From Data to Destination: Using Machine Learning to Plan Your Perfect Trip with Travel Copilot

Yaxuan Deng
rebeccadengyx@qq.com
Beijing Institute of Technology
Beijing, China

Lintian Lei
lintian_lei@qq.com
Beijing Institute of Technology
Beijing, China

Yuchen Xi
xiyuchen0417@gmail.com
Beijing Institute of Technology
Beijing, China

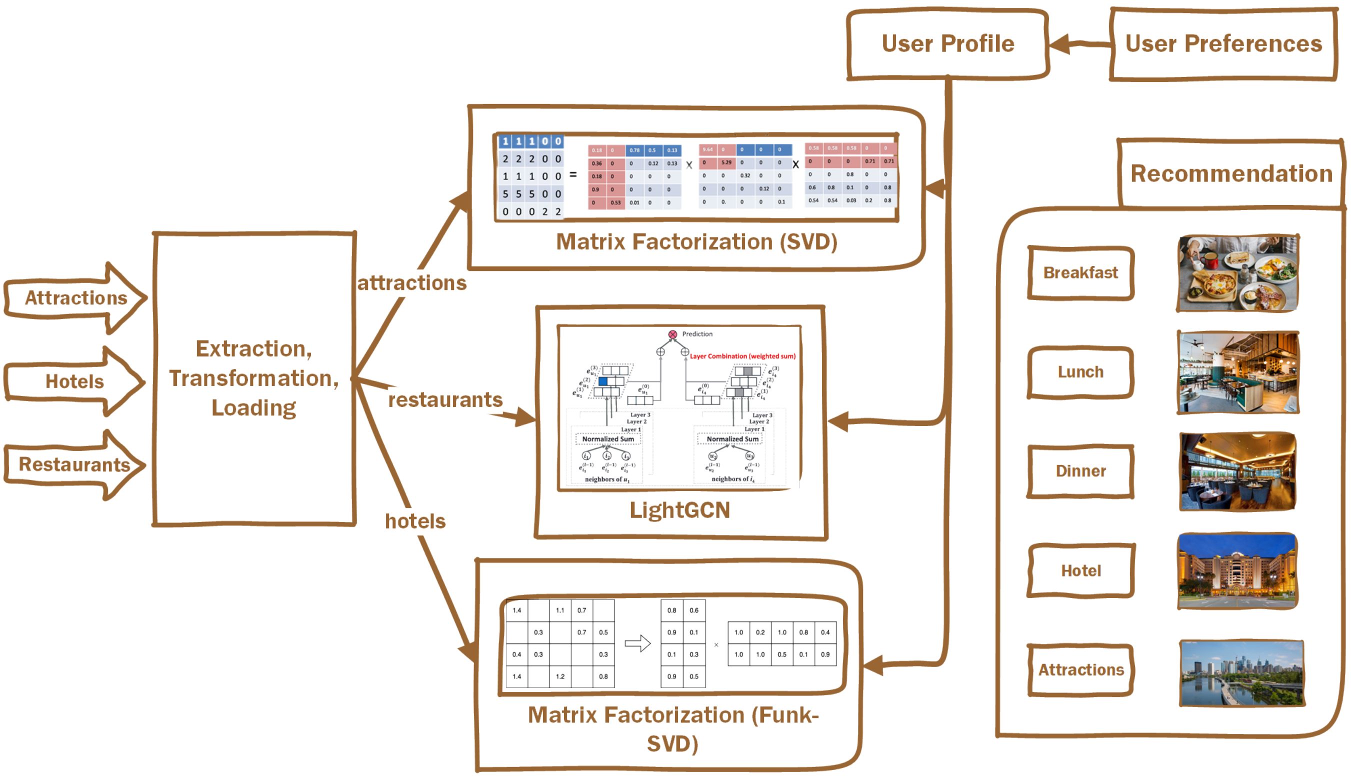


Figure 1: The model consists of three main components: data preprocessing, recommendation generation, and plan generation. In the data preprocessing step, we filter and clean the Yelp dataset to ensure that we're working with relevant, current, and organized data. In the recommendation generation step, we use algorithms to provide recommendations based on the characteristics of hotels, restaurants, and attractions in the Yelp dataset. In the plan generation step, we filter our recommendation results based on user's selected choices to generate target travel plans that are tailored to the user's preferences and inclinations for this trip on our GUI.

ABSTRACT

Traveling can be a fun and exciting experience, but it can be challenging to plan a perfect trip with satisfying attractions, hotels, and restaurants. To simplify this process, we have developed Travel Copilot, a system that utilizes matrix factorization and LightGCN to extract a user's travel preference profile from the Yelp Academic dataset. Based on the user's specific needs for their trip, as well as their historical preferences and factors such as distance, Travel Copilot generates a personalized and optimal travel plan. To ensure high efficiency and accuracy, Travel Copilot employs three different recommendation models for attractions, hotels, and restaurants, respectively. We also discuss the possible future works and improvements of Travel Copilot. The codes are available at https://github.com/Rebabit/travel-copilot-recommender.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; Artificial intelligence.

KEYWORDS

yelp academic dataset, matrix factorization, recommendation, chat-GPT, graph neural network

ACM Reference Format:

Yaxuan Deng, Lintian Lei, and Yuchen Xi. 2023. From Data to Destination: Using Machine Learning to Plan Your Perfect Trip with Travel Copilot. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Overview

Traveling is a wonderful experience, but a truly perfect trip requires a well-crafted travel plan. To ensure the best possible experience, we often find ourselves scouring various review websites before our travels, seeking out businesses with high ratings and the necessary amenities to meet our needs, and making reservations one by one. Additionally, finding the right businesses also involves factoring in distance. For someone who simply wants to unwind during their leisure time, the pain of spending copious amounts of time and energy researching and organizing these details should not be a part of what should otherwise be a comfortable and enjoyable vacation.

To provide travelers with a seamless experience, we have introduced Travel Copilot as a travel assistant in this project. It aims to free people from the heavy burden of travel planning and allow them to immerse themselves in the journey, ultimately allowing travel to reclaim its original meaning in life.

1.2 Problem Definition

The problem we are solving is practical and significant for anyone who is planning a trip. To better understand the importance of this issue, let's imagine the following scenario:

Imagine you are a person with specific preferences who always seeks the ultimate experience. You enjoy exploring new things, experiencing different cultures, and meeting new people. Currently, you are planning a trip to an unfamiliar city known for its beautiful coastline and friendly locals. You are looking forward to spending an unforgettable time there.

To achieve the best experience, you need to do a lot of preparation work. You need to gather information on the city's history, culture, scenery, cuisine, and other aspects. You also need to find the most suitable travel, accommodation, and transportation options for you. Additionally, you need to create a reasonable itinerary to avoid missing any worthwhile places.

Suppose you want to visit a friendly and hospitable seaside attraction, have lunch at the most vibrant local restaurant with coastal specialties at noon, and stay in a comfortable and quiet hotel at night, just like the lodgings you've previously checked-in. In this case, you may have the following concerns:

What are some seaside attractions in this city that are worth visiting? How far apart are they and what is the transportation like? What are their features and stories? What kind of tourists and activities are they suitable for?

What are some coastal specialty restaurants in this city? How are their dishes and prices? Do they have online reviews and recommendations? When are they busiest or least busy?

What are some quiet and comfortable hotels in this city? How are their location and facilities? Do they have any discounts or packages? Do they offer pick-up or guide services?

To solve these puzzles, you may need to go to websites, popular apps to carefully collect information, read reviews - because you may know, if you go directly to browse the local restaurant list by rating, you may find one that is not necessarily a coastal specialty, similarly, the highest rated hotel may be a sea view hotel, but it is not necessarily quiet.

Even if you find the most suitable attractions, hotels and restaurants for you after all the hard work, you may be frustrated to find that they are too far apart, so that the time on the road is so much that it wears away your travel interest. Or you may find that the restaurant or hotel you booked is full or does not match the online description, leaving you disappointed.

All of this may make you, who pursue a perfect experience, give up this trip. After all, travel is not overtime, why bother yourself?

The original purpose of travel is to relax and enjoy different scenery and experience different life after work, not an extension of work.

1.3 Our Solution : Travel Copilot

To solve the troubles mentioned above, we developed Travel Copilot. It is a cleverly designed recommendation system based on the Yelp dataset. It learns various preference information about the target user from their Yelp usage history, including their check-in records, reviews, and ratings. Then it recommends attractions, hotels, and restaurants that suit their unique tastes through diverse recommendation models. In addition, users can also select some special travel needs to further improve the accuracy of the recommendations, such as "whether parking is needed, whether it is suitable for children, whether to bring pets" and so on.

Travel Copilot's recommendation models include two types: matrix factorization-based recommendation models and graph convolutional neural network-based recommendation models, where the matrix factorization-based recommendation models use two different matrix decomposition methods: SVD and funk-SVD. For attractions, we used a matrix factorization recommendation model with funk-SVD; for hotels, we used a matrix factorization recommendation model with SVD; for restaurants, we used a lightweight but more accurate graph convolutional neural network to make recommendations.

To enhance the user's interaction experience with the model, we also designed an additional UI interface based on Python to allow users to enter their travel needs, travel destinations, travel expenses, travel time, etc., and finally display our recommended attractions in a graphic and text way.

Our Travel Copilot showed very good accuracy, fully extracted various potential user feature information contained in the Yelp dataset. It showed the level of human recommendation, and at the same time, because of the complex and simple structure of Travel Copilot's model, its recommendation efficiency is also very high.

2 YELP DATASET AND EXPLORATION

2.1 Yelp Academic Dataset

The Yelp dataset contains a wealth of content. The Yelp dataset is a subset of the business, review, and user data from Yelp, a famous merchant review website in the United States, for personal, educational, and academic purposes. It is provided in JSON format and contains about 150,000 businesses, 7 million reviews, and 200,000 images from 8 metropolitan areas. The Yelp dataset can be used to learn knowledge and skills in database, natural language processing, visualization, and other aspects. It is also one of the most widely used NLP challenge datasets in the world.

2.2 Data Exploration

2.2.1 Businesses. Each business's data consists of multiple fields, such as the business's unique identifier, name, address, city, state, zip code, latitude, longitude, average rating, review count, open status, and categories. These fields can help us understand the basic situation, geographic location, user feedback, and business scope of the businesses.

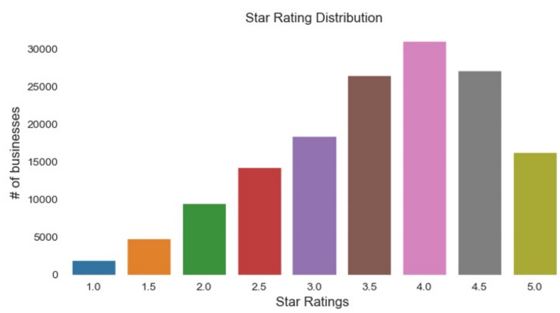


Figure 2: Business Stars Distribution

We counted the number of stores at different scores, as shown in the Fig.2, the number of merchants with a stars of 4.0 is the largest. And we can also find that the restaurants records in the dataset is the most in Fig.3.

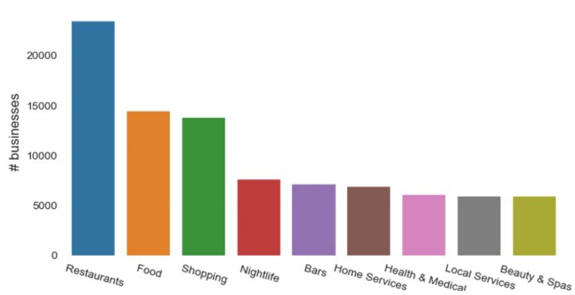


Figure 3: Different types of business

We can also see the distribution of attributes for different types of stores in Fig.**??**.

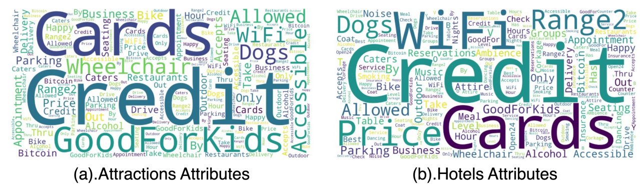


Figure 4: Different business attributes

We can also see the attribute distribution of different types of stores. For example, for attractions, people care more about whether they support wheelchairs and are suitable for children, and for restaurants, whether they support wifi, etc. Shown in Fig.4

2.2.2 Users and reviews. The dataset contains a large amount of user information, which can be used to learn and analyze various aspects of user characteristics, behaviors and preferences. Each user's data consists of multiple fields, such as user's unique identifier, name, number of reviews, join time, number of votes, number of fans, average rating, honor year, number of compliments and so on. These fields can help us understand the user's basic situation, activity level, influence and evaluation style.

The dataset also contains a large amount of review information, which can be used to learn and analyze various aspects of review content, sentiment and impact. Each review's data consists of multiple fields, such as review's unique identifier, reviewer's unique identifier, reviewed business's unique identifier, star rating given by the reviewer, number of votes received by the review and time of review creation and so on. These fields can help us understand the review's basic situation, evaluation quality, user feedback and time distribution.

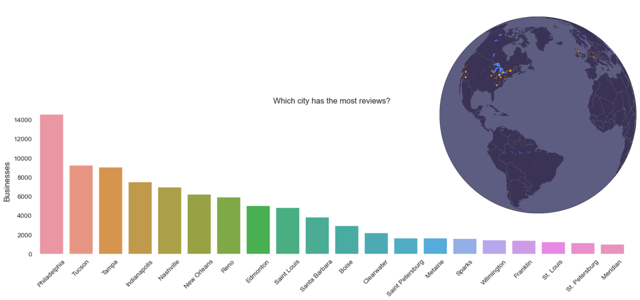


Figure 5: Reviews and Users Distribution

It can be seen that comments and users are mainly distributed in large cities such as the United States and Europe in the Fig.5, and the city with the most comments and users is Philadelphia.

3 DATA PREPROCESSING

3.1 Text Filtering

In the first step of data preprocessing, we focused on filtering the data by selecting only categories related to the attractions, restaurants, and hotel industry. This step was important as it helped to eliminate irrelevant data and ensure that we are working with data that is closely related to our project's objective. By filtering the data, we were able to focus our analysis on data that is most relevant to travelers, such as reviews of restaurants, attractions, and hotels.

- `restaurant_categories`: 'Restaurants', 'Food', 'Sandwiches', 'Fast Food', 'Pizza', 'Coffee & Tea', 'Breakfast & Brunch', 'Burgers'
- `tour_categories`:'Tours', 'Arts & Entertainment'
- `hotel_categories`: 'Hotels', 'Hotels & Travel'

3.2 Data Cleaning

After filtering the data, we proceeded with data cleaning. In this step, we kept only the data with reviews and removed any data that is marked as closed. We did this to ensure that we're working with the most current and relevant information. By removing the closed data, we eliminated data that is no longer relevant to travelers and may not provide valuable insights. This step was essential in reducing noise and ensuring that our analysis is based on the most up-to-date and relevant data.

3.3 Index Maintenance

To keep track of our data, we assigned a continuous row ID for both the businesses and reviews dataset. This step helped us to organize our data and make it easier to access and analyze. By assigning a unique ID to each row in the dataset, we were able to easily link the reviews with the corresponding businesses, making it easier to analyze the data and gain insights.

3.4 Bipartite Graph Construction

Finally, we constructed a bipartite graph. In this step, we created an edge between a user and a restaurant if the user has good reviews for the restaurant. This step was important for further processing and analysis.

4 RECOMMENDATION GENERATION

After preprocessing the data, we select algorithms that can provide recommendations based on the characteristics of hotels, restaurants, and attractions in the Yelp dataset. Specifically, we used Light-GCN for restaurants and SVD for hotels and attractions.

Light-GCN is a state-of-the-art graph convolutional neural network algorithm that is well-suited for capturing the non-linear relationships between users and restaurants. This algorithm is particularly useful when there are large amounts of data, as it can handle sparse and noisy data more effectively than other algorithms.

On the other hand, SVD[3] is a matrix factorization algorithm that is better suited for smaller data sets, such as hotels and attractions in our dataset. It is computationally efficient and well-suited for handling large and sparse datasets, making it a suitable choice for our use case.

4.1 SVD Algorithm

4.1.1 Funk-SVD Algorithm for Hotel Recommendation. Funk-SVD is a matrix factorization algorithm that is widely used in recommendation systems. It is particularly useful for handling sparse matrices, making it an excellent choice for hotel recommendation, where missing data is often a challenge.

In Funk-SVD, we represent users and items (hotels, in this case) by their latent factors. These factors capture the essential characteristics of users and items, and the dot product of the corresponding vectors gives an estimate of the user's preference for a particular item. Funk-SVD decomposes the rating matrix into two matrices, one for user features and one for item features. The algorithm optimizes the values of these matrices to minimize the difference between the predicted and actual ratings.

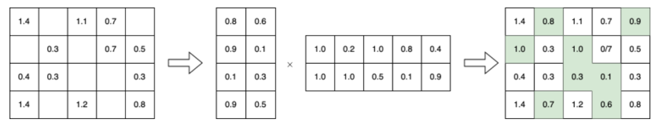


Figure 6: The priciple of Funk-SVD.

One of the unique aspects of Funk-SVD is the use of stochastic gradient descent (SGD) to optimize the model. SGD allows us to update the values of user and item feature matrices incrementally, which is especially useful when dealing with large datasets. Additionally, Funk-SVD allows us to incorporate additional information, such as user demographics or item attributes, into the model.

To use Funk-SVD for hotel recommendation in the Yelp dataset, we first create a user-item matrix, where rows represent users, columns represent hotels, and each cell contains a rating (or a missing value) that a user has given to a hotel. We then apply Funk-SVD to factorize this matrix into two smaller matrices: P and Q.

The matrix P represents user factors, where each row corresponds to a user and each column represents a latent feature. The matrix Q represents hotel factors, where each row corresponds to a hotel and each column represents a latent feature.

Once we have these matrices, we can reconstruct the original user-item matrix by taking the dot product of P and Q. We can then use this reconstructed matrix to predict ratings for hotels that a user has not yet rated. We can recommend hotels that have the highest predicted ratings for a particular user.

4.1.2 SVD Algorithm for Attraction Recommendation. SVD is another popular matrix factorization algorithm that is widely used in recommendation systems. It is particularly well-suited for handling dense matrices, which makes it an excellent choice for attraction recommendation, where we often encounter relatively clean and less sparse data.

In SVD, we represent users and items (attractions, in this case) by their latent factors. These latent factors capture the essential characteristics of users and items, and the dot product of the corresponding vectors gives an estimate of the user's preference for a particular item. SVD decomposes the rating matrix into three matrices, one for user features, one for item features, and one for the singular values. The algorithm optimizes the values of these matrices to minimize the difference between the predicted and actual ratings.

One of the advantages of SVD is its computational efficiency. It is faster than many other matrix factorization algorithms, which makes it an excellent choice for handling large datasets. Additionally, SVD is easily interpretable, which means we can gain insights into the factors that drive user preferences.

To use SVD for attractions in the Yelp dataset, we first create a user-item matrix where rows represent users, columns represent attractions, and each cell contains a rating (or a missing value) that a user has given to an attraction. We then apply SVD to factorize this matrix into three smaller matrices: U, S, and V^T.

The matrix U represents user factors, where each row corresponds to a user and each column represents a latent feature. The matrix V^T represents attraction factors, where each row corresponds to an attraction and each column represents a latent feature. The diagonal matrix S contains the singular values of the original matrix, which capture the importance of each latent feature.

Once we have these matrices, we can reconstruct the original user-item matrix by taking the dot product of U, S, and V^T. We can then use this reconstructed matrix to predict ratings for attractions that a user has not yet rated. We can recommend attractions that have the highest predicted ratings for a particular user.

The differences between Funk-SVD and SVD is as follows:

Table 1: Comparison of Funk-SVD and SVD

Features	Funk-SVD	SVD
Form of Matrix Factorization	user matrix × item matrix	left singular matrix × diagonal matrix × right singular matrix
Method	gradient descent	singular value decomposition
Optimization Goal	minimize the mean squared error of known and predicted ratings	minimize the norm of the original and reconstructed matrix
Advantage	handle missing values directly	optimal decomposed matrix
Disadvantage	no guarantee that the factorized matrix will be optimal	the original matrix is required to have no missing values

4.2 Light-GCN Algorithm for Restaurant Recommendation

LightGCN[1] is a graph convolutional neural network for collaborative filtering recommendation tasks. In recommender systems, collaborative filtering is a commonly used method to make recommendations by analyzing the interaction between users and items. LightGCN adopts a simplified method to remove the complex matrix transformation and nonlinear activation function in traditional GCN, thereby improving the computational efficiency of the model while maintaining the recommendation accuracy.

The reasons why we choose to use LightGCN as the recommendation model are multifold. First, LightGCN adopts a collaborative filtering method based on graph structure, which can capture the complex relationship between users and items, including high-dimensional features and collaborative filtering signals, thereby improving the accuracy of recommendation. Secondly, LightGCN reduces the complexity by simplifying the model structure, so it has better computational efficiency on large-scale data sets. In addition, LightGCN is an unsupervised learning method that does not need to rely on additional labeled data, which is an advantage for real-world recommendation tasks where it is difficult to obtain a large number of labels.

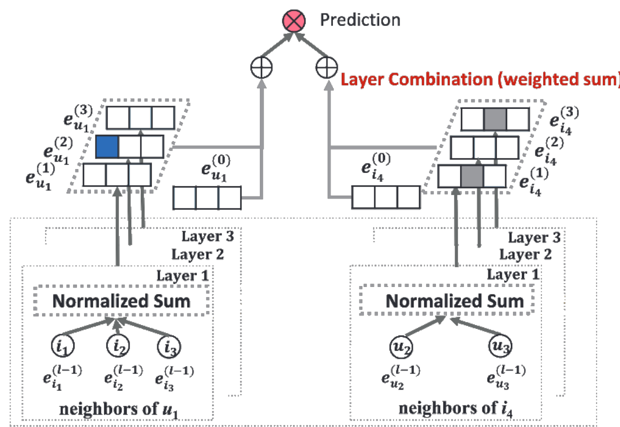


Figure 7: The priciple of Light-GCN.

As shown in Figure 7, the process of using LightGCN for restaurant recommendation on Yelp is as follows:

First, LightGCN randomly generates some vectors as the initial features of each node (user or item). Second, LightGCN will perform multiple iterations. In each iteration, each node (user or project) will get some information from its adjacent nodes (nodes that interact with it) and update its own vector. In this way, the vector of each node will contain more layers of graph structure information. Then, LightGCN will weight and sum the vectors obtained in each iteration to obtain the final features of each node (user or item). Finally, using the final vector, LightGCN can calculate the similarity between any two nodes (users or items), and sort the items according to the similarity, and recommend them to users.

5 PLAN GENERATION

Once we generate recommendations, we give users some options to choose from for their trip, such as preferred travel dates, facilities, and preferred destinations. We filter our recommendation results based on these options to generate target travel plans that are tailored to the user's preferences and inclinations for this trip.

Using the algorithms described above, we generate personalized recommendations that are highly relevant to the user's interests and preferences. By analyzing the user's past reviews and ratings, we can generate recommendations that are unique and personalized to their preferences. This step is essential for providing personalized recommendations to travelers and ensuring that they have a memorable and enjoyable travel experience.

5.1 Attraction Recommendations

Our system provides personalized attraction recommendations based on the user's interests and preferences. By allowing users to select their preferred attraction tags and analyzing their past interests, our system can recommend attractions that are most relevant to their preferences. This ensures that users can explore attractions that match their interests and have an enjoyable experience during their trip.

The recommendation process starts with the user selecting their preferred attraction tags from a wide range of options. These tags can include categories such as historical sites, museums, natural parks, and cultural experiences. The system then analyzes the user's past interests and preferences, along with the selected tags, to generate a list of relevant attractions. The user is presented with three attraction options to choose from, based on their personalized interests and preferences.

By providing personalized attraction recommendations, our system helps users discover new attractions that align with their interests and preferences, enhancing their travel experience and satisfaction. Users can explore attractions that they may not have discovered otherwise, resulting in a more enjoyable and fulfilling travel experience.

5.2 Hotel Recommendations

Our system also provides personalized hotel recommendations based on the user's preferences and requirements. After the user selects their preferred attractions, our system recommends nearby hotels that are convenient for their visit. The recommendations take into account factors such as distance from the attractions, hotel facilities, and user preferences for amenities and price range.

The system recommends three hotels to the user, along with information on their facilities and distances from the chosen attractions. Users can select a hotel based on their preferences, budget, and convenience for visiting their desired attractions. This ensures that users can find suitable accommodation that meets their requirements and enhances their travel experience.

5.3 Restaurant Recommendations

Our system also recommends restaurants for users based on their preferences and requirements. The system first recommends a breakfast restaurant near the user's hotel to ensure they start their day with a satisfying meal. Then, it continues to provide recommendations for lunch and dinner based on the user's preferences for cuisine type and price range.

The system recommends three restaurants for each mealtime, taking into account the user's preferences and convenience for visiting the attractions and hotels. Users can select restaurants based on their culinary preferences, budget, and proximity to their planned activities. This helps users discover local dining options that match their preferences and enhances their overall travel experience.

Collectively, our Travel Copilot system provides personalized recommendations for attractions, hotels, and restaurants, tailored to the user's interests, preferences, and requirements. By leveraging past user data and utilizing recommendation algorithms, our system ensures that users have a memorable and enjoyable travel experience that meets their needs and enhances their satisfaction.

6 IMPLEMENTATION AND RESULTS

6.1 Interactive recommendation system

Our Travel Copilot system interacts with users through a user-friendly interface implemented using PyWebIO, a Python library for building web-based interactive applications. In the Appendix, we provide additional details. Here's how the system interacts with users:

(1) User inputs the number of days for their travel plan: The system prompts the user to input the number of days they plan to travel using a text input field. The user can enter the desired number of days and submit the input.
(2) User fills in personal information and target city: The system asks the user to fill in their personal information, such as name, email, and target city for their trip. The user can input the required information in the respective fields provided by the system.
(3) User selects the type of attractions they are interested in: The system presents the user with a wide range of attraction tags to choose from, such as historical sites, museums, natural parks, and cultural experiences. The user can select the type of attractions they are interested in by clicking on the relevant tags.
(4) System recommends attractions based on user preferences: Based on the user's selections and their past interests, the system generates personalized attraction recommendations. The system displays three attraction options for the user to choose from.
(5) System recommends nearby hotels: After the user selects their preferred attractions, the system recommends nearby hotels that are convenient for their visit. The system provides information on hotel facilities and distances from the chosen attractions. The user can select a hotel based on their preferences, budget, and convenience.
(6) System recommends restaurants: The system recommends restaurants for the user's meals, starting with a breakfast restaurant near the hotel. The system continues to provide more detailed categories, such as cuisine type and price range, for the user to choose from. The system recommends three restaurants for each mealtime, and the user can select restaurants based on their preferences.
(7) System generates travel plans: Once the user has selected one attraction, one hotel, and three restaurants, the system generates a one-day travel plan for the user. If the user has chosen a longer travel duration, the system continues to recommend more travel plans for each day.

On the whole, the usage process of our Travel Copilot involves the user providing input for their travel preferences and requirements, and the system generates personalized recommendations for attractions, hotels, and restaurants accordingly. The user can make selections based on their preferences, and the system generates a tailored travel plan for their desired trip duration. The interactive interface provided by PyWebIO makes it easy for users to input their preferences and receive personalized recommendations, enhancing their travel planning experience.

6.2 Model Training Results

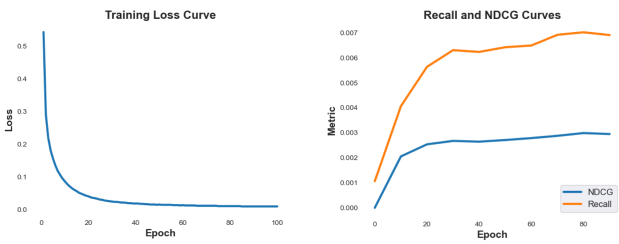


Figure 8: The result of model training.

Based on the training results, visualized in Figure 8, we can analyze the effect of the model from the perspective of loss, recall, and NDCG (Normalized Discounted Cumulative Gain).

Loss: The model shows a decreasing trend in loss over epochs, which is expected in a typical training process. The initial loss is relatively high and gradually decreases as the model learns from the training data. Lower loss indicates that the model is improving in its ability to predict the target outcomes and optimize its parameters.

Recall: The recall values are relatively low initially but gradually increase over epochs. Recall is a measure of the model's ability to correctly identify positive samples. Higher recall values indicate that the model is becoming more effective in identifying relevant items from the dataset.

NDCG: The NDCG values are relatively low initially but show improvement over epochs. NDCG is a measure of the ranking quality of the recommended items. Higher NDCG values indicate that the model is generating better-ranked recommendations, which are more likely to be relevant to users.

Generally, based on the training results provided, the model seems to be improving in terms of loss, recall, and NDCG over epochs, indicating that it is learning from the data and generating better recommendations. Additionally, it would be helpful to compare these results with the performance of other models or baselines to get a better understanding of the model's effectiveness.

7 LEARNINGS FROM THIS PROJECT

Data science is an important discipline in the era of information transformation, which involves multiple aspects of data collection, processing, analysis, mining and application. We have gained a lot of knowledge and improved various abilities through this course project, including:

The ability to use python. Python is a widely used programming language, which has the characteristics of concise, efficient, easy to read and flexible, suitable for various data science tasks. We learned the basic syntax, data structures, functions and modules of python, as well as how to use python to perform data cleaning, transformation and merging operations.

The ability to conduct various exploratory experiments with notebooks. Notebooks are interactive programming environments that allow us to write code, run code, view results and add comments in the same document, which is convenient for us to conduct various exploratory experiments in data science. We learned how to use notebooks to perform data import, exploration, analysis and modeling steps, as well as how to use markdown language to format and typeset documents.

The ability to present data using various visualization tools in python. Visualization is an indispensable part of data science, which can help us better understand data, discover patterns and anomalies in data, convey information and insights from data and so on. We learned how to use various visualization tools in python, such as matplotlib, seaborn, plotly and so on, to create various types of charts, such as line charts, bar charts, scatter plots, box plots, heat maps and so on, as well as how to adjust the style, layout and interaction of charts.

We learned how to fully exploit the geographic information contained in the spatial coordinates of the data, and use this information to improve the performance and user experience of the recommendation system. Secondly, we learned and mastered various recommendation models, including matrix factorization recommendation model, graph convolutional neural network recommendation model, etc., and understood their principles, advantages and disadvantages, and application scenarios. Thirdly, we learned how to remotely develop and deploy machine learning models, and use cloud platforms and tools to achieve efficient collaboration and execution. Finally, we learned how to use Javascript, CSS, Html to develop visualization web pages, and present our recommendation system results to users in an intuitive and beautiful way.

In addition, we also learned how to use PyQT to develop visualization interfaces, which can make it easier for users to use our recommendation system. At the same time, we also improved our team collaboration skills, spatial geographic information processing skills, and cultivated a sense of big data information. It can be said that this project not only exercised our cooperation skills, but also effectively cultivated our data processing skills.

8 FUTURE WORKS

8.1 Federate multiple datasets[2] with spatial coordinates

Geographic coordinates can link various types of key information together, which enables us to combine the Yelp dataset with other sources of data that include geographic coordinates, and thereby enhance the breadth and precision of our recommendations. For instance, we can incorporate data on the spatial distribution of business tax revenue, the profit margins of stores in different locations, and other governmental data to better estimate the spending level of our suggested travel destinations, and accordingly recommend plans that suit the user's budget. Another example is that for the Chicago area, we can access a frequently updated dataset of Chicago crime records from the Chicago Data Work Center, which contains information on the type and location of criminal offenses. By using this dataset, we can effectively assess the safety of our recommended attractions, hotels, and restaurants (which is especially important for travelers with children). Theoretically, any city-related dataset that has geographic spatial information can be integrated into our TravelCopilot system using spatial coordinates as anchor points.

8.2 Chatgpt

The UI interaction has been accomplished, but voice dialogue could also be utilized to perform the data collection and recommendation delivery of the recommendation system. A series of prompts could be devised to enable GPT to elicit the user's specific requirements and preferences for this trip through the dialogue with the user. A well-crafted prompt could extract these specific requirements and preferences as finite, discrete keywords or criteria - this is essentially analogous to our UI interface, except that we allow the user to actively engage in multiple-choice questions. GPT could not only handle data input, but also data output. The information of the recommended attractions, hotels, restaurants, etc., arranged by time and different attributes of the vendors, rationales for recommending them to the user, etc., could be entrusted to GPT to organize into a comprehensive travel plan for the user to hear.

A possible question is, why don't we let GPT do all the work directly? For example, tell it where I want to go and ask it to make a travel plan for me. This is because the current GPT is still a short-lived large language model. Short-lived means that it cannot collect a large amount of user historical travel data to infer the user's travel preferences. Language model means that it will pay more attention to the contextual meaning of the text when outputting, rather than the real features of the attractions mapped by the large amount of attributes and review information contained in the real Yelp dataset. In fact, we believe that the attribute list and the reviews implying its features for each attraction are relatively scattered and low in data volume, making it difficult for GPT to effectively process and learn.

Therefore, on the one hand, you can see the combination of GPT and our recommendation model as our model introducing GPT. On the other hand, you can also see it as GPT introducing our model as a kernel support, shown in Fig.9, where Hard Data means the discrete data that we usually use only in coding, contrary to the natural language. The data of our model is specialized and interpretable. It can make a GPT recommendation system shift from text-centric to data-centric. In this way, we retain both a mathematical, data-based, interpretable recommendation model and the natural attribute of GPT that is easy to communicate with people. This is exactly the advanced form of Travel Copilot that we envisioned.

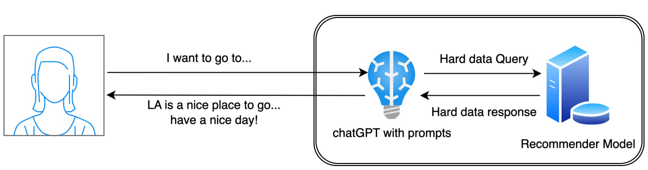


Figure 9: GPT support

So it is possible to bring up with a novel approach to combine GPT and our recommendation model to create a voice-based travel planning system. We can use GPT to elicit the user's travel needs and preferences through dialogue prompts, and to deliver the recommended travel plan in a natural and engaging way using real-data-focused model. We can use our recommendation model to process and learn from the specialized and interpretable data of the Yelp dataset, and to provide data-driven and explainable recommendations for attractions, hotels, and restaurants. By integrating GPT and our recommendation model, we aim to achieve a balance between text-centric and data-centric systems, and to realize our vision of Travel Copilot as a high-level form of travel assistance.

REFERENCES

[1] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[2] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated multi-task learning. *Advances in neural information processing systems* 30 (2017).

[3] Yuefeng Zhang. 2022. An Introduction to Matrix factorization and Factorization Machines in Recommendation System, and Beyond. *arXiv preprint arXiv:2203.11026* (2022).

A APPENDIX SYSTEM DISPLAY

This is the rendering after we run it on the web page.

Welcome to Travel Copilot!🎉

This is a *travel recommendation system*📖.

Fill in the following information according to the guidance.

We will provide you with a customized travel plan✈️.

Step 1: Date Schedule📅

Choose your travel time, we will make a plan for you according to the time.🌞

You will travel for 1 days. 🌛

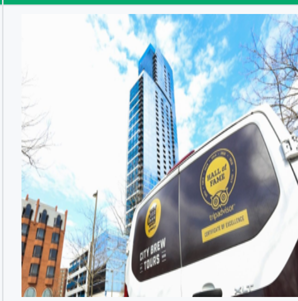
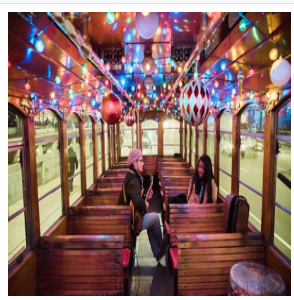
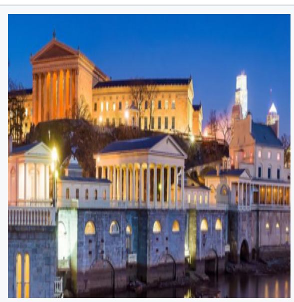
Step 2: Select City🌍

First you need to tell me which city you want to go to?🚗

You will travel to Philadelphia.🏯This is a very good choice.🗺️

Step 3: Choose the attractions you want to visit🌉

There are many recommended attractions, what kind of preferences do you have?⛱️

Name	Address	Stars	Photo
Philly Brew Tours by City Brew Tours	1277 Filbert St	5.0	
Founding Footsteps	302 Arch St	5.0	
The Musical History Tours	6TH St And Market St	5.0	

Here are the recommended attractions, you can choose the ones you like.⛰️

Step 4: Choose the hotel you want to stay.🏨

Next, we recommend some hotels for you, and you can also choose some needs.💒

Name	Address	Stars	Distance	Photo
Chez Colette	120 S 17th St	3.5	0.8 km	
Union League of Philadelphia	140 S Broad St	4.0	0.5 km	
Blue Cross RiverRink Winterfest	101 S Columbus Blvd	4.0	1.8 km	

Here are the recommended hotels, you can choose the ones you like.🏠

Step 5: Choose the restaurants you want to visit. 🍽️

We will recommend some restaurants for you, there are very rich cuisines for you to choose from.🍔

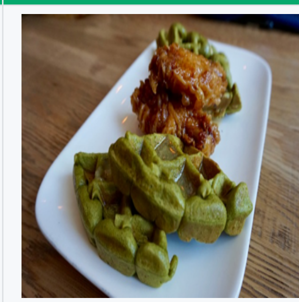
First of all, we recommend some breakfasts near the hotel for you.🥞

Name	Address	Stars	Distance	Photo

Square 1682	121 S 17th St	3.5	0.7 km	
More Than Just Ice Cream	1119 Locust St	3.5	0.6 km	
Marathon	121 S 16th St	3.5	0.6 km	

Here are the recommended breakfasts, you can choose the ones you like.🥓

Next, we recommend some restaurants for you to have lunch and dinner.🍕

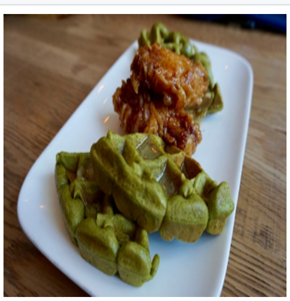
Name	Address	Stars	Distance	Photo
Southgate	1801 Lombard St	4.0	1.3 km	
Honey's Sit-N-Eat	800 N 4th St	4.0	1.8 km	
Copabanana	4000 Spruce St	3.0	3.6 km	

Here are the recommended restaurants, you can choose the ones you like.🍟

Step 6: Display the final plan.📝

Here is your final plan.📝

Plan	Photo	Info
Tour		Name: The Musical History Tours Address: 6TH St And Market St Stars: 5.0
Hotel		Name: Blue Cross RiverRink Winterfest Address: 101 S Columbus Blvd Stars: 4.0
Breakfast		Name: Square 1682 Address: 121 S 17th St Stars: 3.5
Lunch		Name: Southgate Address: 1801 Lombard St Stars: 4.0

Dinner		Name: Honey's Sit-N-Eat Address: 800 N 4th St Stars: 4.0