A Tourist Destination Recommender System for Nepali Tourism Industry

Manohar Dahal[1*], Dipin Mainali[2], Aman Shrestha[3], Bibek Joshi[4] and Suramya Sharma Dahal[5]

[1]Department of Electronics and Computer Engineering, Thapathali Campus, IOE, TU, email: manohardahal40@gmail.com

[2] Department of Electronics and Computer Engineering, Thapathali Campus, IOE, TU, email: nikeshmainali321@gmail.com

[3] Department of Electronics and Computer Engineering, Thapathali Campus, IOE, TU, email: amanshrestha147181@gmail.com

[4] Department of Electronics and Computer Engineering, Thapathali Campus, IOE, TU, email: Joshibibek5@gmail.com

[5]Associate Professor, Department of Electronics Engineering, Kathmandu Engineering College, email: suramya.sharma@kecktm.edu.np

***Abstract*— Tourism is one of the major sources of revenue in developing countries with natural beauty like Nepal. Understanding tourist preferences is essential to optimize this industry. Many places suffer obscurity due to not being recommended to the tourists who would've liked the destination. Our goal is to analyse what every tourist likes and recommend them accordingly. In case of Nepal, since there are a lot of under-explored potential tourist destinations and not many data available for the reviews and ratings, the data becomes sparse and limited. To tackle the challenge, our system utilizes a blend of content-based and collaborative filtering, which is known as hybrid filtering. The system leverages the strengths of Cosine Similarity, k-Nearest Neighbors, and Matrix Factorization to personalize recommendations based on user preferences and available destination information even in a sparse as well as limited dataset. This approach not only tackles the under-exploration of hidden gems but also presents a scalable framework applicable to any domain with limited data, potentially impacting personalized recommendations across various industries.**

***Keywords* — Recommender System, k-Nearest Neighbours, Matrix Factorization, Cosine Similarity, Personalization, Tourist destinations, content-based filtering, collaborative filtering, hybrid filtering**

I. INTRODUCTION

The tourism industry has experienced significant growth in recent years. This surge can be partly attributed to the rapid development of communication and information technology across the globe. The widespread use of the internet has simplified the process for potential tourists to access vast amounts of global data on points of interest, travel plans, and destinations. E-tourism thrives in both the social and economic sectors, and this is true for Nepal as well.

Though a small country, Nepal is blessed with breathtaking natural beauty. The tourism industry can be further flourished, bringing additional benefits to the country. Every year, millions are generated from tourism, along with the advantages of exposure to the world, globalization, and cultural exchange with foreigners.

For any traveler, foreign or local, the first step to visiting a place is to be aware of its existence and its potential to be a worthwhile destination. Tourists have different preferences that shape how they rate a destination. If tourists don't know about a place they might love and spend time in, it's a loss to the tourism industry.

Unsurprisingly, there are many websites and online platforms offering travel packages that manage tours from start to finish. However, these platforms often follow trends and prioritize profit, neglecting to create packages for lesser-known destinations that are more challenging to access due to potentially lower booking numbers. Additionally, many tourists prefer independent travel, not wanting their tours managed.

Recommending destinations can be difficult even for local travelers, for two reasons. First, it's challenging to accurately assess what a tourist might like or dislike based solely on their background. Second, even local travelers

**Corresponding Author*

have limited knowledge, often only familiar with places they've visited or heard about. Recommendations based solely on personal experiences can leave a confused foreign tourist with limited time struggling to navigate mixed reviews from strangers. They have no way of knowing whose preferences align with their own.

To address all these challenges, a well-designed automated recommender system is essential. Within the realm of e-tourism, a recommender system stands as a powerful tool for bridging the information gap between tourists and the abundance of potential destinations. This technology can be described as a subclass of information filtering system, designed to analyze user data and preferences to generate personalized recommendations for items – in this case, tourist destinations. Similar to domain-specific recommender systems used by e-commerce platforms and movie streaming services, Nepal's e-tourism industry can leverage these intelligent systems to analyze tourist data and curate recommendations that perfectly align with each visitor's unique travel style and interests. By implementing recommender systems, Nepal's e-tourism can not only empower tourists with a streamlined decision-making process but also promote lesser-known destinations, fostering a more balanced and enriching travel experience for all.

II. RELATED WORKS

We are not the first to note the benefits of combining collaborative and content-based filtering. Recommender systems have been applied in many fields. In 2006, Netflix announced a contest to improve its recommender system, releasing a training set of over 100 million ratings from 500,000 anonymous customers on 17,000 movies [1]. Participants submitted predicted ratings for a test set of about 3 million ratings, with Netflix calculating a root-mean-square error (RMSE) against the held-out truth. The first team to improve the RMSE by 10% would win $1 million. If no team reached this goal, a $50,000 Progress Prize was awarded annually to the leading team. On September 21, 2009, the $1 million grand prize was awarded to the BellKor's Pragmatic Chaos team, which achieved a 10.06% improvement over Netflix's algorithm [2].

[3] describes the different methods of computing similarities between item vectors like cosine similarity, Euclidean distance and Pearson's coefficient of correlation. The item vectors can be both the movies or the users which are called user based or item based collaborative filtering respectively. If A and B are the vectors in n-dimensional hyperspace, then the cosine similarity between them can be calculated as

$$\text{cosine similarity} = Sc(A,B) = \cos(\theta) = \frac{A.B}{|A|.|B|} \quad \text{..... (1)}$$

Correlation between two users x and y can be measured by computing the Pearson correlation which measures the extent to which two variables linearly relate with each other. For the user-based algorithm, the Pearson correlation between users x and y is

$$r = \frac{\sum(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum(X_i-\bar{X})^2 \sum(Y_i-\bar{Y})^2}} \quad \text{.....(2)}$$

where the i ϵ I summations are over the items that both the users x and y have rated and $\bar{x}$ and $\bar{y}$ is the average rating of the co-rated items of the x^{th} and y^{th} user.

[4] describes the k-NN and Naïve Bayes classifier algorithm for text and document mining. Bayesian classifiers are based on a statistical principle. Each processed term is assigned a probability that it belongs to a certain category. This probability is calculated from the occurrences of the term in the training documents where the categories are already known. Given a set of r document vectors $D = \{d_1, \ldots, d_r\}$, classified along a set C of q classes, $C=\{c_1, \ldots, c_r\}$, Bayesian classifiers estimate the probabilities of each class ck given a document dj as:

$$P(c_k \mid d_j) = \frac{P(d_j|c_k)\cdot P(c_k)}{P(d_j)} \quad \text{.....(3)}$$

Where, P(dj) is the probability that a randomly picked document has vector dj as its representation, and P(ck) the probability that a randomly picked document belongs to vector ck.

The k-Nearest Neighbor (k-NN) algorithm, developed by Evelyn Fix and Joseph Hodges in 1951 and expanded by Thomas Cover, is a non-parametric method for classification and regression. It uses the k closest training examples as input. For continuous variables, Euclidean distance is commonly used, while for discrete variables,

metrics like Hamming distance are used. In gene expression microarray data, k-NN employs correlation coefficients such as Pearson and Spearman as metrics.

[5] describes Stochastic Gradient Descent (SGD) as an iterative algorithm commonly used in matrix factorization and machine learning. SGD optimizes the loss function by applying gradients to the factor matrices. The update rules for the matrices W and H are:

$$W_i = W_i - \alpha\left(e_{i,j}H_j - \lambda W_i\right) \quad \ldots\ldots\ldots(4)$$
$$H_i = H_i - \alpha\left(e_{i,j}W_j - \lambda H_i\right) \quad \ldots\ldots\ldots(5)$$

where $e_{i,j}$ is the loss (or error) calculated for the rating $r_{i,j}$ in the current iteration. α is the learning rate that can be selected different for each factor matrix, and λ is the regularization parameter used to avoid overfitting. In the tourism industry of China, [6] has implemented a web-based tourism recommender system consisting of methods like web-crawlers, collaborative filtering and cosine similarity for recommending tourism sites to the users. Cold start problem in recommendation [7] can be item based and user-based, item based refers to addition of new item in the system and user based refer to addition of new user and it is discussed with the solution. [8] describes how we can combine both content based and collaborative filtering techniques. It discusses cosine similarities and Karl Pearson's correlation coefficient as a medium for mapping of different users.

III. METHODOLOGY

A. Data Collection

Data collection posed the main challenge, requiring information on various places along with their associated genres and keywords. These were gathered from reviews across blogs, websites, and social media. Our database now includes 503 distinct places from all 7 provinces of Nepal. Each place received a unique ID and was classified into genres, potentially spanning multiple genres. For instance, a place might be both adventurous and religious. Keywords were also assigned based on the place's characteristics and notable features. Genres and keywords were then converted into binary format and combined to create the "parameters" for each place. These parameters served as vectors with n dimensions, facilitating distance calculations within the recommender system.

For content-based recommendation, we didn't need to collect ratings beforehand, as it relied solely on the characteristics of the items. However, for collaborative filtering, we lacked sufficient data as there were no existing ratings prior to our website's launch. To address this, we conducted a survey to gather ratings. We created 10 samples, each containing 50 random places from our database, and distributed them to different groups of individuals. They provided ratings for the places they had visited. In total, we collected ratings from about 100 users before implementing the collaborative filtering algorithm.

B. Recommendations

Recommendations were central to this project, necessitating an understanding of users' moods or interests. Recommending content outside of a user's interest renders the system unusable. We identified two types of users in the system.

1) New users: The challenge with recommending places to new users arises from their lack of ratings for existing places. To address this, we collect user preferences during their initial login. These preferences consist of a list of binary values corresponding to the total number of genres available. This list serves as a 7-dimensional vector used to calculate cosine similarity between each place and the user's preferences. Prior to this calculation, the genres of each place are converted into a binary list called 'genre_bin'. The dataframe just before computing the initial recommendations reflects this setup.

Table 1: dataframe after assigning genre_bin

pID	pName	culture	adventure	wildlife	sightseeing	history	religious	child_friendly	tags	province	genre_bin
1	Satasidham	1	0	0	1	0	1	0	waterfall, pond, garden, cave , hindu , temple...	1	[1, 0, 0, 1, 0, 1, 0]
2	Arjundhara Dham	1	0	0	1	1	1	0	hindu, temple, pond, gurukul, farm	1	[1, 0, 0, 1, 1, 1, 0]
3	Kichakavadh	1	0	0	1	1	1	0	hindu, pond, garden, temple, castle remnants	1	[1, 0, 0, 1, 1, 1, 0]
4	Biratpokhar	0	0	0	1	1	0	0	hindu, pond, garden , boat ride	1	[0, 0, 0, 1, 1, 0, 0]
5	Krishnathumki	1	0	0	0	1	1	0	hindu , temple , hills , forest	1	[1, 0, 0, 0, 1, 1, 0]
...	...	...	...	...	...	...	...	...	...	...	...
499	Jhilmila lake	0	0	0	1	0	0	0	lake	7	[0, 0, 0, 1, 0, 0, 0]
500	Chadani bridge	0	0	0	1	0	0	0	river	7	[0, 0, 0, 1, 0, 0, 0]
501	Ghoda Ghodi lake	1	0	1	1	0	1	0	lake,forest, wetland , hindu, temple	7	[1, 0, 1, 1, 0, 1, 0]
502	Godawari Ram Temple	0	0	0	0	0	1	0	hindu, temple	7	[0, 0, 0, 0, 0, 1, 0]
503	Karnali Bridge	0	0	0	1	0	1	0	river, landmark , picnic spot , dating spot , ...	7	[0, 0, 0, 1, 0, 1, 0]

Cosine distance between the genre_bin and user's preferences is calculated and is sorted in ascending order (from nearest to farthest) and top 30 of the result is shown in the home-page.

2) Existing users: For existing users, a hybrid approach of content-based and collaborative filtering is employed. In **content-based filtering**, tags are converted into binary vectors and combined with genre information to create parameters for each place. A user's profile is generated based on their rated places, and a weighted profile for each place is computed using the rating data.

In this system, a rating of 3 represents neutral preference, while 4 and 5 indicate liked and 2 and 1 indicate disliked. Ratings are adjusted by subtracting the average rating (3) from the user's ratings. These adjusted ratings are then averaged within each parameter to calculate the weighted profile. The resulting weighted profiles are assigned to each place based on their parameters, which are then converted into vectors or lists.

Table 2: sample ratings provided by a user

pID	pName	rating	profile
248	Upper mustang trek	5	[1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, ...
253	Kali Gandaki valley trek	4	[1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, ...
429	Shey Phoksundo National Park	5	[1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, ...
276	Mahendra Cave	2	[0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, ...

Table 3: Sample dataframe after weighted profile is calculated for each place

pID	pName	culture	adventure	wildlife	sightseeing	history	religious	child_friendly	tags	province	genre_bin	init	profile	weighted_profile
32	Urlabari, jhapa	1	0	0	0	0	0	0	village	1	[1, 0, 0, 0, 0, 0, 0]	1.0	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1.6666666666666667, 0.0, 0.0, 0.0, 0.0, 0.0, ...
57	Diktel	1	0	0	0	0	0	0	village, hills	1	[1, 0, 0, 0, 0, 0, 0]	1.0	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1.6666666666666667, 0.0, 0.0, 0.0, 0.0, 0.0, ...
58	Dole, Nepal	1	0	0	0	0	0	0	hills, village, homestay	1	[1, 0, 0, 0, 0, 0, 0]	1.0	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1.6666666666666667, 0.0, 0.0, 0.0, 0.0, 0.0, ...
65	Thechambu	1	0	0	0	0	0	0	hiking, village, hills	1	[1, 0, 0, 0, 0, 0, 0]	1.0	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...	[1.6666666666666667, 0.0, 0.0, 0.0, 0.0, 0.0, ...
68	Phidim	1	0	0	0	0	0	0	City, hills, river	1	[1, 0, 0, 0, 0, 0, 0]	1.0	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,	[1.6666666666666667, 0.0, 0.0, 0.0, 0.0, 0.0, ...

We calculate 'k' neighbors for each place in our database based on the rated places, with k determined as the square root of the number of rated places. If k is a float, we convert it to an integer and ensure it's odd if originally even. From these neighbors, we determine the frequency of liked and disliked places, considering ratings greater than or equal to 3 as liked and otherwise as disliked. This classification results in places being categorized as 'likeable' or 'dislikeable'. We filter out the 'likeable' places and compute the cosine similarity of each with the user's profile for further processing.

$$\text{cosine similarity} = \frac{\mathbf{i} \cdot \mathbf{j}}{|\mathbf{i}| \times |\mathbf{j}|} \ldots\ldots(6)$$

Where ‘i’ is the user’s profile and ‘j’ is the places’ profile.

In **collaborative filtering**, recommendations are based on a feedback matrix derived from user activity. Matrix factorization helps uncover latent features by multiplying different entity types, such as items and users. This technique, applied to collaborative filtering, aims to discern the relationship between items (places) and user entities using their ratings. The feedback matrix, sized |U| x |D|, captures all user ratings on places. The objective is to uncover K latent features.

With K = 7, matrices P (|U| x k) and Q (|D| x k) represent user-feature and item-feature associations, respectively. The product of these matrices yields R, predicting user ratings for places. The prediction is achieved by computing the dot product of the user and place vectors corresponding to u_i and d_j.

Initially, matrices P and Q are randomly initialized, and their product difference, termed M, is calculated. Stochastic gradient descent is then employed to iteratively minimize this difference, aiming to reduce the root mean square error (RMSE), a measure of prediction accuracy.

Places unrated by users receive a predicted rating of 0 in the user-place rating matrix, determined by the dot product of P and Q. Recommendations are made to users based on predicted ratings equal to or greater than 3. The cold start problem is addressed by leveraging content-based recommendation methods.

Table 4: sample user's ratings in places

user_id	1	2	3	4	5	6	7	8	9	10	...	494	495	496	497	498	499	500	501	502	503
33	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

Table 5: sample result obtained from matrix factorization

user_id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
33	2.840462	2.900413	2.909645	2.993261	3.096320	1.023177	2.982202	3.097740	3.024909	3.035303	2.984016	3.047338	2.838270	2.774378	3.030733
34	2.984841	2.968045	2.962930	2.971142	2.981841	1.911738	2.974985	2.959278	2.970138	2.977445	2.999445	2.981714	2.977052	2.983777	2.979000

In our system, place recommendations are made using matrix factorization only when a user has rated over 50 places and the total number of users exceeds 50, enhancing prediction accuracy. To address sparsity issues, we continuously train the matrix factorization algorithm on user rating data. However, since this process cannot be performed in real-time, we schedule computation daily at midnight using the Django apscheduler library.

Once a sufficient number of ratings are obtained, we obtain two sets of recommendations from content-based and collaborative filtering approaches. These results are combined and displayed on the front page, categorized into 'top recommendations' and 'other recommendations'. In the 'top recommendations' section, places are selected if they rank among the top 15 in content-based filtering results and have a calculated rating above 4 from collaborative filtering. This ensures a high likelihood of user satisfaction. The remaining recommendations are shuffled and displayed randomly to prevent user boredom from seeing the same content repeatedly. Both 'top recommendations' and 'other recommendations' sections are shuffled independently to ensure variety for the user.

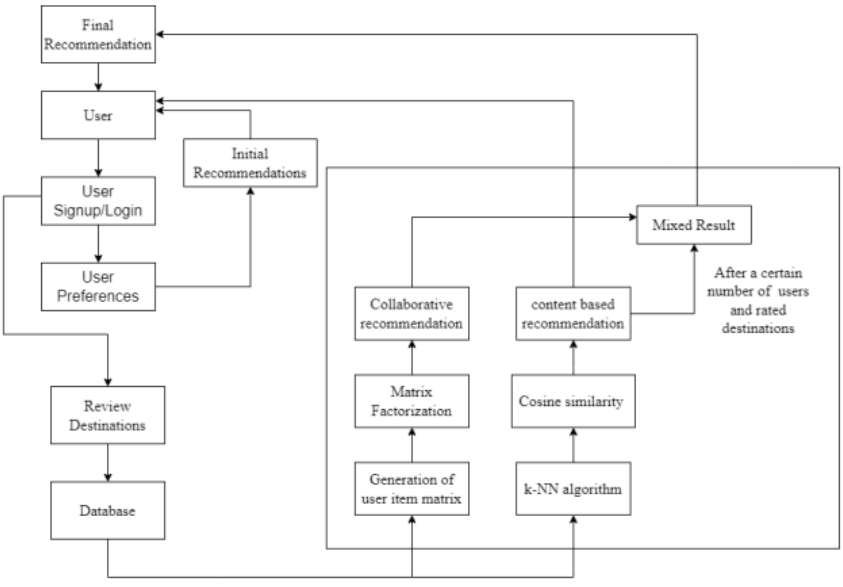

Figure 1: System Block Diagram

C. Results

1) Initial recommendation: For a new user, when they create an account and log in for the first time, they are redirected to a profile setup form where they can select preferences from different categories. According to those preferences, they receive initial recommendations on the homepage. The main page consists of popular places and recommended places.

2) Content based recommendation: After the user rates certain places, a list of recommendations is displayed to the user under "Recommendations for You". Content-based filtering techniques are used for this purpose. Content will be generated for users with fewer ratings.

3) Hybrid recommendation: Using matrix factorization technique, collaborative recommendation is formed and its intersection with the content-based recommendation, subject to certain conditions, is provided to the user under "Top Recommendations for You". The remaining list of matrix factorization recommendations is combined with content-based recommendations in the "Recommendations for You" section.

4) Rate and Review Places: After getting recommendations, the user can view, visit and rate the places. He/she can also add comments as a form of review. The user can also search for places on his/her own and rate or review the place which will be reflected in the database.

D. Conclusion:

In conclusion, our project has successfully developed an automated recommender system tailored for Nepal's e-tourism industry. By leveraging advanced technologies, we've bridged the gap between tourists and destinations, providing personalized recommendations that align with individual preferences. Our system promotes both well-known and lesser-known destinations, contributing to a more balanced and enriching tourism landscape. Overall, we've demonstrated the potential of recommender systems to enhance the travel experience and support the sustainable growth of Nepal's tourism industry. In addition, the versatility of our recommender algorithm extends beyond e-tourism, as it can be applied to other domains and effectively operates even with limited data.

E. Future Enhancements:

- Web-crawlers can be used to give more specific recommendations.
- Implicit ratings (view-time, no. of views, clicks, searches, etc.) can be embedded in the system so as to enhance the recommendations.
- Hotel recommendations can be added with an economical point of view.
- Neural network and advanced machine learning can be used for predicting the user's exact preferences.

F. References:

[1] I. Pilaszy, B. Nemmeth, D. Tikk, G. Takács, Matrix Factorization and Neighbor Based Algorithms for the netflix prize problem, ACM, 2009.

[2] A. T. a. M. Jahrer, "The BigChaos Solution to the Netflix Grand Prize," 2009.

[3] R. H. Singh, S. Maurya, T. Tripathi, T. Narula, Gaurav Srivastav, "Movie Recommendation System using Cosine Similarity and Knn," International Journal of Engineering and Advanced Technology, vol. 9, no. 5, 2020.

[4] V Bijalwan, P. Kumari, J. Pascual, "KNN based Machine Learning Approach for Text and Document Mining," International Journal of Database Theory and Application, vol. 7, pp. 61-70, 2014.

[5] Ö. F. Aktulum, "MATRIX FACTORIZATION WITH STOCHASTIC GRADIENT DESCENT FOR RECOMMENDER SYSTEMS," 2019.

[6] Z. L. B. Wang, "Tourism recommendation system based on data mining," Journal of Physics: Conference Series, 2019.

[7] S. R. V. P. Vinodhini G, "A State Of The Art Survey On Cold Start Problem In A Collaborative Filtering System," International Journal of Scientific and Technology Research, vol. 9, pp. 2606-2612, 2020.

[8] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," Advances in Artificial Intelligence, vol. 2009, pp. 1-19, 2009.