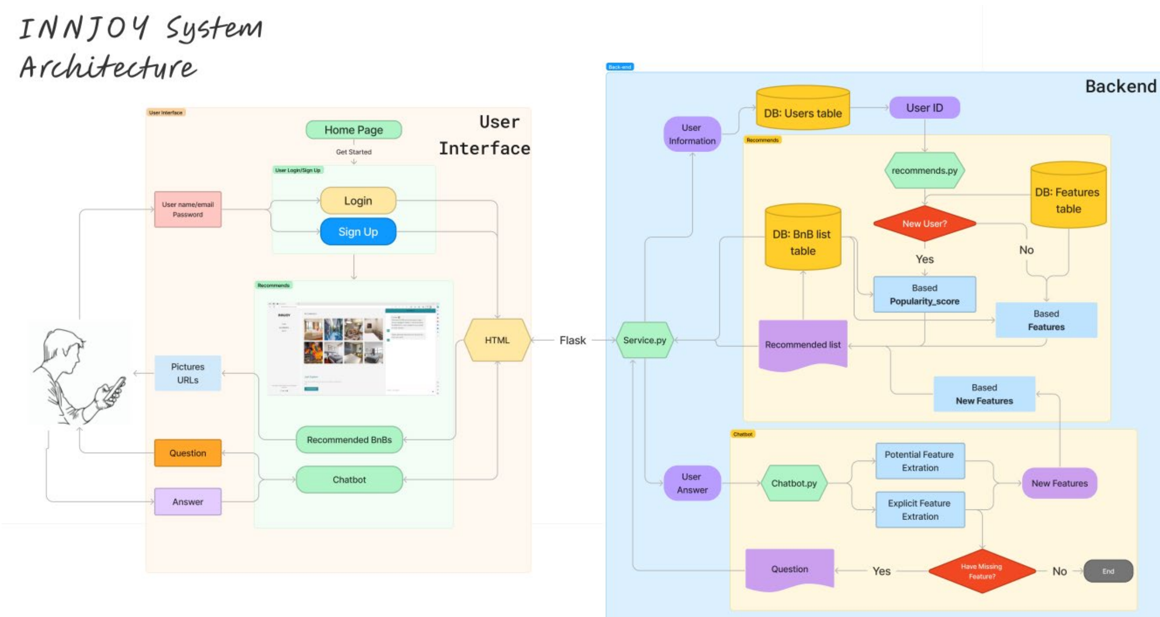



Distribution of geographical_location
count
East Region
Central Region
North-East Region
West Region
North Region
geographical_location
Distribution of purpose
Community Experience
Cultural Exploration
Business
Honeymoon
School
Nature Exploration
Shopping
Vacation
purpose
Distribution of Number_of_people
Family
All
Single
Couple
Party
Number_of_people
Distribution of surroundings
Cityscape
Riverside Areas
Hill Areas
Nature Reserves
Forests
Coastal Areas
surroundings
Distribution of transportation
Near MRT
Near Bus
Near Airport
Near Port
transportation
Distribution of room_type
Private room
Hotel room
Entire home/apt
Shared room
room_type
Distribution of host_is_superhost
0
1
host_is_superhost

5 Project Architecture & Implementation

5.1 Architecture Overview

Our system can be simply divided into two parts, the front-end and the back-end. The front-end mainly includes user interaction, providing communication with users, and obtaining user input text, among other functions. The back-end receives data passed from the front-end, processes user inputs through the rule-based system and the NCF (Neural Collaborative Filtering) model, and recommends accommodations from the database according to the user's preferences.



User Interface:

The User Interaction Pages of the InnJoy Recommendation System include the Home Page, Login & Sign Up Pages, Recommends Page, and About Page. These pages are designed using HTML and employ various frontend technologies to enhance user experience and interaction. InnJoy's Vacation Rentals Recommendation System offers a seamless and personalized user experience. Upon accessing the recommendation page, users are directed to the login/signup interface. New users receive random recommendations based on overall popularity scores, introducing them to a diverse array of vacation rentals. For returning users, the system employs their historical data and preferences to curate personalized suggestions, ensuring a tailored experience.

The system's unique feature lies in its chatbot interaction. Users can articulate preferences through text input, prompting the system to dynamically update recommendations in real-time. The chatbot not only responds to user input but also probes for additional details, guaranteeing a comprehensive understanding of user requirements. As the conversation unfolds, the system immediately incorporates user preferences, refining recommendations to match evolving needs.

Furthermore, the system promotes interactive exploration by enabling users to click on vacation rental images for detailed information. This seamless transition to property pages empowers users to make well-informed decisions. InnJoy's approach ensures a fluid and engaging user journey, where real-time interactions and personalized recommendations converge to create a user-centric platform, fostering satisfaction and facilitating effortless vacation rental selection.

Backend Functionality:

1. User Authentication:

--The backend handles user registration and login processes, securely storing user information in the Users table.

--Upon successful login, the backend records the UserID, enabling personalized user experiences and order history tracking.

2. User Classification and Recommendation Algorithm:

--Based on the UserID, the backend distinguishes between new and returning users.

--For new users, random recommendations are generated utilizing the BnB List table.

--Returning users' historical order features are analyzed, guiding the recommendation algorithm to provide tailored suggestions, enhancing user satisfaction.

3. Chatbot Interaction and Feature Extraction:

--The backend processes user responses from the chatbot, extracting relevant features for recommendation refinement.

--Unextracted features prompt the backend to generate targeted questions, guiding users for further input, ensuring a comprehensive understanding of their preferences.

4. Dynamic Feature Updates and BnB Recommendations:

--Real-time user responses lead to the extraction of new features, instantly updating the recommendation criteria.

--These dynamic updates enable the backend to refresh the recommended vacation rentals, aligning with users' evolving preferences and needs.

5.2 Process Flow

When a user accesses the InnJoy Vacation Rentals Recommendation System and navigates to the chatbot page, the system initiates a series of processes to enhance user interaction and provide personalized recommendations. First, the system verifies the user's login credentials. If the user is not logged in, they are redirected to the login page where they can either log in with existing credentials or register for a new account.

Upon successful login, the user's dialogue inputs are sent to the backend for processing. The backend, implemented using Flask, handles user authentication, ensuring secure communication between the frontend and backend components. The custom chatbot algorithms, developed in-house, handle the interpretation of user input. The system extracts intents, entities, and relevant features from the user's messages using these algorithms, allowing for a nuanced understanding of user requirements.

For new users, the system generates initial recommendations based on the overall popularity score of vacation rentals stored in the BnB List table. Returning users, identified through their UserID, benefit from a more personalized experience. Their historical order features are analyzed, enabling the system to tailor recommendations to match their preferences.

During the chatbot interaction, user responses are continuously processed by the backend. New features extracted from user input are dynamically incorporated into the recommendation algorithm. If certain features are not extracted, the system generates specific questions, prompting users for further input and ensuring a comprehensive understanding of their requirements.

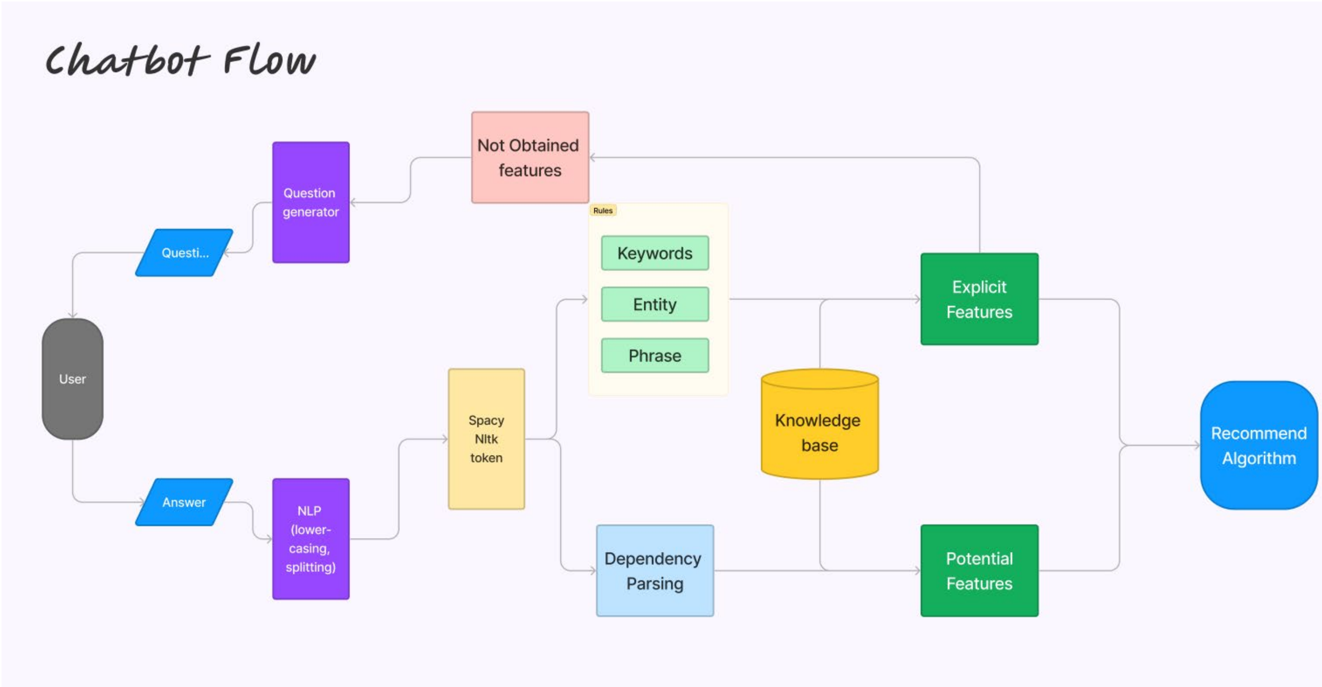
The important data, including vacation rental recommendations, chatbot responses, and relevant features, are packaged and sent to the frontend for display. Through this seamless coordination between the frontend and backend, the InnJoy Vacation Rentals Recommendation System offers an engaging and user-centric experience, enabling users to explore personalized recommendations and interact meaningfully with the platform.

5.3 System Modules

5.3.1 Chatbot

We aim to differentiate our system from other similar competitive products by employing a chatbot to interact with users, progressively extracting the necessary features through these interactions. This approach enhances the user experience by offering superior interactivity and a personal touch, subsequently recommending suitable accommodations based on the user's preferences.

In the chatbot module, we primarily focus on implementing three functions: posing questions to users, employing advanced Natural Language Processing (NLP) techniques to extract key features from user responses, and organizing these features to be sent to the Neural Collaborative Filtering (NCF) model.



Before get into the analysis of user response text, an essential task is to define which features can be captured from the users. Following internal discussions within our team and extensive research, we have decided to categorize the features found in user text into two types: explicit features and potential features. Explicit features are inherently present in the training dataset we have collected, including attributes like **'neighbourhood_cleansed', 'host_is_superhost', 'room_type', 'price', 'average_price',** and **'accommodates'**. On the other hand, potential features are derived from analyzing and processing the corresponding user review data within the training set. Through thorough analysis, we have identified five implicit features: **'geographical_location', 'purpose', 'number_of_people', 'surroundings',** and **'transportation'**.

The reason behind such categorization is: firstly, explicit features are more intuitively represented in user text, like price, room type, etc. We can collect these features by directly posing questions or prompting users, thereby gathering the requisite information. Secondly, potential features are the result of analyzing and summarizing reviews from the training dataset. We believe that it's feasible to extract these features from user text without directly querying the users. Through this delineation of features, we can adopt different machine learning approaches for varying categories of features, which in turn reduces the number of times asking questions, and also ensures the accuracy of feature extraction.

User Query Section

To obtain either explicit or implicit features from users, posing questions is crucial. Initially, we prepare a question pool. When aiming to capture the corresponding explicit features, we randomly select a question from this pool to present to the user:

```
question_room_type=['What kind of room do you want to live in?',
  'What kind of room do you want to stay in,private hotel, entire home or shared room?',
  "Would you prefer a private space or are you open to shared accommodations?",
  "Are you looking for a standard room, a studio, or a full apartment?",
  "Do you have a preference for a private or shared bathroom?",]
```

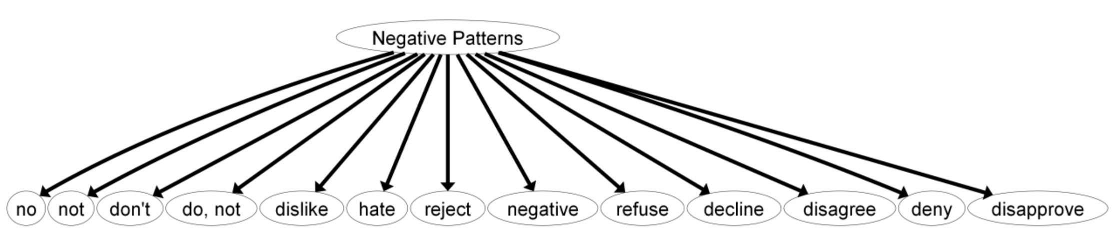
In addition, to avoid monotony in the questioning process, we also utilize NLP libraries for synonym substitution. Specifically, we employ the WordNet resource from the Nltk library to randomly replace certain synonyms within sentences, hoping to add variety and prevent our expressions from being overly monotonous.

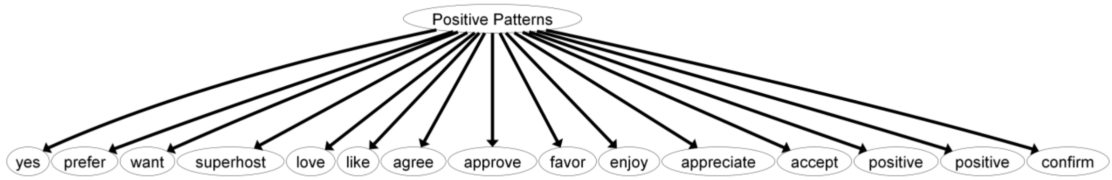
Extracting Explicit Features

Prior to the extraction of features, we pose questions to obtain user text. Subsequent processing of user text is essential, including operations such as splitting, lower-casing, lemmatization, and removal of stopwords. Through these standardization procedures, we can mitigate the interference caused by the diversity of user input, allowing us to identify and extract explicit features based on established rules. In handling these features, we employ methods like keyword matching, entity recognition, and rule-base knowledge to collect user characteristics.

Taking the identification of whether a user prefers the host to be a superhost as an example, there could be two categories of outcomes: negative, indicating no preference for the host being a superhost, and positive, indicating a preference for the host being a superhost. By analyzing the database, we identify common keywords associated with these preferences, categorizing them into negative and positive groups. Initially, we process the user text into tokens using the spaCy library, then separate each token and eliminate stop words. These keywords are then added to spaCy's matcher. Ultimately, we examine whether positive and negative features can be extracted from the tokens within user text.

Thus, employing a similar methodology, we will construct a rule-based system to perform the screening.





Additionally, we aim to ascertain the geographical location where users prefer to reside, which necessitates the use of entity recognition to accomplish. In this project, by utilizing spaCy's EntityRuler, we construct a new rule using the place names found in our dataset. This aids in identifying whether the desired place names are present within the user text.

```
patterns = [{"label": "GPE", "pattern": "Marina Bay"},
            {"label": "GPE", "pattern": "Orchard Road"},
            # ...
           ]
```

Potential Features

There are various reasons for categorizing intent features within user text as implicit features, one of the significant ones being our desire to minimize the number of questions asked, while extracting as much information as possible from the existing user responses. Hence, a notable aspect of implicit features often relates to sentence structure. By analyzing the structure, sentences are splitting into various verb-noun combinations through Dependency Parsing. Dependency Parsing aims to identify the relationships among words within sentences, representing these relationships through dependency labels, such as subject, object, modifier, etc. We then establish a rule system with these dependencies and match them for similarity with values within the implicit features to identify the most similar value of an implicit feature.

For instance, considering a designated implicit feature "Near Bus," users might not explicitly mention this phrase in their responses. Instead, descriptions might be along the lines of: "...want a convenient place, if close to bus station is better." If fortunate, and users provide similar descriptions, we can dissect the sentence dependencies and identify that the feature most synonymous is "Near Bus." Similarly, we would employ analogous methods to handle other features.

During our experimentation phase, we also attempted to use Latent Dirichlet Allocation (LDA) to discern user intent. However, LDA tends to be more accurate in identifying themes within longer texts, while our interactions primarily consist of short text exchanges. Therefore, after multiple trials, we settled on the approach of dissecting sentence dependency structures followed by similarity matching to identify implicit feature values.

5.3.2 NCF Model

Model Selection:

After delving into and understanding recommendation systems, we have decided to employ the Neural Collaborative Filtering (NCF) algorithm as our primary recommendation algorithm. The rationale is as follows:

1. Hybrid Input Features:

Our data comprises continuous features (such as price, number of reviews, and ratings), categorical features (like location, purpose, room type, etc.), and the IDs of users and items. Traditional collaborative filtering methods often struggle to handle such hybrid feature inputs. NCF can amalgamate these diverse inputs through embeddings and a deep neural network, achieving feature fusion and interaction.

2. Non-linear Relationships:

Compared to traditional matrix factorization techniques, NCF is capable of capturing complex non-linear relationships between users and items. By utilizing a deep neural network, the NCF model can represent more intricate interactions between users and items.

3. Flexibility:

NCF allows us to easily incorporate other meaningful features (such as price, rating, etc.) into the model, whereas traditional collaborative filtering methods might require modifications or the addition of extra techniques to achieve this. This perfectly adapts to the characteristic of our project: acquiring new features through real-time interaction with users and making listing recommendations based on these new features.

The decision to employ the Neural Collaborative Filtering (NCF) algorithm aligns well with the complex and hybrid nature of our data. Its capability to handle both categorical and continuous features while capturing non-linear relationships between users and items provides a robust and flexible framework for our recommendation system. Moreover, the ease of integrating additional meaningful features makes NCF a highly adaptable choice for our project, enabling a more personalized and responsive recommendation experience for our users.

Data Preprocessing:

Normalization of Continuous Features:

- Action: Normalize continuous features (price, number_of_reviews, review_scores_rating) to have a mean of 0 and a standard deviation of 1.

- Reason: Normalization helps ensure that different continuous features are on a similar scale, preventing any adverse effects on the model due to overly large ranges of some features.

- Benefit: Improves the stability and speed of model training, and ensures comparability between features.

One-Hot Encoding of Categorical Features:

- Action: Perform one-hot encoding for categorical features (geographical_location, purpose, Number_of_people, surroundings, transportation, room_type, host_is_superhost).

- Reason: One-hot encoding transforms categorical features into binary form, enabling the model to understand and process categorical information.

- Benefit: Allows the model to learn the impact of different values of each categorical feature on the target without introducing unreasonable numerical relationships.

Label Encoding for Unique Keys:

- Action: Utilize LabelEncoder to encode unique identifiers (reviewer_id and listing_id) into integers.

- Reason: The model needs to handle these identifiers, but they cannot be fed into the model directly, hence encoding is required.

- Benefit: Maps identifiers to integers, enabling the model to process them, and reducing the dimensionality of the data.

Selecting Columns with Boolean Type and Converting to Float:

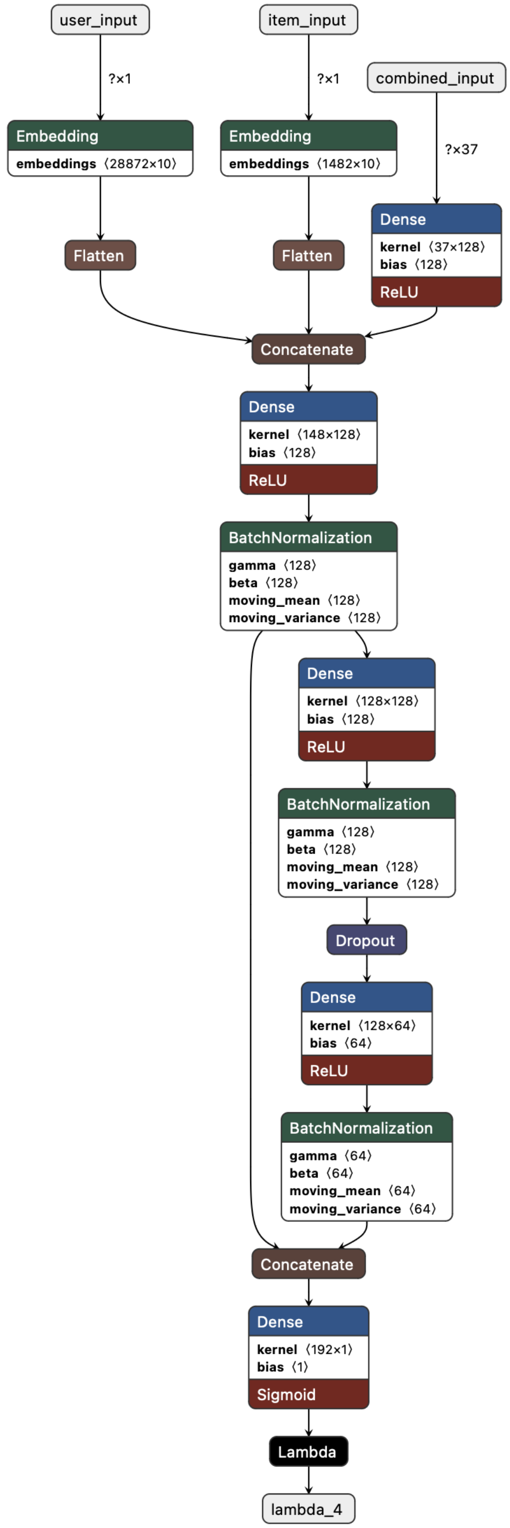
- Action: Select columns with Boolean type, then convert these columns' data types to floating-point numbers.

- Reason: Boolean features usually indicate whether a certain condition is met or not, converting them to floating-point numbers ensures a consistent data type with other features.

- Benefit: Ensures all features are fed into the model with the same data type, avoiding data type mismatch issues.

The above preprocessing steps are crucial to prepare the data for training and ensuring that the Neural Collaborative Filtering (NCF) algorithm can effectively learn from the data. By standardizing continuous features, one-hot encoding categorical features, label encoding unique keys, and converting Boolean columns to float, we are essentially ensuring that the data fed into the model is clean, well-structured, and ready for training. Each of these preprocessing steps brings us one step closer to building a robust and efficient recommendation system.

Model Construction:



This model serves as the deep learning model used for the recommendation system in this project:

1. Setting Embedding Dimension:

- `embedding_dim = 10`: The dimension of the embedding layer is set, which determines the size of the learned representation for users and items.

2. User and Item Embedding Layers with Regularization:

- `user_input` and `item_input`: These are the input layers for user and item IDs.

- `Embedding` layers: These layers are used to learn low-dimensional representations (embeddings) for users and items, with regularization applied to prevent overfitting.

- `user_vector` and `item_vector`: These layers flatten the outputs of the embedding layers, obtaining vector representations for users and items.

3. Merging Continuous and Categorical Features:

- `combined_features`: A feature list is created containing continuous and one-hot encoded categorical features, which will be concatenated in the next step.

4. Merging Input Layers:

- `combined_input`: This is the input layer for merging features.

- `Dense` layer: A dense layer is applied to process the merged features, reducing their dimensionality and adding non-linearity.

5. Combining All Feature Vectors:

- `all_vectors`: This gathers all feature vectors including user vector, item vector, and processed merged features.

6. Deep Neural Network with Residual Connections:

- `residual` layers: These layers introduce residual connections, aiding in alleviating the vanishing gradient problem, enabling the model to learn more effectively.

- `dense` layers: These layers create a deep neural network, including batch normalization and dropout layers for regularization.

7. Merging with Residual Connections:

- `dense`: The output of the deep neural network is merged with the output of the residual connections.

8. Output Layer:

- `outputs`: This layer generates the final output, which is a single value.

- `Lambda` layer: A Lambda function is applied to scale the output from [0,1] to [1,5], which is the desired output range for the regression task.

9. Building and Compiling the Model:

- `Model` class: A Keras model is created by specifying inputs and outputs.

- Learning Rate Decay: Exponential decay of the learning rate is employed to adaptively adjust the learning rate during training.

- `Adam` optimizer: Adam is a popular optimization algorithm used for training deep neural networks.

- Loss Function: Mean Squared Error (MSE) is used as the loss function for the regression task.

Advantages:

- The model incorporates embeddings of users and items, capturing the latent relationships between users and items.

- It handles a variety of features, including continuous and categorical features, offering flexibility in dealing with different types of data.

- Residual connections assist the model in learning more effectively.

- Batch normalization and dropout layers enhance the model's generalization capability.

- Learning rate decay helps to improve training stability.

In summary, this model integrates the embedding representations of users and items, processes a variety of features, and applies residual connections and deep neural networks to learn complex patterns in the data. It is highly suitable for recommendation tasks, especially personalized recommendations that need to consider users, items, and other features. The constructed model is poised to effectively learn from the data and provide meaningful recommendations, tailoring to the nuances of user-item interactions and other feature influences within the recommendation domain.

6 Project Performance & Validation

Hyperparameters:

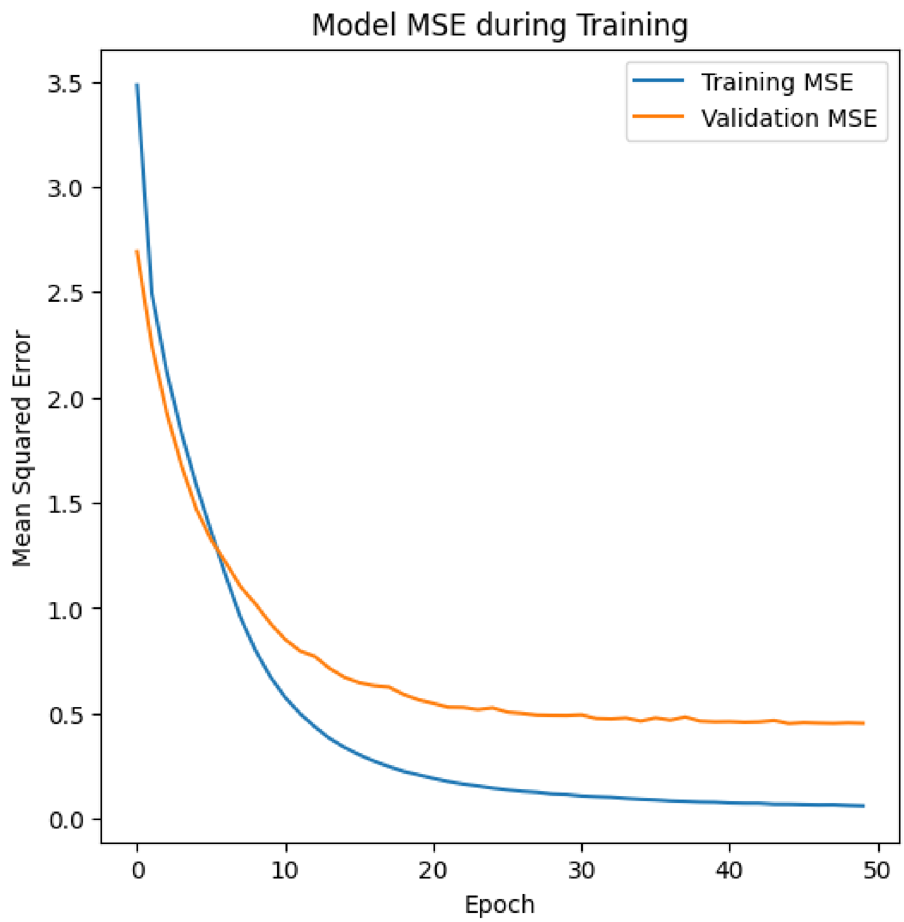
In choosing the model hyperparameters, to achieve the best model training effect, we attempted to adjust 4 hyperparameters including initial_learning_rate, batch_size, embedding_dim, and dropout. The results are shown in the following graph:

initial_learning_rate	batch_size	embedding_dim	dropout	MSE
1E−04	32	10	0.2	**0.4582**
1E−03	64	10	0.3	0.4603
5E−04	64	10	0.3	0.4617
1E−04	32	10	0.3	0.4849
5E−04	32	10	0.3	0.5042
1E−04	64	15	0.3	0.5168
1E−03	32	10	0.2	0.6251
5E−04	32	10	0.2	0.7409

To achieve the best predictive performance, we aim for the Mean Squared Error (MSE) to be as small as possible. Therefore, the hyperparameter combination we chose is:

initial_learning_rate	batch_size	embedding_dim	dropout
1E−04	32	10	0.2

Model training and evaluation:

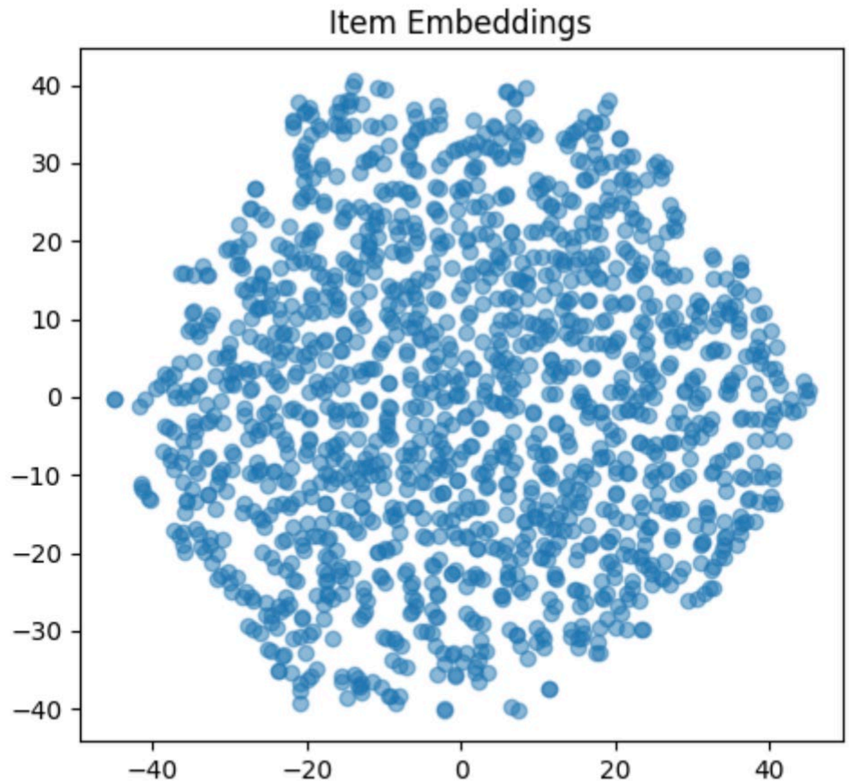


Test MSE: 0.45
Test RMSE: 0.67

The training data indicates that although the training loss is consistently lower than the validation loss, the difference is not particularly large, especially in the later stages; despite slight fluctuations, there is no obvious upward trend in validation loss. The model underwent training for 50 epochs, and although the validation loss slightly increased in some epochs, there is no significant upward trend. There are no apparent signs of overfitting in the model. Both training loss and validation loss continue to decrease, so the model is not underfitting. While in other models, a Test MSE of 0.45 might be considered a large error, in this model with a prediction output range of [1-5] scores, we believe 0.45 is a rather ideal result in the domain of homestay recommendations.

Overall, the model seems to have reached a relatively good balance.

Given the characteristics of the dataset, we mainly focus on the performance of item embeddings. Using the scikit-learn library's t-SNE, we can reduce the embedding vectors from their original high-dimensional space to 2D or 3D for easier visualization.



We can observe that the distribution of item embeddings is quite broad, indicating that the model has identified certain differences among the items. When we input new item information into the model, the disparities among item embeddings might lead to precise and distinct predictions. As the primary objective of our product is to provide recommendations based on new item features input by users, and considering the results of the Test MSE, we believe that the current model is in a satisfactory state.

7 Challenge & Recommendation

7.1 Challenges

1. Data Collection and Privacy Concerns:

Data collection is a critical step in building robust and effective machine learning models, including those used in recommendation systems. However, it comes with a set of challenges, especially when dealing with user data, which might entail personal or sensitive information. Ensuring the privacy and security of user data is paramount. This may require anonymizing data, obtaining user consent, or employing privacy-preserving techniques like differential privacy. Moreover, legal and ethical guidelines, such as the General Data Protection Regulation (GDPR) in Europe, also dictate how personal data should be handled, making compliance another crucial aspect.

2. Data Cleaning and Feature Selection:

The raw data collected often contains noise, missing values, or irrelevant information that can adversely affect the performance of the model. Data cleaning aims to address these issues by removing or correcting erroneous data, filling missing values, and handling outliers. Feature selection is the process of identifying the most relevant features (or attributes) from the data that are useful for solving the problem at hand. This often involves a combination of domain knowledge and algorithmic techniques. Domain experts can provide invaluable insights into which features are likely to be relevant, while algorithmic feature selection methods can automate the process of identifying useful features from the data. Together, they ensure that the model is trained on high-quality and relevant data, which in turn, significantly impacts the model's performance.

3. Collection of Training Data for Language Models:

The performance of language models is heavily reliant on the quality and quantity of training data available. Collecting a diverse and representative dataset is essential for training robust language models. The training data should encompass a wide range of linguistic variations, topics, and styles to ensure the model generalizes well across different use cases. Additionally, the data collection process should adhere to ethical guidelines, ensuring that the data is collected in a fair and unbiased manner. The source of training data could be publicly available datasets, proprietary data, or data obtained through partnerships with other organizations. Moreover, data augmentation techniques can be employed to expand the training dataset and introduce more variability, further improving the model's ability to understand and process language effectively.

7.2 Future Improvements

During the process of posing questions to users, there's room to increase text diversity which can make interactions more engaging. Collecting feedback from users during the exchange can be instrumental for further optimization, helping to improve the user experience and the efficacy of the information gathering process.

The current feature extraction process is rule-based which inevitably has its limitations, or may lack sufficient generalization capacity. Given ample time in the future, there is a desire to collect or even create a text dialogue dataset. This dataset can be leveraged to fine-tune models like T5 or train chatbots using platforms like Rasa through transfer learning. This approach can potentially lead to more accurate extraction of user features by understanding user intents and responses in a more sophisticated and nuanced manner. By employing machine learning models, the chatbot can adapt and improve over time, better understanding and extracting the required features from user interactions.

APPENDIX OF REPORT A

PROJECT PROPOSAL

Date of proposal: 2 Oct 2023
Project Title: InnJoy: A Personalized Homestay Matching Service
Group ID (As Enrolled in Canvas Class Groups): Group 18 (FT) **Group Members (name , Student ID):** YANZIHAN ZHANGYAOXI A0285851U TAOXU WEICHUANJIE A0285709N
Sponsor/Client: *(Company Name, Address and Contact Name, Email, if any)* Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore NATIONAL UNIVERSITY OF SINGAPORE (NUS) Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg
Background/Aims/Objectives: Traveling is a wonderful way to enrich our lives with different cultures and landscapes. However, finding a suitable place to stay and explore can be difficult, especially in unfamiliar places. Traditional hotels may not offer the authentic and personalized experience that travelers desire. That is why we created ***InnJoy: A Personalized Homestay Matching Service***. It is an intelligent system that helps travelers find their perfect homestays, attractions and routes. InnJoy is more than just a recommendation system, it is a smart travel companion. It uses advanced algorithms and data analysis to match travelers with homestays that fit their preferences, budgets and moods. It also provides suggestions for attractions and activities that are hidden gems in the place, as well as optimal routes to reach them. With InnJoy, travelers can enjoy a more immersive and memorable travel experience, while discovering

the local culture and scenery.

Our vision is to connect travelers with the world through InnJoy. We believe that every traveler deserves to find their ideal homestay and place, and we are here to make it happen.

The main goal of InnJoy is to create a homestay recommendation system that can provide users with personalized and satisfying travel experiences. To achieve this, the system will:

- Use machine learning and artificial intelligence technologies to develop a homestay recommendation algorithm that can understand and predict the needs and preferences of bnb users, based on their profiles, preferences, and past behaviors.
- Collect and analyze tourism data from all over the world, using various sources such as online reviews, social media, travel blogs, etc., to recommend various attractions for users, ranging from popular to hidden, from traditional to novel, according to their interests and expectations.
- Integrate a highly intelligent chatbot that can understand users' specific questions, provide them with 24/7 personalized recommendations and instant answers, and achieve the true "anytime, anywhere, responsive" service. The chatbot will also be able to handle complex requests such as booking, canceling, rescheduling, etc., and provide users with relevant information and reminders.
- Regularly collect and analyze user feedback, using methods such as surveys, ratings, comments, etc., to continuously optimize the recommendation system, and ensure that it can always meet the changing and upgrading needs of travelers. The system will also use user feedback to improve its own learning and performance.

Project Descriptions:

➢ **Recommendation Algorithms**

Collaborative filtering algorithms:

- User-Based Collaborative Filtering (UBCF): It recommends homestays that are liked by other users who have similar interests to the target user.
- Item-Based Collaborative Filtering (IBCF): It recommends homestays that are similar to the target homestay to the user.

Content-based recommendation algorithms:

- Feature extraction and selection: It extracts key features from the homestay data, such as location, price, facilities, rating, etc. It uses machine learning algorithms (such as decision trees, random forests) to make recommendations based on the relationships between features.

Deep learning models:

- Neural Collaborative Filtering (NCF): NCF uses neural networks to learn hidden feature vectors of users and items, reflecting their preferences and similarities. It uses these vectors to compute the match degree between users and items, predicting ratings or click probabilities. It recommends the most suitable homestays to users based on the match degree.

➢ **Personalized recommendation strategies:**

New user recommendation strategies:

- Interactive Onboarding: When new users first visit the app, guide them to fill out a questionnaire or choose options about their personal preferences. Provide a user-friendly interface, using images, icons or short videos, to help users express their interests more intuitively.
- Popular Recommendations: Recommend the most popular homestays in the current city or region to new users. Based on popularity, number of reviews and high ratings, ensure that the first homestay experience for new users is satisfactory.
- Category-Based Recommendations: According to the information provided by users in the interactive onboarding, recommend them specific categories of homestays, such as "family-friendly", "luxury suites" or "independent cottages". Based on the features selected by users, combine content-based recommendation algorithms to recommend corresponding homestay types.

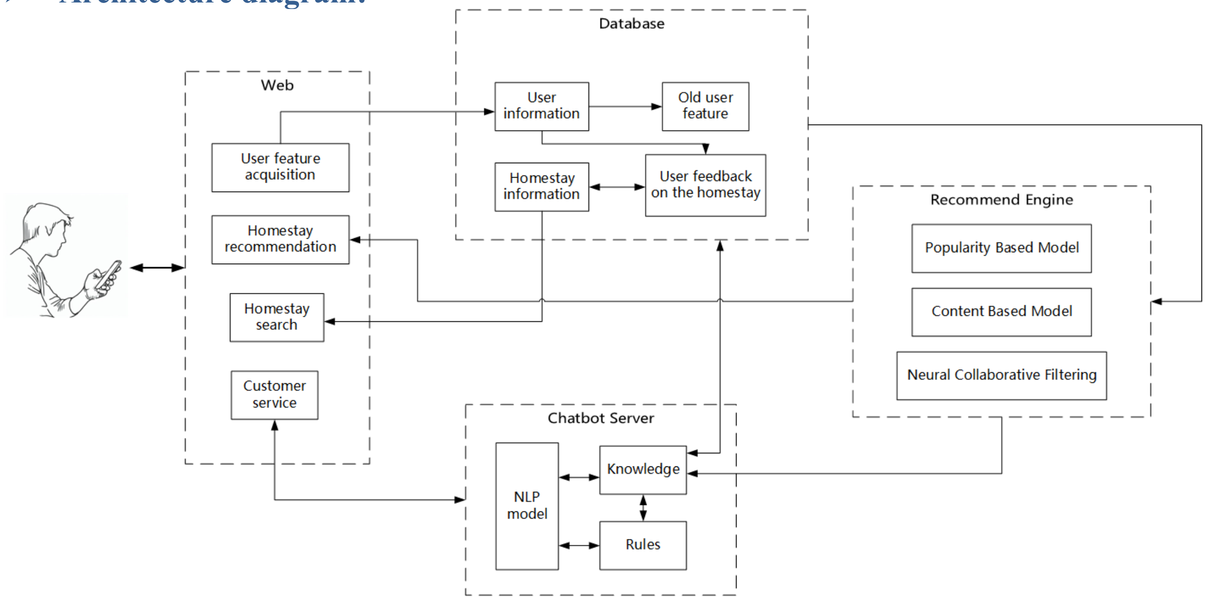
Old user recommendation strategies:

- Historical Behavior Analysis: Analyze the booking history of old users, including their preferred locations, price ranges, room types, etc. Based on these historical data, recommend homestays that are similar to their previous preferences.
- User-Based Collaborative Filtering: Use user-item collaborative filtering to find other users who have similar behavior to old users. Based on the booking history of similar users, recommend homestays that are liked by these similar users to old users.
- Neural Collaborative Filtering: Use neural network models, combined with the historical data of old users, to learn the complex relationships between users and homestays. With the help of deep learning, capture the preferences and needs of users more accurately, and provide highly personalized recommendations.

➢ **Chatbots:**

- Natural Language Processing (NLP): Using NLP techniques, chatbots can understand the natural language queries of users and provide accurate responses.
- Machine Learning: Based on machine learning algorithms, chatbots can gradually learn and improve the accuracy of their answers and the understanding of user needs.
- Integrated APIs: Integrating external APIs, chatbots can access real-time information such as weather, traffic and location, and provide more comprehensive services.

➢ Architecture diagram:



➢ Schedule:

Date	Task
Before 6 Oct	Detailed tech investigation, Feasibility assessment
6 Oct	Discussion
6 Oct – 20 Oct	Propel each section separately
11,15 Oct	Discussion
15 Oct – 20 Oct	System integration and test
20 Oct – 25 Oct	Report writing, video production

APPENDIX OF REPORT B

Mapped System Functionalities against knowledge, techniques and skills of modular courses

Modular Courses	System Functionalities / Techniqe Applied
Machine Reasoning (MR)	• **Knowledge Elicitation and extraction**: Extracting meaningful features from lodging information is a crucial step in building a recommendation system or conducting data analysis • **Knowledge Representation**: Filter user features, and extract new features from the text, integrating them into the model input. Eventually, predict the scores of the accommodations based on the model output, and recommend the high-scored accommodations to the users.. • **Rule Based System** In summarizing language reviews, design a rule-based chatbot to obtain user features.
Reasoning System (RS)	• **NCF Recommend System** Utilizing neural networks, analyze the relationship between users and accommodations separately, and construct a model for recommendations.
Cognitive System (CGS)	• **Chatbot Design** Experiment with various language models and a variety of Natural Language Processing libraries, ultimately designing a chatbot capable of interacting with users and obtaining features.

APPENDIX OF REPORT C

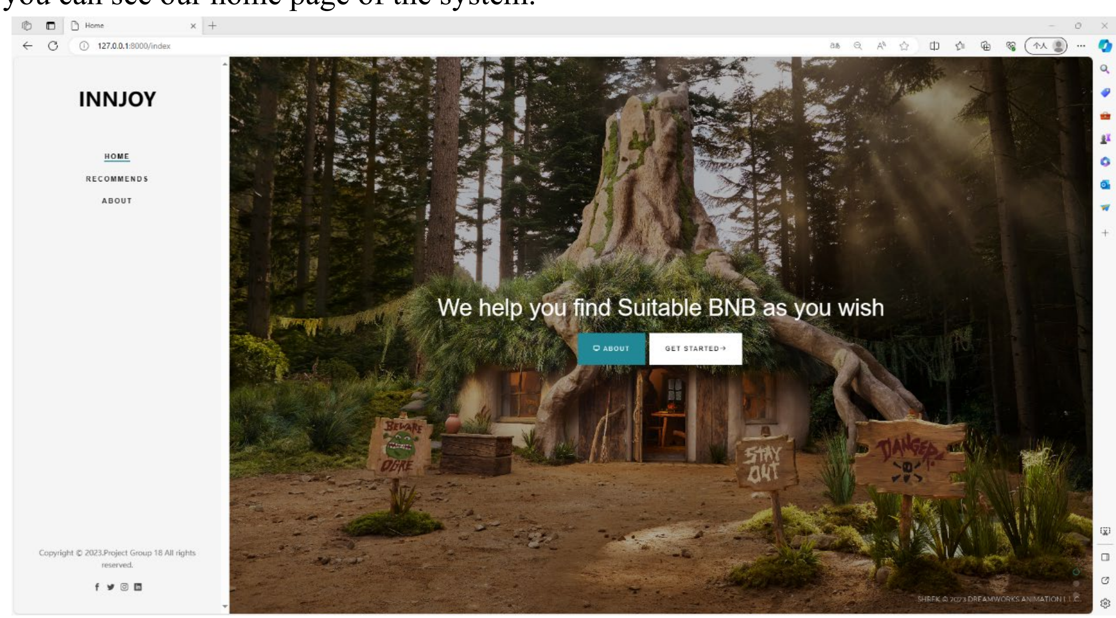
Installation and User Guide

Environment Requirement:

- **Python 3.9:**
  Install it from the official Python website: https://www.python.org/
- **Package:**
  Inside the system code folder, you will find a file named requirements.txt. Run the following command to install the required dependencies using pip:
  > **pip install -r requirements.txt**
  The above command will install Flask, Flask-CORS, Flask-SocketIO, NumPy, Pandas, TensorFlow, scikit-learn, joblib, Spacy, NLTK, and Folium packages necessary for the system to run.

Running the System:

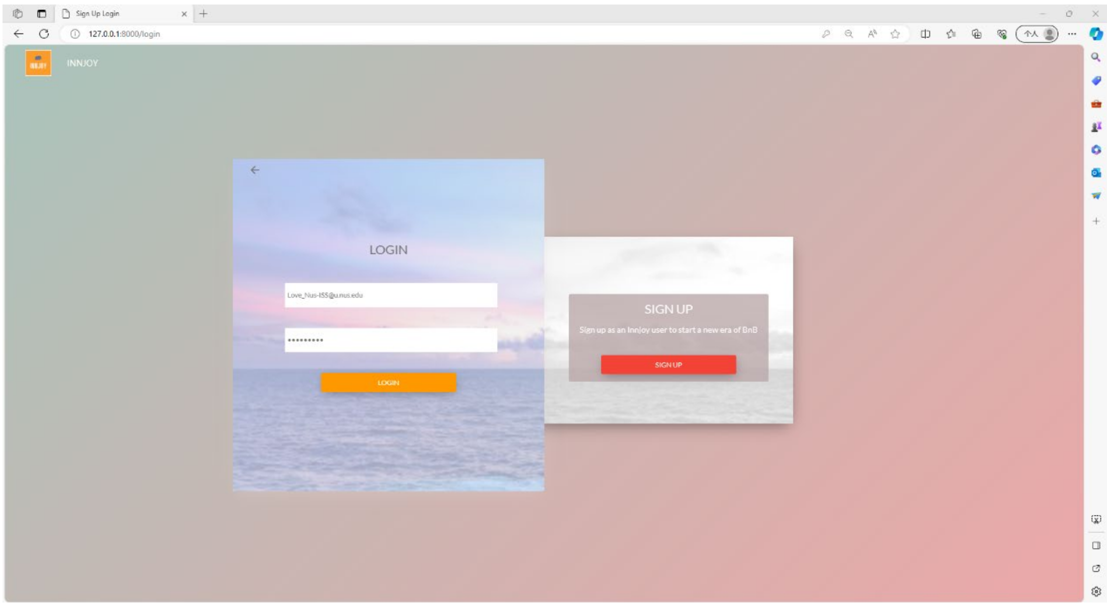
1. **Start the Flask Server:**
   Run the following command to start the Flask server in the terminal or command prompt, navigate to the system code folder:
   > **python service.py**
2. **Access our System:**
   Once the Flask server is running, open your web browser. Enter the following URL to access the INNJOY Recommendation System:
   http://localhost:8000/index
   Then you can see our home page of the system.



Using our System:

1. Registration and Login:

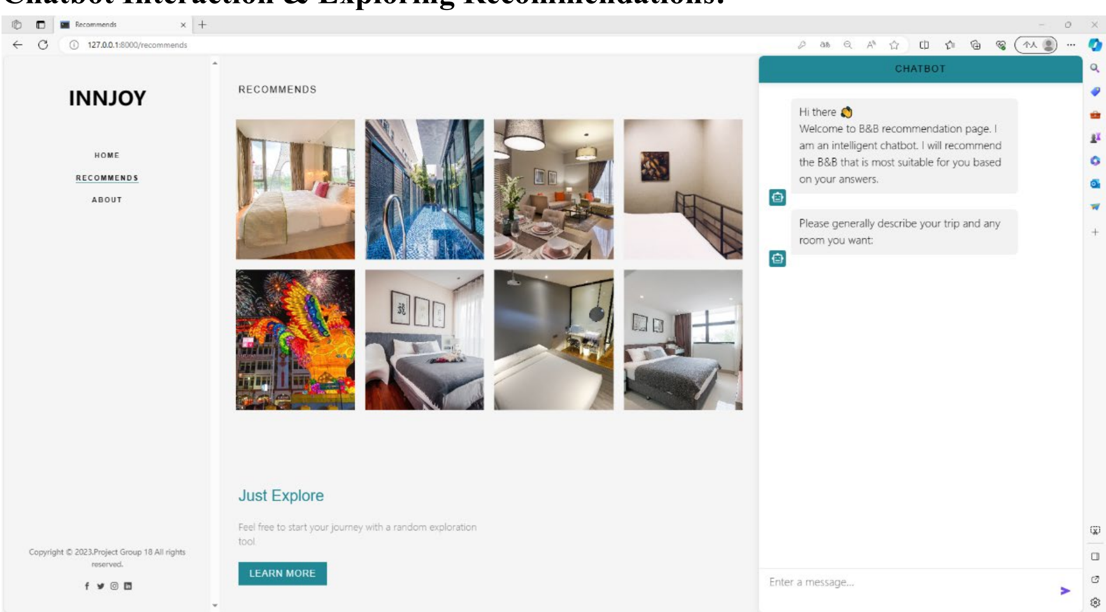
When you click Get started on the Home page, or click the Recommends tab on the left navigation bar, if you have not logged in recently, you will automatically jump to the Login & Sign up page.



- If you are a new user, click on the "SIGN UP" button to create an account. Fill in the required details.
- If you are a returning user, click on the "LOGIN" button and enter your credentials to log in.

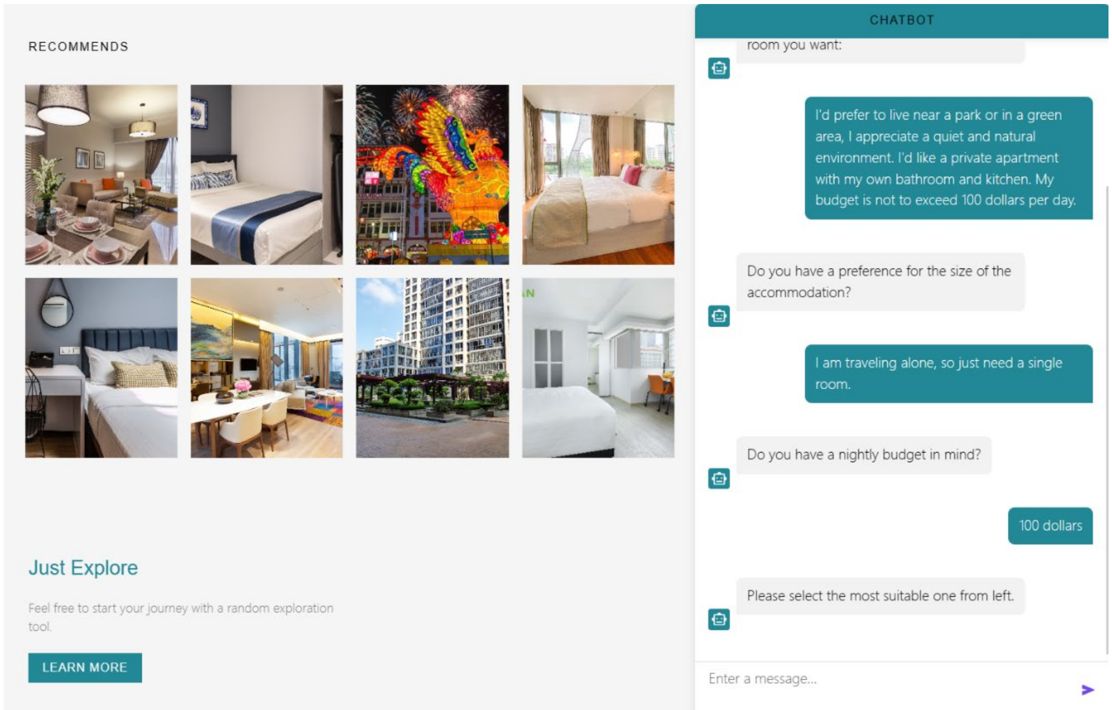
After logging in, you will be redirected to the Recommends page.

2. Chatbot Interaction & Exploring Recommendations:

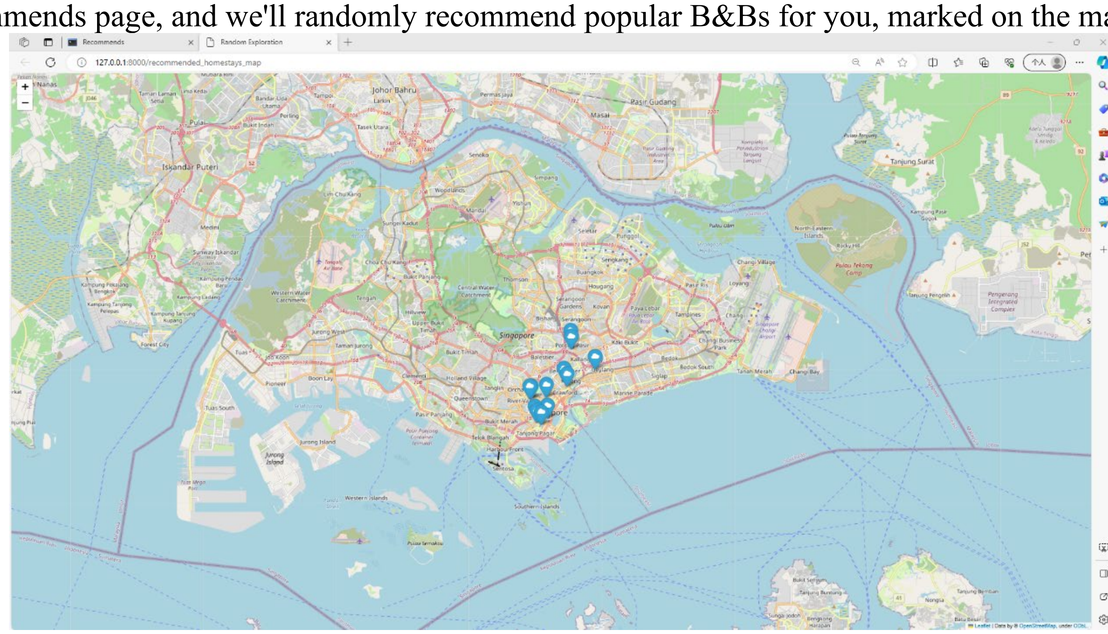


- When you first enter the page, the recommendation system will recommend 8 B&Bs that are more suitable for you based on their popularity scores and your historical orders.
- Use the chatbot interface to input your preferences and requirements for vacation rentals. The chatbot will guide you through the conversation, asking questions if necessary, to understand your preferences better.

- After each answer, the system will provide personalized recommended B&Bs based on your answer. You can click on the picture at any time to enter the B&B reservation page and view more detailed B&B information.



3. **Explore popular BnBs randomly:**
When you're confused about your travel plans, you can click learn more in Just explore below the recommends page, and we'll randomly recommend popular B&Bs for you, marked on the map**.**



Wish you a happy use!

APPENDIX OF REPORT D

Individual Reports

Individual Report: Tao Xu (A0285941U)

Personal Contribution

In this project, I performed as a core development role, primarily responsible for the design and development of the chatbot, ensuring accurate extraction of key features from user texts to recommend the most suitable accommodations in Singapore. In the aspect of feature definition and selection from the dataset, I participated in the definition and selection of features to ensure that our model could accurately interpret and cater to the users' needs and preferences. Simultaneously, I was responsible for language model research and deployment. In the early stages of the project, I conducted in-depth technical research, evaluating various cutting-edge language models and frameworks such as T5, Rasa, and Bert. Ultimately, based on the project requirements and technical assessment, I selected the most suitable technical path for the team. The development of the chatbot' was a major part of my work, which included question identification and natural language processing (NLP). By utilizing powerful NLP libraries like spacy and nltk, I successfully built an efficient chatbot system capable of understanding and responding to user inquiries. In the aspect of user feature extraction and analysis, I proposed and implemented a comprehensive feature extraction scheme. Through designing a rule system and employing similarity matching algorithms, we successfully mined valuable features from user text interactions. These features were organized in a standardized format for the Neural Collaborative Filtering (NCF) model to use, thus achieving accurate recommendations.

Learning outcome

In this project, I acquired a wealth of knowledge regarding recommendation algorithms and Neural Collaborative Filtering (NCF). This experience enlightened me that recommendation algorithms necessitate a consideration of various domains of knowledge including sociology and business behavior. Simultaneously, it became evidently clear that data acquisition and feature engineering are pivotal in real-world projects. Throughout this project, we had numerous discussions, modifications, expansions, and feature explorations concerning the dataset. A well-structured dataset often lays the foundation for success.

As I completed my part of the work, I delved into and learned about cutting-edge language models and attempted to utilize these models for feature extraction in the project. This process also deepened my understanding of Natural Language Processing (NLP), providing me with a preliminary insight into rule construction and exploration within this domain. Additionally, the iterative debugging and refinement during the system construction phase significantly enhanced my capabilities in project code management.

Throughout the project process, I fully integrated myself into the various roles within the team. I realized the paramount importance of team progress management and communication. It's essential for the entire team to maintain harmony and work synchronously at the same pace. Within the scope of project code management, I recognized the necessity for individuals to adhere to standardized file formats and unify program encapsulation interfaces. Maintaining good communication and having moderate rational collisions are the cornerstones of a well-functioning team.

Knowledge and Skill Application

The implementation of this project highlighted the importance of continuous learning to me. I not only enhanced my technical capabilities, but also gained a clear roadmap on how to better apply advanced Natural Language Processing (NLP) techniques and address data issues in the industrial sector in future projects. Simultaneously, I will take industrial data processing issues seriously and strive to maintain good communication, progress management, and code interface standards in team collaborations, to ensure the success of the projects and promote continuous growth for both individuals and the team. In project engineering, it's essential to exhibit flexibility in addressing real-world problems. In the future, the ability to translate an industrial problem into a machine learning problem is extremely crucial. Therefore, mastering data collection, processing and analysis, along with problem modeling capabilities, are vital skills for development in the AI domain.

Individual Report: Wei Chuanjie A0285709N)

Personal Contribution

1. Conceptualization and Ideation

I played a pivotal role in the conceptualization and ideation phase, formulating and envisioning the overarching concepts and direction for the entire project. My active involvement in brainstorming and discussions helped in laying a strong foundation for the project.

2. Model Development and Recommendation Strategies

I spearheaded the development of the Neural Collaborative Filtering (NCF) model. My focus was on the integration and optimization of three core recommendation methodologies, which significantly contributed to the robustness and efficiency of our recommender system.

3. Design and Implementation of Recommendation Features

3.1 Designed and implemented three distinct recommendation features catering to different user needs including personalized recommendations for existing users, new user recommendations based on user similarity, and recommendations based on weighted popularity.

3.2 Deeply understood user needs and satisfied them through algorithmic design, enhancing user satisfaction and experience.

3.3 Continuously evaluated and optimized the performance of the recommendation system to ensure the accuracy and efficiency of recommendations.

4. Team Management and Project Coordination

As a coordinator, I managed teams to ensure alignment with project objectives. My efforts were instrumental in setting and adhering to project timelines and schedules, ensuring the smooth progression and timely completion of the project

Learning outcome

1.Comprehensive Understanding

Through this project, I gained a comprehensive understanding of recommender systems, especially the NCF model. The theoretical knowledge coupled with practical application provided a deep insight into the workings of recommendation algorithms.

2.Data Preprocessing Techniques

I learned the importance and implementation of various data preprocessing techniques such as feature normalization, one-hot encoding, and Label Encoding, which are crucial for preparing the data for training the model.

3.Model Optimization and Evaluation

The project honed my skills in model optimization and evaluation. The iterative process of tweaking the model, analyzing the results, and making necessary adjustments provided a hands-on experience in improving model performance and evaluating its effectiveness.

4.Integration of Recommendation System Theory and Practice

4.1Through the design and implementation of three different recommendation features, a deep understanding of the theory and practical application of personalized recommendation, collaborative filtering, and popularity-based recommendation techniques was acquired.

4.2Learned how to design recommendation algorithms based on user needs and business objectives, and validated the effectiveness and performance of algorithms through practical application.

4.3 Gained experience in applying deep learning models and recommendation system techniques in real projects, as well as knowledge on how to optimize recommendation algorithms based on project requirements and user feedback.

Knowledge and Skill Application

1.Data Description and Preprocessing

With a focus on Airbnb listings, the dataset primarily contained information regarding the listings while lacking ample user data due to collection challenges. The preprocessing involved standardizing continuous features, one-hot encoding categorical features, Label Encoding unique identifiers, and converting boolean features to float.

2.Model Selection and Construction

The NCF model was chosen for its capability to handle mixed feature inputs, capture non-linear relationships, and its flexibility to incorporate additional meaningful features. The model construction involved setting embedding dimensions, creating user and item embedding layers, merging features, constructing a deep neural network with residual connections, and designing an output layer for the regression task.

3.Model Performance and Analysis

The model exhibited a Test Mean Square Error (MSE) of 0.45 and a Root Mean Square Error (RMSE) of 0.67. Although there were slight fluctuations, the validation loss did not exhibit a rising trend, indicating no apparent overfitting. The model demonstrated a reasonable balance in performance, especially considering the prediction output range of [1-5] in the domain of bnb recommendations.

4.Design and Optimization of Recommendation Algorithms:

4.1Utilized knowledge in deep learning and recommendation systems to design and implement recommendation features catering to different user needs including personalized recommendations,

recommendations based on user similarity, and recommendations based on weighted popularity.

4.2 Applied theoretical knowledge of collaborative filtering, deep learning, and recommendation systems to solve practical recommendation problems in the project, improving the accuracy of recommendations and user satisfaction.

4.3Deepened understanding of recommendation system technologies and deep learning models through practical project application, gained valuable practical experience, and learned how to optimize recommendation algorithms based on user feedback and business requirements.

Individual Report: Yan Zihan (A0285706W)

Personal Contribution

In this project, my primary responsibility was handling data. The data primarily consisted of information related to lodgings (like Airbnb) and user-related details. The challenge with processing lodging information was the extensive variety of information categories present. Dozens of different features were not all necessary for us. Hence, data cleansing was essential to eliminate irrelevant features. To address this, I created a dictionary that described corresponding features, aiding in the selection of pertinent information. After identifying the necessary features, additional implicit features were generated by integrating them with practical information. Subsequently, data preprocessing was conducted to convert all information into the required format and eliminate redundant features.

Regarding user information, the challenge lay in the limited number of features compared to lodging data. This necessitated the creation of new features. By analyzing user review texts, I generated user ratings for lodgings and identified user preferences.

Furthermore, I contributed to designing a part of the recommendation system. This segment achieved recommending lodgings based on their ratings, specifically targeting new users without any historical information, guiding them towards more popular lodgings.

In the end of this project, I was responsible for ensuring the overall stability and reliability of the project. I managed various project stages and maintained active communication with team members to ensure project integrity. Building upon this responsibility, I completed the production of videos showcasing the project and system.

Learning outcome

Throughout the entire project creation process, I gained a wealth of knowledge related to data processing and the design of recommendation systems. Additionally, I developed a deeper understanding of market dynamics and user requirements. The ultimate presentation of the system was the result of continual communication with team members. Recognizing that generating and documenting new ideas during communication is an essential part of the overall project design, I became more acutely aware of the significance of teamwork.

A high-performing system cannot exist without robust data support. In our case, one of the initial challenges lay in data acquisition. Considering user privacy concerns, much of the lodging information was not publicly available. Consequently, we needed to maximize the use of available data while pinpointing the most relevant information from the vast pool of data. Throughout the data processing, I continuously learned various effective processing methods, experimented with different feature combinations, and flexibly applied various machine learning algorithms. Proficiency in data cleansing, filtering, and related procedures enhanced my sensitivity towards data.

In the design of the recommendation system, the method I was responsible for required aligning with market trends to provide recommendations. Through investigations, I acquired a more in-depth understanding of the market. Furthermore, I learned to transform this understanding into existing data processing, allowing a better comprehension of the practical application and design of recommendation systems. There isn't a universally applicable recommendation algorithm or a unified evaluation standard. A good recommendation algorithm is one that suits the specific design requirements. This uniqueness of recommendation algorithms lies precisely in this characteristic - continuous improvement based on actual conditions leads to better system performance.

Lastly, the comprehensive testing of the project enabled me to oversee the entire project, learning how to design testing procedures from various perspectives and evaluate system stability.

Knowledge and Skill Application

In this project, the knowledge and skills applied can be useful across various professional fields:

1.Data Processing: The practical application of machine learning methods in real data processing is crucial for data and algorithm engineers. Data processing is required almost everywhere, and sensitivity towards it allows for exceptional performance when dealing with complex data handling, providing unique insights. This is extensively applicable in fields like data science and feature engineering.

2.Market Research & Product Positioning: Understanding market needs and user behavior patterns is crucial for product positioning and marketing strategies. Designing recommendation systems and analyzing user behavior enhances customer experience, offering personalized services.

3.Industry Applications: Recommendation system development finds application in various industries, guiding users to suitable choices. It can be used in hospitality and travel industries to enhance booking systems and recommendation engines. Similarly, in e-commerce platforms, it improves personalized recommendations and shopping experiences.

4.Product Testing: Applying testing experience in software development ensures final product quality and stability. Abundant experience helps in comprehensive considerations during development, preventing issues. Additionally, designing test cases and processes within the quality assurance team enhances product quality.

5.Project Management & Team Collaboration: Experience in project management aids in controlling project processes and timelines. A strong risk awareness enables timely identification and resolution of potential problems. Emphasizing communication among team members helps in efficient project management and fosters beneficial ideas during communication.

In conclusion, this project has provided a diverse set of skills and knowledge applicable across various professions, which can be flexibly utilized in various aspects of my future work.

Individual Report: Zhang Yaoxi (A0285851U)

Personal Contribution

In the InnJoy Recommendation System project, my primary contributions revolved around conceptualizing and implementing the interactive chatbot-based user experience. I designed the entire user interaction flow, ensuring seamless communication between users and the system. I constructed the user interaction pages and the registration/login system using HTML, providing an intuitive and visually appealing interface for users to engage with the chatbot and explore recommendations. Furthermore, I leveraged Python's Flask framework to establish robust communication between the frontend and backend, implementing intricate logic for user input processing and feedback generation. Additionally, I integrated both the recommendation algorithm and the chatbot algorithm into the backend, ensuring a cohesive user experience. As the project manager, I orchestrated the project plan, effectively coordinated team members' efforts, and facilitated seamless integration of various components.

Learning outcome

Through this project, I acquired comprehensive knowledge and skills in innovative user interaction methods, specifically in utilizing chatbots to enhance user experience and satisfaction. I gained insights into the design principles and technical implementations of chatbots, learning how to customize their functions and styles based on diverse contexts and objectives. This experience deepened my understanding of user-centric design and honed my ability to create engaging and user-friendly interfaces.

Moreover, I delved into web application development using the Flask framework, mastering its syntax, modules, and operations such as data transmission, route setting, and template rendering. Building the interfaces for both frontend and backend components expanded my proficiency in full-stack development, equipping me with the skills needed to create cohesive and responsive web applications.

Incorporating recommendation algorithms into the project enhanced my expertise in providing personalized services, boosting user retention and conversion rates. I grasped the fundamental principles and classifications of recommendation algorithms, enabling me to select appropriate algorithms based on data features and business requirements. Additionally, I acquired skills in data analysis and visualization techniques, enabling me to evaluate and optimize the effectiveness of recommendation systems.

Maintaining the project's database underscored the importance of data collection and processing in refining recommendation algorithms. I learned techniques to transform unstructured data into structured formats, crucial for effective algorithm implementation. This experience highlighted the significance of data quality and preprocessing in ensuring accurate and meaningful recommendations.

Lastly, serving as a project manager provided me with invaluable skills in effective project management. I developed the ability to create well-planned project schedules, allocate tasks efficiently, monitor progress, resolve issues, coordinate resources, and maintain open communication within the team. These skills enhanced my leadership capabilities and equipped me to manage complex software development projects successfully.

Knowledge and Skill Application

The knowledge and skills acquired during this project are highly applicable in various professional contexts:

1. Industry Applications: I can apply my expertise in chatbot design and recommendation algorithms to industries like e-commerce, customer service, and healthcare, enhancing user engagement and satisfaction.

2. Software Development: I can leverage my full-stack development skills, especially in Flask, to contribute to the creation of robust and user-friendly web applications, ensuring seamless user experiences.

3. Data Analysis and AI: My proficiency in data analysis and visualization techniques equips me to analyze complex datasets and draw meaningful insights, crucial for informed decision-making in data-driven industries.

4. Project Management: As a project manager, I can efficiently plan, execute, and monitor software development projects, ensuring timely delivery and successful project outcomes. My skills in team coordination and communication are vital in any collaborative work environment.

5. Entrepreneurship: I can use my knowledge to develop innovative startup ideas, focusing on user-centric applications and services, applying chatbot technology and recommendation systems to create unique and valuable products for consumers.

In summary, the knowledge and skills gained from this project have equipped me with a diverse skill set applicable across various professional domains, enabling me to contribute effectively to software development, data analysis, artificial intelligence, project management, and entrepreneurial ventures.