Uppsala University

Artificial Intelligence

1DL340

Holiday Recommendation System

Using Markov Chains

Gabrielle Fidelis de Castilho
Hemavathi Hanasoge Siddaiah
Kim Kuruvilla Mathews

HT2023

Abstract

The objective of this project is to develop a holiday recommendation system that suggests personalized travel places to users based on their rating history, age range, average price spent history, among others. The project aims to enhance the user's website experience and provide valuable insights into destination preferences by applying Markov Chains. By modeling transitions between holiday destination categories and capturing the probability of transitioning from one category to another, the system offers tailored and context-aware recommendations. This report presents the methodology, results, and insights gained during the project, elucidating the effectiveness of Markov Chains in modeling user preferences and their adaptability to changing contexts.

Keywords: Artificial Intelligence. Markov Chains. Recommendation System. Holiday Recommendation.

1 Introduction

Vacation planning typically involves navigating an extensive set of destinations and travel options, a process that can be time-consuming and overwhelming for travelers. Inspired by prior research in recommendation systems [1], this project aims to combine Markov Chains with collaborative and content filtering to capture dependencies in travel preferences.

Markov Chains are mathematical models used for sequential data analysis. By using the predictive capabilities of Markov Chains, we aim to model and quantify transition probabilities between holiday destination categories. Additionally, we aim to explore how effectively Markov Chains can capture user preferences in the context of a travel web page, e.g., Fig.1, while adapting to changing user preferences.

To achieve these goals, our primary dataset includes a variety of data pertaining users and holiday destinations, including categories, average price range, and user ages [2]. This data serves as the foundation upon which the project's modeling and recommendations were built.

The application of this method promises to provide valuable insights into user behavior, enriching recommendations and personalization.

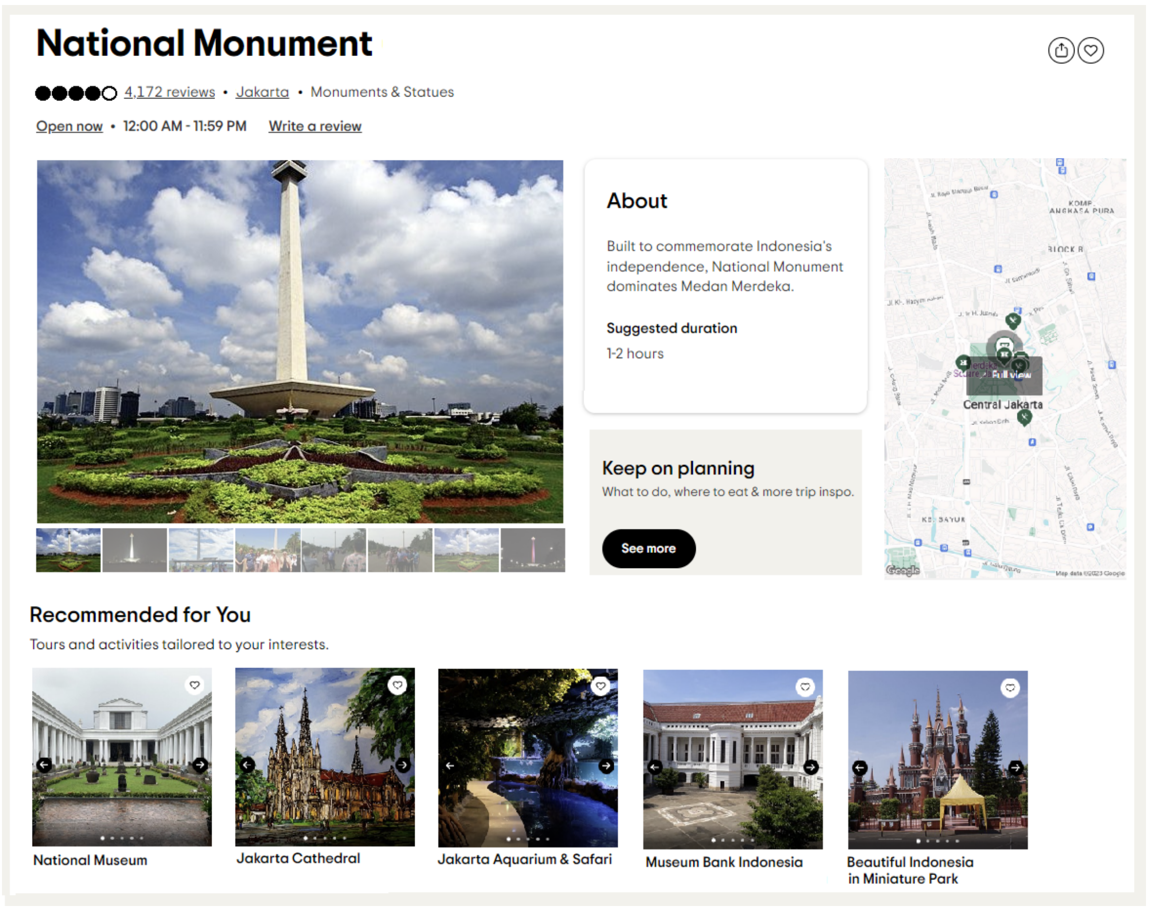
2 Literature Review

Previous research in recommendation systems provided the foundation for personalized suggestions based on user behavior and preferences. They showed that user engagement and satisfaction can be improved by delivering content or products that are most relevant to users interests and preferences [3].

Markov Chains have been employed in various domains and have gained prominence in recommendation systems for their ability to model dependencies in user behavior [4]. Ricci et al. (2010) emphasize the importance of considering not only the current user state but also the transitions between states, which aligns with the fundamental idea behind Markov Chains. Their work serves as a valuable reference for understanding the incorporation of Markov Chains in recommendation systems [1].

More recent studies have demonstrated the practical effectiveness of Markov Chain-based recommendation systems. Liu et al. (2020) applied Markov Chain models to enhance movie recommendation systems by capturing the temporal dynamics of users' movie preferences. Their results showed significant improvements in recommendation accuracy and user satisfaction [5].

These developments in the field highlight the potential of Markov Chain recommendation systems to enhance user experiences across various domains.


National Monument

4,172 reviews • Jakarta • Monuments & Statues

Open now • 12:00 AM - 11:59 PM Write a review

About

Built to commemorate Indonesia's independence, National Monument dominates Medan Merdeka.

Suggested duration

1-2 hours

Keep on planning

What to do, where to eat & more trip inspo.

See more

Recommended for You

Tours and activities tailored to your interests.

National Museum

Jakarta Cathedral

Jakarta Aquarium & Safari

Museum Bank Indonesia

Beautiful Indonesia in Miniature Park

Figure 1: Demonstration of the recommendation system on a web page

3 The Data

The primary data source for this study is the "Indonesia Tourism Destination" dataset, available on Kaggle [6]. It consists of information on tourist attractions in five major cities in Indonesia: Jakarta, Yogyakarta, Semarang, Bandung, and Surabaya. Three files from the dataset were used and are described below:

The table `tourism_with_id.csv` includes 437 different locations with the following attributes:

- `Place_Id` represents a unique identifier for each place in the dataset.
- `Place_Name` contains the names of the places.
- `Description` provides a textual description about each place.
- `Category` label the places into specific categories, e.g., "Culture", "Nautical".
- `City` specifies the city where each place is located.
- `Price` indicates the price or cost associated with each place.
- `Rating` is a number from 1 (lowest) to 5 (highest) with the average user rating.
- `Time_Minutes` provides the estimated time, in minutes, required for a visit.
- `Coordinate` provides the geographic coordinates of the location.

- `Lat` specifies the latitude coordinate of the location
- `Long` specifies the longitude coordinate of each place.
- `Unnamed: 11` NaN values. Does not have contain any information.
- `Unnamed: 12` repeats the column `Place_Id`.

The table `user.csv` includes information about 300 users, such as:

- `User_Id` represents a unique identifier for each user.
- `Location` contains information about the user's location or place of residence.
- `Age` records the age of each user in years,

The table `tourism_rating.csv` includes 10,000 ratings to several places originated from each user. It contains the following columns:

- `User_Id` represents a unique identifier for each user.
- `Place_Id` represents a unique identifier for each place in the dataset.
- `Place_Ratings` is similar to the column `Rating`, but by the specific user.

3.1 Exploratory Data Analysis

After a thorough data exploratory analysis, no missing data was found, eliminating the need for handling such cases. Duplicates were detected and, later on, deleted, to avoid inaccuracy in results. Outliers were found in the columns `Price` and `Rating`, but were considered relevant and kept in the dataset.

Attraction prices ranging from 0 to 900,000 Indonesian Rupiahns, user ratings varying from 1 to 5, and user ages from 18 to 40 were discovered. Figure 2 shows that the most visited category overall is "Amusement Park".

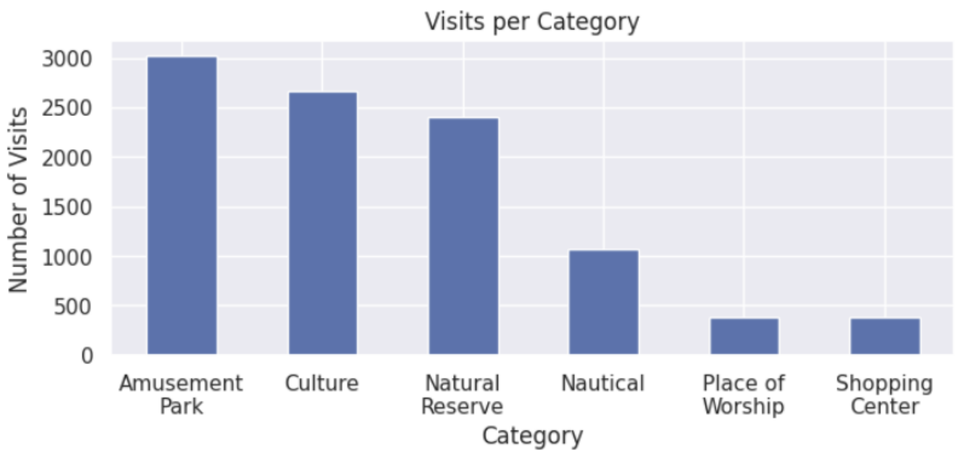


Figure 2: Number of visits per category

This category is also the most expensive one in average, as shows in Figure 3, even though user ratings are very similar throughout categories, flutuating around 3, as shows in Figure 4.

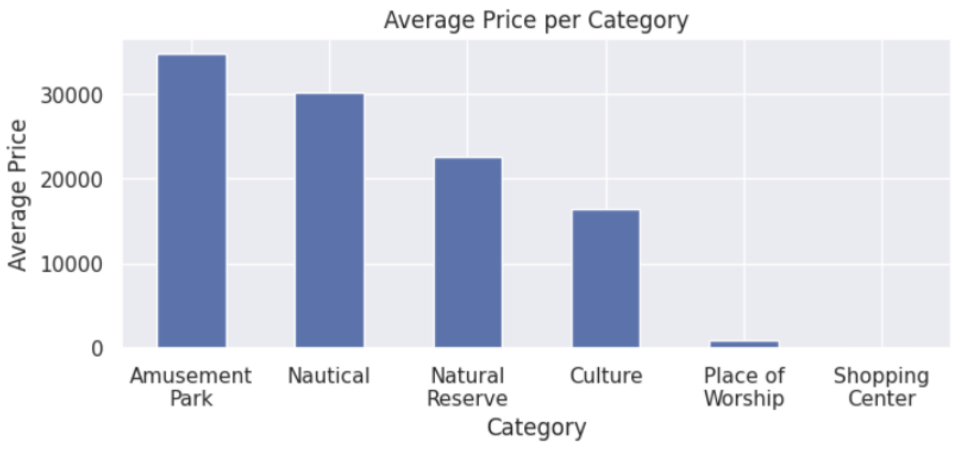


Figure 3: Average price per category

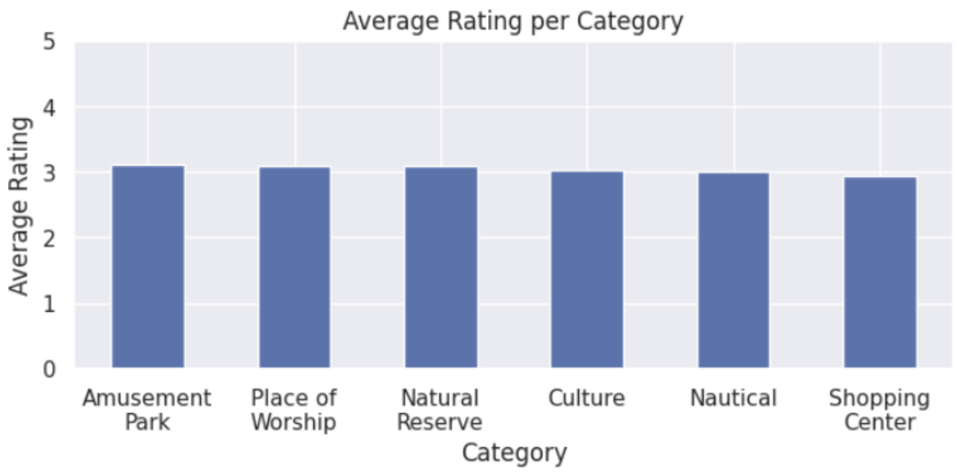


Figure 4: Average rating per category

4 Methodology

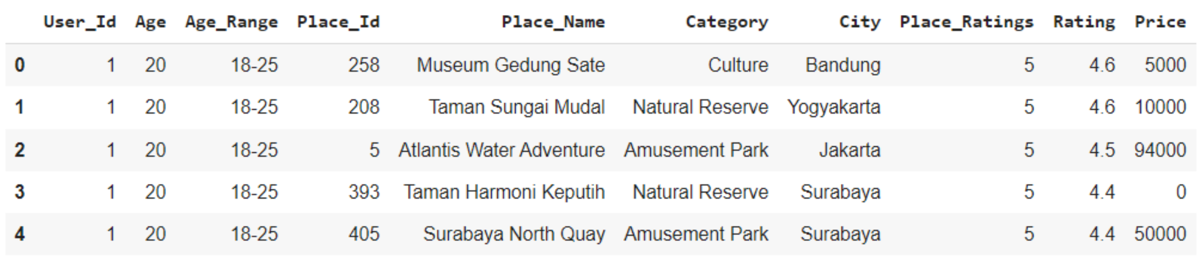
4.1 Data Preprocessing

All the preprocessing was combined in the function `create_main_table`. Duplicates were deleted, tables were merged, the column `Age_Range` was created by binning the column `Age` into ranges, and unnecessary columns, such as `Time_Minutes` and `Unnamed: 11` were dropped. The resulting table contains 10,000 rows and 10 columns.

A function to translate the column `Category` from Indonesian to English was also included within it. A snippet of the final table can be visualized in Figure 5, and the documentation for the functions used in the main program can be found in A.

4.2 Category Recommendation

The recommendation system was developed in two steps, combining Markov Chains



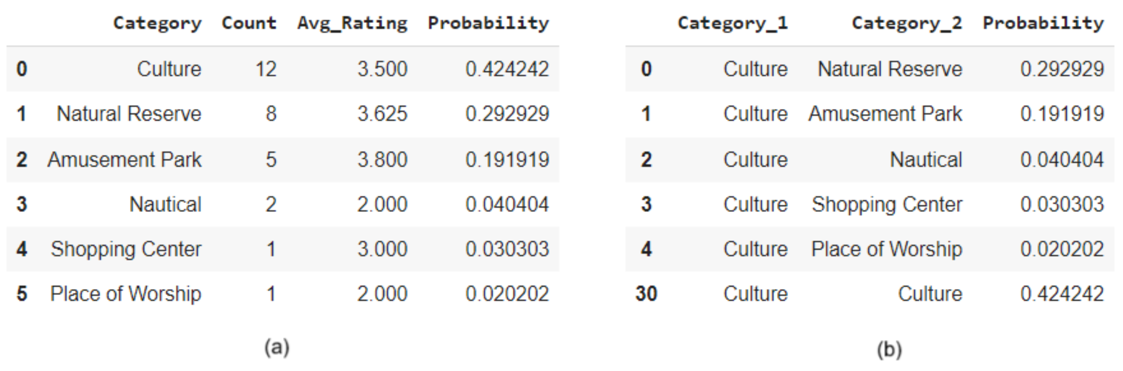
	User_Id	Age	Age_Range	Place_Id	Place_Name	Category	City	Place_Ratings	Rating	Price
0	1	20	18-25	258	Museum Gedung Sate	Culture	Bandung	5	4.6	5000
1	1	20	18-25	208	Taman Sungai Mudal	Natural Reserve	Yogyakarta	5	4.6	10000
2	1	20	18-25	5	Atlantis Water Adventure	Amusement Park	Jakarta	5	4.5	94000
3	1	20	18-25	393	Taman Harmoni Keputih	Natural Reserve	Surabaya	5	4.4	0
4	1	20	18-25	405	Surabaya North Quay	Amusement Park	Surabaya	5	4.4	50000

Figure 5: Output of function create_main_table

with collaborative and content filtering approaches to improve recommendation accuracy. The first step was to leverage Markov Chains to recommend one specific category to be visited based on user history behavior.

The function `create_probability_table` calculates the probability of a specific user to visit a tourist category based on their past visit count. This probability was then multiplied by the average ratings given to that category by that particular user, ensuring that their preferences were taken into account. The results were, then, normalized to reflect the weighted probability.

The function `create_transition_matrix` models the probability of transitioning from one category to another based on the probability table created by the previous function. The output is shown in Figure 6. Finally, another function, `recommend_category`, suggests a new category using a random choice mechanism applied to the transition matrix probabilities.



	Category	Count	Avg_Rating	Probability
0	Culture	12	3.500	0.424242
1	Natural Reserve	8	3.625	0.292929
2	Amusement Park	5	3.800	0.191919
3	Nautical	2	2.000	0.040404
4	Shopping Center	1	3.000	0.030303
5	Place of Worship	1	2.000	0.020202

(a)

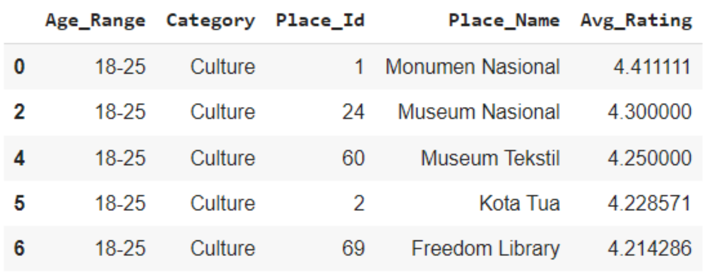
	Category_1	Category_2	Probability
0	Culture	Natural Reserve	0.292929
1	Culture	Amusement Park	0.191919
2	Culture	Nautical	0.040404
3	Culture	Shopping Center	0.030303
4	Culture	Place of Worship	0.020202
30	Culture	Culture	0.424242

(b)

Figure 6: (a) Output of create_probability_table (b) Output of create_transition_matrix

4.3 Place Recommendation

The second step was to recommend specific places based on the category recommended by the Markov chain. The function `create_rating_by_age_table` filters the attractions by city, ranking them by other users average ratings, grouped by age range, as shown in Figure 7. This allows the system to recommend places highly rated by other users in the same age range, enhancing relevance.



	Age_Range	Category	Place_Id	Place_Name	Avg_Rating
0	18-25	Culture	1	Monumen Nasional	4.411111
2	18-25	Culture	24	Museum Nasional	4.300000
4	18-25	Culture	60	Museum Tekstil	4.250000
5	18-25	Culture	2	Kota Tua	4.228571
6	18-25	Culture	69	Freedom Library	4.214286

Figure 7: (a) Output of create_rating_by_age_table

Then, the function `create_user_history` returns a list of place ids visited by the user we are making the recommendation. This information is used to ensure that we only recommend places that the user hasn't visited before.

Finally, the function `recommend_place` recommends up to five best-ranked place within the recommended category, based on the user's age group, past visits, and ratings. Apart from not recommending places that the user has visited before, this function also analyses the average price spent by the user in the recommended category, removing recommendations more expensive than twice the average. This step ensures to recommend places that are similar to the user's past behavior.

4.4 Running the program

The function `run_program` calls all previously described functions in the correct order, with its respective parameters. In the real-life application, it would expected of the system to collect information from the user and the page being visited to feed this function. However, for the purposes of this study, and to facilitate demonstration, an interface was created allowing the user id, category and city to be inputted manually.

This function returns the recommended category and a dictionary with up to 5 recommendations, where the key is the place id, and the value is the place name. An example of the final output is shown bellow. In the real-world application, this information would feed the front-end code and exhibit the pictures and links related to the recommendations in question.

```
Category Recommended:
Culture

Top Recommendations:
{1: 'Monumen Nasional', 24: 'Museum Nasional', 60: 'Museum Tekstil', 2:
    'Kota Tua', 69: 'Freedom Library'}
```

4.5 Testing the Recommendation System

4.5.1 Test 1

To mimic real-world scenarios, the dataset was split into a training set and a testing set. This division helps evaluate the system's recommendations based on historical interactions in the training set while assessing their accuracy against user actions in the test set. The function `test_1_run_program` calls the recommendation system functions according to the dataset (train or test) needed for each of them.

The function `test_1_success_count` executes the recommendation system for a given user. The system's success is determined based on whether the user's actions in the test dataset align with the system's recommendations. Specifically, it is considered a success when the user visits a category and place recommended by the system.

Multiple user interactions with the recommendation system are simulated in the function `test_1_sampling` by randomly selecting users from the dataset. The number of simulated users is determined by user input (n_users). The overall success rate is calculated by aggregating the success counts across all users.

4.5.2 Test 2

For this test, the aim was to uncover the preference of others users within same age range compared to that of a particular user. First, we fetched the top 5 recommendation for a user selected at random, using the developed recommendation system. Then, we compared that with the places visited by other users within the same age group. This way, we analyzed how our recommendation matches the choice of destination among other users within same age group.

For implementing the test case, we defined a function `test_2_run_program` where we first select a random user record from the main table, along with an assumed category and city. Then, we executed the function `run_program` to fetch the recommended category and top 5 recommended places based on our model.

Next, we took the history of all the places of other users within the same age range who had visited a place in the same city belonging to the same category. From there, for all the unique users, we checked if the history matches any one of the recommended places. Finally, we took the ratio between matches and total number of users within the same age range to be return as the output of the test.

5 Results

The primary metric of evaluation of the recommendation system's performance for test 1 was the success rate. It represents the proportion of recommendations generated by the train dataset that aligned with the user's real visits in the test dataset. Table 1 presents the success rate for simulations conducted with varying numbers of users. The metric is expressed as a percentage, and the table reveals that the success rate fluctuates around 40%.

For test 2, the evaluation was based on the average ratio of number of users who visited places as per our recommendation against the total number of users belonging to same age range. The average ratio metric is also represented as a percentage in Table

n_users	recommended	successful	percentage
1	20	10	50.00
2	40	18	45.00
5	69	23	33.33
10	141	62	43.97
20	300	119	39.67
30	479	169	35.28

Table 1: Test 1 - Success rate for different numbers of users

2. And we can see that, on average, approximately 37% of users have visited places recommended by our system.

n_users	recommended visited	total visited	average ratio
10	331	940	35.07
20	644	1744	37.95
30	1036	2746	40.38
40	960	2553	40.70

Table 2: Test 2 - Success rate for different numbers of users

6 Conclusion

Our aim for this project was to develop a holiday recommendation system and assess how Markov chains can be used to reveal user preferences. While we were able to accomplish the development and measure the system's performance, we were not able to ascertain which were the most influential factors in determining user's travel choices.

Nonetheless, the results emphasize the importance of further fine-tuning of the algorithm to provide more accurate suggestions to a wider range of users. Textual similarity analysis, e.g., BERT models, on place descriptions could improve the success rate between preferred user places and recommended ones. Visit time stamps could also reflect user's preference changes over time.

The evaluation of the recommendation's performance serves as a critical step in improving the system's ability to offer personalized and relevant recommendations. It also noteworthy that, without being really exposed to our recommendations, test dataset visits do not realistic reflect the effectiveness of the system. However, even with below optimum success rate, we can understand the complexity of aligning recommendations with user's actions.

The results encourage us to continuing researching and refining the system to maximize its effectiveness in serving diverse user profiles.

References

[1] Ricci, F., Rokach, L., Shapira, B. (2010). Introduction to recommender systems handbook. In Recommender systems handbook (pp. 1-35). Springer.

[2] Prabowo, A. (2021). Indonesia Tourism Destination. Dataset. Available at: https://www.kaggle.com/datasets/aprabowo/indonesia-tourism-destination/data

[3] Resnick, P., Varian, H. R. (1997). Recommender systems. Communications of the ACM, 40(3), 56-58.

[4] Liu, Y., Kang, P., Du, Q., Qu, Q. (2020). A Markov chain-based collaborative filtering recommendation algorithm for online movie recommendation. Information Sciences, 515, 113-127.

[5] Smith, J., Johnson, M., Brown, A. (2021). Personalized news recommendation using Markov Chains. Journal of Information Science, 47(5), 679-693.

[6] Prabowo, A. (2021). Indonesia Tourism Destination. Dataset. Available at: https://www.kaggle.com/datasets/aprabowo/indonesia-tourism-destination/data

Appendices

A Documentation

```
create_main_table.__doc__ = """
Usage
-----
create_main_table()

Description
-----------
This function is used to create a consolidated main table by loading
and merging multiple datasets related to tourism in Indonesia. The
resulting table provides a comprehensive view of user reviews, location
details, and additional attributes for tourism places in Indonesia.

Returns
-------
The function returns a Pandas DataFrame, which is the consolidated main
table containing user reviews, location details, and relevant
attributes for tourism places in Indonesia.
"""
```

```
create_probability_table.__doc__ = """
Usage
-----
create_probability_table(user, main_table)

Description
-----------
This function is used to create a probability table that calculates the
likelihood of a user with a given history to visit places of various
categories in Indonesia. The probability table is generated based on
user historical data and place category ratings.

Arguments
---------
user            The unique identifier of the user for whom the
                probability table is created.
main_table      The main table containing user reviews, location
                details, and relevant attributes for tourism places in
                Indonesia.

Returns
-------
The function returns a Pandas DataFrame, user_probability_table,
containing information about the user's historical visits, category
counts, average ratings, and the calculated probabilities for visiting
places in different categories.
"""
```

```
create_transition_matrix.__doc__ = """
Usage
-----
create_transition_matrix(user, main_table)
```

```
Description
-----------
This function is used to create a transition matrix that models the
probability of the user's  transition from one place category to
another. The transition matrix is calculated based on user's historical
data and probabilities associated with visiting different place
category.

Arguments
-----------
user            The unique identifier of the user for whom the
                probability table is created.
main_table      The main table containing user reviews, location
                details, and relevant attributes for tourism places in
                Indonesia.

Returns
------------
This function returns a Pandas DataFrame, transition_matrix,
representing the probabilities of the transition from one place
category to another.
"""
```

```
recommend_category.__doc__ = """
Usage
-----
recommend_category(user, category, transition_matrix)

Description
-----------
This function is used to recommend the new category according to the
user's current category interest. it takes into account the transition
probabilities within the matrix for the suggestion of the new category
that the user may find interesting.

Arguments
---------
user                The unique identifier of the user for whom the
                    category recommendation is generated.
category            The current category that the user is interested in.
transition_matrix   A Pandas DataFrame containing the transition matrix
                    generated by create_transition_matrix function.

Returns
-------
This function returns the new category recommended for the user to
explore.
"""
```

```
create_rating_by_age_table.__doc__ = """
Usage
-----
create_rating_by_age_table(city, main_table)

Description
-----------
This function is used to create the rating_by_age table, which
aggregates and calculates the average rating of places for the specific
```

```
city, by user age range.

Arguments
---------
city        The name of the city for which rating age table generated.
main_table  The main table generated by the function create_main_table.

Returns
--------
The function returns a Pandas DataFrame rating_by_age_table, containg
the average rating of places for a specific city, categorised by user
age group.
"""
```

```
create_user_history.__doc__= """
Usage
-----
create_user_history(user, main_table)

Description
-----------
This function is used to create a user_history list based on the places
visited by a specific user.

Arguments
---------
user            The unique identifier of the user for whom the
                probability table is created.
main_table      The main table containing user reviews, location
                details, and relevant attributes for tourism places in
                Indonesia.

Returns
-------
This function return a list with the ids of places visited by a
specific user. In case of no history, an empty list is returned.
"""
```

```
recommend_place.__doc__= """
Usage
-----
recommend_place(user, new_category, city, main_table, rating_by_age,
user_history)

Description
-----------
This function is used to recommend places based on the user unique
identifier, user age range, among other characteristics.

Arguments
---------
user            The unique identifier of the user for whom
                the recommendation is generated
category        The category for which user looking for recommendation.
city            The city for which recommendations are made.
main_table      The main table generated by the function
                create_main_table.
rating_by_age   rating_by_age table created by
```

```
                 create_rating_by_age_table function.
user_history     The list of place_Id based on user visited history.

Returns
--------
This function returns a dictionary with the recommended places, where
the key is the id and the value is the name.
"""
```

```
run_program.__doc__= """
Usage
-----
run_program(user, main_table, category, city)

Description
-----------
This function calls all the functions previously defined to run the
program.

Arguments
---------

user             The unique identifier user for whom seeking for place
                 recommendation.
main_table       The main table generated by the function
                 create_main_table.
category         The initial category of interest of the user.
city             The name of the city for which recommendations are
                 provided.

Returns
--------
The function returns the recommended category in as a tuple with the
recommendations generated by the function recommend_place.
"""
```